Before you Begin

This topic covers the prerequisites and limitations related to using the NITRO APIs.

Prerequisites

To use NITRO, you must have a basic understanding of the Citrix ADC appliance and you must make sure that the client application has one of the following:

- Access to a Citrix ADC appliance.
- To use REST APIs through HTTP, you must have a system to generate HTTP or HTTPS requests (payload in JSON format) to the Citrix ADC appliance. You can use any programming language or tool.
- For Java clients, you must have a system where Java Development Kit (JDK) 1.5 or later is available. The JDK can be downloaded from http://www.oracle.com/technetwork/java/javase/downloads/index.html.

  The NITRO library (available in <NITRO_SDK_HOME>/lib) must be installed on the client path. For installation instructions, read the <NITRO_SDK_HOME>/README.txt file.
- For .NET clients, you must have a system with .NET framework 3.5 or later installed. The .NET framework can be downloaded from http://www.microsoft.com/downloads/en/default.aspx.

  The NITRO library (available in <NITRO_SDK_HOME>/lib) must be installed on the client path. For installation instructions, read the <NITRO_SDK_HOME>/README.txt file.
- For Python clients, you must have a system with Python 2.7 or above version.

  The NITRO library (available in <NITRO_SDK_HOME>/lib) must be installed on the client path. For installation instructions, read the <NITRO_SDK_HOME>/README.txt file.

Obtaining the NITRO SDK Package

The NITRO package is available as a tar file on the Downloads page of the configuration utility of the Citrix ADC appliance. You must download and untar the file to a folder on your local system. This folder is referred to as \<NITRO_SDK_HOME\> in this documentation.

The folder contains the NITRO libraries in the lib subfolder. The libraries must be added to the client application classpath to access NITRO functionality. The \<NITRO_SDK_HOME\> folder also provides samples and documentation that can help you understand the NITRO SDK.

Limitations

The section lists the Citrix ADC operations that cannot be performed by using NITRO API.

`Note:` These operations can be performed on the Citrix ADC CLI or the GUI.

- install API and diff API on nsconfig resource
- UI-internal APIs (update, unset, and get)
- show ns info
- shutdown
- Application firewall API
- importwsdl
- importcustom
- importxmlschema
- importxmlerrorpage
- importhtmlerrorpage
- rmcustom
- rmxmlschema
- rmxmlerrorpage
- rmhtmlerrorpage
- CLI-specific API
- start nstrace/stop nstrace/show nstrace
- scp
- configaudit
- show defaults
- show permission
- batch
- source

NITRO Changes Across Releases

Some NITRO API have changed across releases. This topic details information which can help you avoid compatibility issues in your application. The changes are categorized as:

- Changes made from 12.0-57.24 -> 12.1-48.13
- Changes made from 11.1-53.3 -> 12.0-48.x
- Changes Made from 10.5 57.x -> 11.0
- Changes made from 9.3 -> 10.1 or 10.5

Changes made from 12.0-57.24 -> 12.1-4813

- `Changes across NITRO flavors`

  The NITRO changes that were made in release 12.1-48.13 when compared with release 12.0-57.24.

TYPE OF CHANGE	RESOURCE	METHOD/ ATTRIBUTE	REMARKS
Attribute removed	nsicapprofile	insertxserverip, insertxclientip, sendreqinresmod, insertxauthuseruri, onservererrorresponse	-
Attribute removed	lsnclient	td	“td” is applicable to lsnclientonly in conjunction with network/network6. This attribute is now a part of lsnclient_network_binding and lsclient_netwrok6_binding resources.

TYPE OF CHANGE	RESOURCE	METHOD/ ATTRIBUTE	REMARKS
Attribute removed	nspartition_stats	minbandwidth	minbandwidth setting on the partition is deprecated.
Attribute removed	cloudparameter	verifyurl	This attribute is renamed to activationcode.
Attribute removed	appfwprofile_safeobject_binding	expression	This attribute is renamed to as_expression.

- `Changes specific to NITRO SDKs`

  The SDK-specific changes that were made in release 12.1-48.13 when compared with release 12.0-57.24.

TYPE OF CHANGE	CLASS	METHOD/ ATTRIBUTE	REMARKS
Method removed	nsacls	(base_response) renumber(nitro_service session)	Alternate method - (base_response)renumber(nitro_service client, nsacls resource)
Method removed	nsacls	(base_response) clear(nitro_service session)	Alternate method - (base_response)clear(nitro_service client, nsacls resource)
Method removed	nsacls	(base_response) apply(nitro_service session)	Alternate method - (base_response)apply(nitro_service client, nsacls resource)
Method removed	nsacls6	(base_response) renumber(nitro_service session)	Alternate method - (base_response)renumber(nitro_service

TYPE OF CHANGE	CLASS	METHOD/ ATTRIBUTE	REMARKS
			client, nsacls6 resource)
Method removed	nsacls6	(base_response) clear(nitro_service session)	Alternate method - (base_response)clear(nitro_service client, nsacls6 resource)
Method removed	nsacls6	(base_response) apply(nitro_service session)	Alternate method - (base_response)apply(nitro_service client, nsacls6 resource)

Changes made from 11.1-53.3 -> 12.0-48x

- `Changes across NITRO flavors`

  The NITRO changes that were made in release 12.0-48.x when compared with release 11.1-53.3.

TYPE OF CHANGE	RESOURCE	METHOD	ATTRIBUTE
Resource removed	vpath, vpathparam	ALL	All resources and parameters related to vpath feature.
Attribute removed	l3param	-	Icmperrgenerate This attribute is related to vpath for which the support is removed in 12.0.
Attribute removed	qos_stats	-	ipcpe2qosfailedrate **Note**: Some of the counters under this resource are removed. For new

TYPE OF CHANGE	RESOURCE	METHOD	ATTRIBUTE
			definitions, see http://docs.citrix.com/en-us/netscaler/12/nitro-api/nitro-rest/api-reference/statistics/qos/qos.html
Attribute removed	vrid_interface_binding	-	trackifnum New resources (vrid_trackifnum_binding and vrid6_trackifnum_binding) are introduced to deal with track interfaces.
Attribute removed	vrid_interface_binding	-	ifaces is replaced with ifnum. Previously, ifaces parameter was used in the response and ifnum was used in the input. Now both POST and GET are consistent and use the same attribute name.
Attribute removed	sslprofile_sslciphersuite_binding	-	cipheraliasname

TYPE OF CHANGE	RESOURCE	METHOD	ATTRIBUTE
			cipheraliasname is a redundant parameter. The required information is provided by ciphername parameter.
Attribute removed	nslicense	-	Ispooledlicensing Ispooledlicensing Is replaced with a new parameter licensingmode which returns if the license of "Local".html"Pooled ".html"CICO" type.
Attribute removed	lbmonitor_sslcertkey_binding	DELETE	crlcheck

- `Changes specific to NITRO SDKs`

  The SDK-specific changes that were made in release 12.0-48.x when compared with release 11.1-53.3.

TYPE OF CHANGE	CLASS	METHOD/ ATTRIBUTE	REMARKS
Attribute type changed	pcpmap	pcpprotocol	Datatype changed from double to string.
Class removed	mediaclassification_stats	-	-

TYPE OF CHANGE	CLASS	METHOD/ ATTRIBUTE	REMARKS
Method removed	vxlan_iptunnel_binding	add(vxlan_iptunnel_binding obj, nitro_service session) delete(vxlan_iptunnel_binding obj, nitro_service session)	Feature is deprecated, use bridgetable with broadcast mac option.
Method removed	inatsession_stats	get(inatsession obj, nitro_service session)	-

Changes Made from 10.5 57.x -> 110

`All NITRO Flavors - Changes from 10.5 57.x to 11.0`

The NITRO changes that were made in release 11.0 when compared with release 10.5 Build 57.x.

TYPE OF CHANGE	RESOURCE	ATTRIBUTE
Attribute removed	nstrace	doruntimemerge
Attribute removed	nstrace	tcpdump
Attribute removed	cacheobject	force

`NITRO SDKs - Changes from 10.5-57.x to 11.0`

The SDK-specific changes that were made in release 11.0 when compared with release 10.5 Build 57.x.

TYPE OF CHANGE	CLASS	METHOD	REPLACE WITH
Attribute missing in method	cacheobject	(base_response) flush(cacheobject obj, nitro_service session) 'force' attribute missing in this method.	-
Method missing	clustersync	(base_response) Force(clustersync obj, nitro_service session)	(base_response) Force(nitro_service session)
Method missing	shutdown	(base_response) Shutdown(shutdown obj, nitro_service session)	(base_response) Shutdown(nitro_service session)
Method missing	systemfile	(systemfile) get(systemfile obj, nitro_service session)	-
Method missing	sslfips	(base_response) reset(sslfips obj, nitro_service session)	(base_response) reset(nitro_service session)

Changes made from 9.3 -> 10.1 or 105

`Note.` No changes are introduced from release 10.1 to release 10.5. Therefore, you should not face any compatibility issues when migrating from release 10.1 to 10.5.

`All NITRO Flavors - Changes from 9.3 to 10.1/10.5`

The NITRO changes that were made in release 10.1/10.5 when compared with release 9.3.

TYPE OF CHANGE	RESOURCE	METHOD	ATTRIBUTE
Resource removed	lbmonitor_lbmetrictable_binding Replaced with the resource 'lbmonitor_metric_binding'.	-	-
Method removed	vserver	GET Perform the GET operation on specific virtual server types such as lb/cr/cs.	-
Method removed	filterpolicy	POST with 'action=unset' This method is removed as unsetting the attributes('action') of a policy makes it invalid.	-
Method removed	auditsyslogpolicy	POST with 'action=unset' This method is removed as unsetting the attributes('action') of a policy makes it invalid.	-

TYPE OF CHANGE	RESOURCE	METHOD	ATTRIBUTE
Method removed	auditnslogpolicy	POST with 'action=unset' This method is removed as unsetting the attributes('action') of a policy makes it invalid.	-
Method removed	authorizationpolicy	POST with 'action=unset' This method is removed as unsetting the attributes('action') of a policy makes it invalid.	-
Return-type changed	snmpengineid	GET Return type changed to an array.	-
Return-type changed	nshostname	GET Return type changed to an array.	-
Attribute-type changed	appfwpolicy_lbvserver_binding	-	activepolicy Data type changed from Boolean to Integer.

TYPE OF CHANGE	RESOURCE	METHOD	ATTRIBUTE
Attribute-type changed	appfwpolicy_appfwglobal_binding	-	activepolicy Data type changed from Boolean to Integer.
Attribute-type changed	vlan	-	portbitmap Data type changed from uint to ulong.
Attribute-type changed	vlan	-	tagbitmap Data type changed from uint to ulong.
Attribute removed	policypatset_pattern_binding	-	indextype This attribute is moved to 'policypatset' resource as this attribute is applicable at patset level.
Attribute removed	system_stats	-	powersupply1failure Replaced with 'powersupply1status'. Note: Change is applicable from release 9.3 Build 65.8.
Attribute removed	system_stats	-	powersupply2failure

TYPE OF CHANGE	RESOURCE	METHOD	ATTRIBUTE
			Replaced with 'powersupply2status'. Note: Change is applicable from release 9.3 Build 65.8.
Attribute removed	server_servicegroup_binding	-	servicetype Replaced with 'svctype'.
Attribute removed	server_service_binding	-	servicetype Replaced with 'svctype'.
Attribute removed	crvserver	-	hits Hits are calculated per policy binding hence moved this parameter to binding resources.
Attribute removed	crvserver	-	dstvsvr Replaced with 'destinationvserver'.
Attribute removed	crvserver	-	destvserver Replaced with 'domain'.
Attribute removed	crvserver	-	dnsvserver

TYPE OF CHANGE	RESOURCE	METHOD	ATTRIBUTE
			Replaced with 'dnsvservername'.
Attribute removed	appflowpolicylabel	-	type Replaced with 'policylabeltype'.
Attribute removed	sslcipher	-	ciphgrpals Replaced with 'ciphergroupname'.
Attribute removed	csvserver_cspolicy_binding	-	targetvserver Replaced with 'targetlbvserver'. Note: This change is applicable for the 'sslcipher_*_binding' resources also.
Attribute removed	csvserver_cspolicy_binding	-	targetvserver Replaced with 'targetlbvserver'.
Attribute removed	rewriteaction	-	allow_unsafe_pi1, allow_unsafe_pi Replaced with 'bypassSafetyCheck'.
Attribute removed	nsconfig	-	nwfwmode Marked as a hidden attribute.

`NITRO SDKs - Changes from 9.3 to 10.1/10.5`

The SDK-specific changes that were made in release 10.1/10.5 when compared with release 9.3.

TYPE OF CHANGE	CLASS	METHOD	REPLACE WITH
Class removed	Routerbgp	-	This class is removed as all router configurations are deprecated in 9.2.
Method signature changed	dnsptrrec	get(dnsptrrec obj, nitro_service session)	get(nitro_service session, String reversedomain)
Method signature changed	dnsaddrec	get(dnsaddrec obj, nitro_service session)	get(nitro_service session, String hostname)
Method signature changed	dnsnsrec	get(dnsnsrec obj, nitro_service session)	get(nitro_service session, String domain)
Method signature changed	snmpengineid	unset(nitro_service session, String[] args)	unset(nitro_service session, snmpengineid resource, String[] args)
Method signature changed	arp	arp.get(nitro_service session, String ipaddress)	arp.get(nitro_service session, arp resource)
Method signature changed	nsip	get(nitro_service session, String ipaddress)	get(nitro_service client, nsip resource)

TYPE OF CHANGE	CLASS	METHOD	REPLACE WITH
Method signature changed	nsip6	get(nitro_service session, String ipv6address)	get(nitro_service session, nsip6 resource)
Method signature changed	dnsmxrec	dnsmxrec.get(dnsmxrec obj, nitro_service session)	dnsmxrec[] get(nitro_service service, dnsmxrec_args args)
Method Missing	authenticationnegotiatepolicy	(base_response) unset(nitro_service session, String[] args, String name)' is missing in	-
Method missing	authenticationnegotiatepolicy	(base_response) unset(authenticationnegotiatepolicy obj, nitro_service session, String[] args)	-
Attribute missing in method	nsconfig	(base_response) update(nsconfig obj, nitro_service session) 'nwfwmode' attribute is missing in this method.	-

Performing Basic Citrix ADC Operations

This section covers procedures for performing the following basic Citrix ADC operations using NITRO APIs.

- Connecting to the Citrix ADC Appliance
- Enabling Citrix ADC Features and Modes
- Saving Citrix ADC Configurations
- Killing a System Session
- Disconnecting from the Citrix ADC Appliance

Connecting to the Citrix ADC Appliance

The first step towards using NITRO is to establish a session with the Citrix ADC appliance and then authenticate the session by using the Citrix ADC administrator's credentials.

Some points to note with regards to session timeout:

- When restricted timeout param is enabled, NITRO, by default, uses the timeout value that is configured for the logged in user. You can customize this value but it must be limited to the value specified for the user. If no value is specified for the user, the default timeout value of 15 minutes is used.
- When restricted timeout param is not enabled, NITRO uses the default value of 30 minutes as session timeout.
- `Using REST APIs through HTTP`

  You must specify the username and password in the login object. The session ID that is created must be specified in the request header of all further operations in the session.
- `Note.`
  - You must have a user account on the appliance to log on to it. The configuration operations that you can perform are limited by the administrative roles assigned to your account.
- To ensure secure communication, use the HTTPS protocol in NITRO requests.

- Instead of creating a NITRO session, you can log on to the Citrix ADC appliance while performing individual operations. To do this, you must specify the username and password in the request header of the NITRO request as follows:
  - X-NITRO-USER:<username\>
  - X-NITRO-PASS:<password\>
  - Content-Type:application/json

For example, to connect and create a session with a Citrix ADC appliance with NSIP address 10.102.29.60 by using the HTTP protocol:

- `Request`

  `HTTP Method:` POST

`URL:` http://10.102.29.60/nitro/v1/config/login

`Request Headers:` Content-Type:application/json

`Request Payload:`

```
{

    "login":

    {

    "username":"admin",

    "password":"verysecret"

    }

}
```

- `Response`

  `HTTP Status Code on Success:` 201 Created

`HTTP Status Code on Failure:` 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix-ADC-specific errors). The response payload provides details of the error.

`Response Header:`

```
Set-Cookie:NITRO_AUTH_TOKEN=<tokenvalue>;

path=/nitro/v1
```

- `Using REST APIs through SDKs`

  You must create an object of the com.citrix.netscaler.nitro.service.nitro_service class by specifying the Citrix ADC IP (NSIP) address and the protocol to connect to the appliance (HTTP or HTTPS). You then use this object and log on to the appliance by specifying the user name and the password of the Citrix ADC administrator.

`Note:`

- For the python SDK, the package path is of the form nssrc.com.citrix.netscaler...
- You must have a user account on that appliance. The configuration operations that you perform are limited by the administrative roles assigned to your account.

The following sample code establishes a session with a Citrix ADC appliance with IP address 10.102.29.60 by using the HTTPS protocol and also sets a session timeout period (in seconds) of 60 minutes.

`Java - Sample code to establish session`

```
//Specify the Citrix ADC appliance IP address and protocol

nitro_service ns_session = new nitro_service("10.102.29.60","https");



//Specify the login credentials

ns_session.login("admin","verysecret",3600);
```

`.NET - Sample code to establish session`

```
//Specify the Citrix ADC appliance IP address and protocol

nitro_service ns_session = new nitro_service("10.102.29.60","https");
```

```
//Specify the login credentials

ns_session.login("admin","verysecret",3600);
```

`Python - Sample code to establish session`

```
# Specify the Citrix ADC appliance IP address and protocol

ns_session = nitro_service("10.102.29.60","https")

# Specify the login credentials

ns_session.login("admin","verysecret",3600)
```

Disable SSL Checks

When using HTTPS, you must make sure that the root CA is added to the truststore. By default, NITRO validates the SSL certificate and verifies the hostname.

- `Using REST APIs through SDKs`

  Disable these validations as shown in the following sample codes.

`Java - Sample code for disabling SSL checks`

```
ns_session.set_certvalidation(false);

ns_session.set_hostnameverification(false);
```

`.NET - Sample code for disabling SSL checks`

```
ns_session.certvalidation = false;

ns_session.hostnameverification = false;
```

`Python - Sample code for disabling SSL checks`

```
ns_session.certvalidation = false
```

Enabling Citrix ADC Features and Modes

Some Citrix ADC features and modes are disabled by default and therefore must be enabled before they can be configured. To enable a Citrix ADC feature or mode, specify the action as "enable" in the URL query string, and in the request payload, specify the feature or mode to be enabled.

- `Using REST APIs through HTTP`

  To disable a feature or mode, in the URL query string, specify the action as "disable".

For example, to enable the load balancing and content switching features:

- `Request`

  `HTTP Method:` POST

`URL:` http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/nsfeature?action=enable

`Request Headers:`

Cookie:NITRO_AUTH_TOKEN=<tokenvalue>

Content-Type:application/json

`Request Payload:`

```
{

    "nsfeature":

    {

        "feature":
```

```
        [
            "LB",
            "CS"
        ]
    }
}
```

- `Response`

  `HTTP Status Code on Success:` 200 OK

`HTTP Status Code on Failure:` 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix-ADC-specific errors). The response payload provides details of the error.

To enable the L2 and fast ramp modes:

- `Request`

  `HTTP Method:` POST

`URL:` http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/nsmode?action=enable

`Request Headers:`

Cookie:NITRO_AUTH_TOKEN=<tokenvalue>

Content-Type:application/json

`Request Payload:`

```
{
    "nsmode":
    {
        "mode":
```

```
        [
            "L2",
            "FR"
        ]
    }
}
```

- `Response`

  `HTTP Status Code on Success:` 200 OK

`HTTP Status Code on Failure:` 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix-ADC-specific errors). The response payload provides details of the error.

Saving Citrix ADC Configurations

To make sure that the configurations persist on rebooting the appliance, you must save the Citrix ADC configurations. To save the configurations, specify the action as "save" in the URL query string.

- `Using REST APIs through HTTP`

  To save the configurations:

- `Request`

  `HTTP Method:` POST

`URL:` http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/nsconfig?action=save

`Request Headers:`

Cookie:NITRO_AUTH_TOKEN=<tokenvalue>

Content-Type:application/json

`Request Payload:`

```
{

    "nsconfig":

    {}

}
```

- `Response`

  `HTTP Status Code on Success:` 200 OK

`HTTP Status Code on Failure:` 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix-ADC-specific errors). The response payload provides details of the error.

Killing a System Session

A Citrix ADC administrator can kill any system session by specifying the action as "kill" in the URL query string and by specifying the required system session ID in the request payload.

- `Using REST APIs through HTTP`

  For example, to kill a system session that has ID as 311:

- `Request`

  `HTTP Method:` POST

`URL:` http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/systemsession?action=kill

`Request Headers:`

Cookie:NITRO_AUTH_TOKEN=<tokenvalue>

Content-Type:application/json

`Request Payload:`

```
{

    "systemsession":

    {
```

```
    "sid":"311"

    }

}
```

- `Response`

  `HTTP Status Code on Success:` 200 OK

`HTTP Status Code on Failure:` 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix-ADC-specific errors). The response payload provides details of the error.

Disconnecting from the Citrix ADC Appliance

Before disconnecting (logging out) from the Citrix ADC appliance, make sure that you have saved the Citrix ADC configurations.

- `Using REST APIs through HTTP`

  To logout of the Citrix ADC appliance:

- `Request`

  `HTTP Method:` POST

`URL:` http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/logout

`Request Headers:`

Cookie:NITRO_AUTH_TOKEN=<tokenvalue>

Content-Type:application/json

`Request Payload:`

```
{

    "logout":{}

}
```

- `Response`

`HTTP Status Code on Success:` 200 OK

`HTTP Status Code on Failure:` 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix-ADC-specific errors). The response payload provides details of the error.

Performing Citrix ADC Resource Operations

Citrix ADC resources are organized into a set of packages or namespaces. Each package or namespace corresponds to a Citrix ADC feature. For example, all load-balancing related resources, such as load balancing virtual server, load balancing group, and load balancing monitor are available in com.citrix.netscaler.nitro.resource.config.lb.

Similarly, all application firewall related resources, such as application firewall policy and application firewall archive are available in com.citrix.netscaler.nitro.resource.config.appfw.

Each Citrix ADC resource is represented by a class. For example, the class that represents a load balancing virtual server is called lbvserver (in com.citrix.netscaler.nitro.resource.config.lb). The state of a resource is represented by properties of a class. You can get and set the properties of the class.

The setter and getter properties are always executed locally on the client. They do not involve any network interaction with the NITRO web service. All properties have basic simple types: integer, long, boolean, and string.

Adding a Citrix ADC Resource

This topic covers adding a Citrix ADC resource by using REST APIs through HTTP or SDKs.

- `Using REST APIs through HTTP`

  To create a new resource (for example, an lbvserver) on the appliance, specify the resource name and other related arguments in the specific resource object.

For example, to create an load balancing virtual server named MyFirstLbVServer:

- `Request`

  `HTTP Method:` POST

`URL:` http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/lbvserver

`Request Headers:`

Cookie:NITRO_AUTH_TOKEN=<tokenvalue>

Content-Type:application/json

`Request Payload:`

```
{

    "lbvserver":

    {

    "name":"MyFirstLbVServer",

    "servicetype":"http"

    }

}
```

- `Response`

  `HTTP Status Code on Success:` 201 Created

`HTTP Status Code on Failure:` 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix-ADC-specific errors). The response payload provides details of the error.

- `Using REST APIs through SDKs`

  To create a new resource, instantiate the resource class, configure the resource by setting its properties locally, and then upload the new resource instance to the Citrix ADC appliance.

The following sample code creates a load balancing virtual server.

`Java - Sample code to add a Citrix ADC resource`

```
//Create an instance of the lbvserver class



lbvserver new_lbvserver_obj = new lbvserver();



//Set the properties of the resource locally
```

```
new_lbvserver_obj.set_name("MyFirstLbVServer");
new_lbvserver_obj.set_ipv46("10.102.29.88");
new_lbvserver_obj.set_port(88);
new_lbvserver_obj.set_servicetype("HTTP");
new_lbvserver_obj.set_lbmethod("ROUNDROBIN");

//Upload the resource to Citrix ADC

lbvserver.add(ns_session,new_lbvserver_obj);
```

`.NET - Sample code to add a Citrix ADC resource`

```
//Create an instance of the lbvserver class

lbvserver new_lbvserver_obj = new lbvserver();

//Set the properties of the resource locally

new_lbvserver_obj.name = "MyFirstLbVServer";
new_lbvserver_obj.ipv46 = "10.102.29.88";
new_lbvserver_obj.port = 88;
new_lbvserver_obj.servicetype = "HTTP";
```

```
new_lbvserver_obj.lbmethod = "ROUNDROBIN";


//Upload the resource to Citrix ADC


lbvserver.add(ns_session,new_lbvserver_obj);
```

`Python - Sample code to add a Citrix ADC resource`

```


# Create an instance of the lbvserver class


new_lbvserver_obj = lbvserver()


# Set the properties of the resource locally


new_lbvserver_obj.name = "MyFirstLbVServer"

new_lbvserver_obj.ipv46 = "10.102.29.88"

new_lbvserver_obj.port = 88

new_lbvserver_obj.servicetype = "HTTP"

new_lbvserver_obj.lbmethod = "ROUNDROBIN"


# Upload the resource to CItrix ADC


```

```
lbvserver.add(ns_session, new_lbvserver_obj)
```

Enabling a Citrix ADC Resource

This topic covers enabling a Citrix ADC resource by using REST APIs through HTTP or SDKs.

- `Using REST APIs through HTTP`

  To enable a resource on the Citrix ADC appliance, specify the action as "enable" in the URL query string, and in the request payload, specify the resource to be enabled. To disable a resource, in the URL query string, specify the action as "disable".

For example, to enable a load balancing virtual server named MyFirstLbVServer.

- `Request`

  `HTTP Method:` POST

`URL:` http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/lbvserver?action=enable

`Request Headers:`

Cookie:NITRO_AUTH_TOKEN=<tokenvalue>

Content-Type:application/json

`Request Payload:`

```
{

    "lbvserver":

    {

    "name":"MyFirstLbVServer"

    }

}
```

- `Response`

`HTTP Status Code on Success:` 200 OK

`HTTP Status Code on Failure:`

4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix-ADC-specific errors). The response payload provides details of the error.

- `Using REST APIs through SDKs`

  To enable a resource, invoke the enable() method and to disable, invoke the disable() method.

The following sample code enables a load balancing virtual server named lb_vip.

`Java - Sample code to enable a Citrix ADC resource`

```
lbvserver obj = new lbvserver();



obj.set_name("lb_vip");



lbvserver.enable(ns_session, obj);
```

`.NET - Sample code to enable a Citrix ADC resource`

```
lbvserver obj = new lbvserver();



obj.name = "lb_vip";



lbvserver.enable(ns_session, obj);
```

`Python - Sample code to enable a Citrix ADC resource`

```
obj = lbvserver()
```

```
obj.name = "lb_vip"



lbvserver.enable(ns_session, obj)
```

Retrieving Properties of Citrix ADC Resources

This topic covers retrieving properties of a Citrix ADC resource by using REST APIs through HTTP or SDKs.

- `Using REST APIs through HTTP`

  NITRO provides multiple approaches using which you can retrieve resources and their relevant details. The following table explains each of these approaches with the required URL.

Retrieving all details of all resources of a specific type	In the URL, specify the type of resource for which you want to retrieve the details. For example, to retrieve all details of load balancing virtual servers available on a Citrix ADC appliance. http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/lbvserver
Retrieving all details of a specific resource	In the URL, specify the name of resource for which you want to retrieve the details. For example, to retrieve all details of a load balancing virtual server named MyFirstLbVServer: http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/lbvserver/MyFirstLbVServer
Retrieving summary or detailed view of resources	In the URL, use the "view" query parameter to specify whether you want to view the

	summary (mandatory parameters) or detail (all parameters). For example, to view the mandatory parameters for all load balancing virtual servers: http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/lbvserver?view=summary Note: By default, the retrieved results are displayed in detail view (?view=detail).
Retrieving all details of resources that have multiple unique identifiers	In the URL, specify the type of resource and use the "args" query parameter to specify the unique attributes and the values for those attributes. For example, to get the application firewall profiles that have unique identifiers "name" and "starturl" as appfw1 and aa respectively. http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/appfwprofile?args=name:appfw1,starturl:aa
Retrieving specific details of all resources of a specific type	In the URL, specify the type of the resource and use the "attrs" query parameter to specify the resource details that you want to retrieve. For example, to retrieve the "name" and "lbmethod" of all load balancing virtual servers: http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/lbvserver?attrs=name,lbmethod
Retrieving specific details of a specific resource	In the URL, specify the type and name of the resource and use the "attrs" query parameter

	to specify the resource details that you want to retrieve. For example, to retrieve the "name" and "lbmethod" of a load balancing virtual server named "MyFirstLbVServer": http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/lbvserver/MyFirstLbVServer?attrs=name,lbmethod
Filtering the retrieved resources	In the URL, specify the type of resource and use the "filter" query parameter to specify the attribute(s) and the value(s) of the attributes. The resources fetched will be filtered based on the filter criteria. Note: The filter query parameter supports the use of PCRE regular expressions. For example, to filter the load balancing virtual servers where the "lbmethod" is ROUNDROBIN. http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/lbvserver?filter=lbmethod:ROUNDROBIN
Retrieving resources in paginated manner	If the request is likely to result in a large number of resources, you can divide the results into pages and retrieve them page by page (paginated). For example, if you are retrieving all the 53 load balancing virtual servers of a Citrix ADC appliance, instead of retrieving all 53 in one response, you can configure the results to be divided into 6 pages each having 10 results. In the URL, specify the name of the resource and use the following query parameters:

	• "pageno" - The page number to be displayed. • "pagesize" - The number of resources to be displayed in each page. For example, to retrieve the load balancing virtual servers in a paginated form, first get a count (using the "count" query parameter shown in below row) of the load balancing virtual servers. Then, accordingly specify the number of results for each page and then specify the page number to be displayed. http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/lbvserver?pagesize=10&pageno=3

For example, to retrieve only the name and load balancing method of all load balancing virtual servers on a Citrix ADC.

- `Request`

  `HTTP Method:` GET

`URL:` http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/lbvserver?attrs=name,lbmethod

`Request Headers:`

Cookie:NITRO_AUTH_TOKEN=<tokenvalue>

Accept:application/json

- `Response`

  `HTTP Status Code on Success:` 200 OK

`HTTP Status Code on Failure:`

4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix-ADC-specific errors). The response payload provides details of the error.

`Response Header:`

Content-Type:application/json

`Response Payload:`

```
{
    lbvserver:
    {
        name: "test",
        lbmethod: "LEASTCONNECTION"
    }
    {
        name: "test1",
        lbmethod: "LEASTCONNECTION"
    }
}
```

- `Using REST APIs through SDKs`

  To retrieve the properties of a resource, you retrieve the resource object from the Citrix ADC appliance. Once the object is retrieved, you can extract the required properties of the resource locally, without further network traffic.

The following sample code retrieves the details of a load balancing virtual server.

`Java - Sample code to get details of resource`

```
//Retrieve the resource object from the Citrix ADC


new_lbvserver_obj = lbvserver.get(ns_session,"MyFirstLbVServer");


//Extract the properties of the resource from the object locally
```

```
System.out.println(new_lbvserver_obj.get_name());

System.out.println(new_lbvserver_obj.get_servicetype());
```

`.NET - Sample code to get details of resource`

```
//Retrieve the resource object from the Citrix ADC


new_lbvserver_obj = lbvserver.get(ns_session,"MyFirstLbVServer");


//Extract the properties of the resource from the object locally


Console.WriteLine(new_lbvserver_obj.name);

Console.WriteLine(new_lbvserver_obj.servicetype);
```

`Python - Sample code to get details of resource`

```


# Retrieve the resource object from the CItrix ADC


new_lbvserver_obj = lbvserver.get(ns_session,"MyFirstLbVServer")


# Extract the properties of the resource from the object locally


print(new_lbvserver_obj.name)
```

```
print(new_lbvserver_obj.servicetype)
```

Filtering Results

This topic covers retrieving filtered information of a resource by using REST APIs through SDKs.

- `Using REST APIs through SDKs`

  You can also retrieve resources by specifying a filter on the value of their properties by using the com.citrix.netscaler.nitro.util.filtervalue class.

For example, you can retrieve all the load balancing virtual servers that have their port set to 80 and servicetype to HTTP.

`Java - Sample code to get filtered results`

```
filtervalue[] filter = new filtervalue[2];

filter[0] = new filtervalue("port","80");

filter[1] = new filtervalue("servicetype","HTTP");

lbvserver[] result = lbvserver.get_filtered(ns_session,filter);
```

`.NET - Sample code to get filtered results`

```
filtervalue[] filter = new filtervalue[2];

filter[0] = new filtervalue("port","80");

filter[1] = new filtervalue("servicetype","HTTP");

lbvserver[] result = lbvserver.get_filtered(ns_session,filter);
```

`Python - Sample code to get filtered results`

```
filter_params = []



      filter_params = [ filtervalue() for _ in range(2)]
```

```
filter_params[0] = filtervalue("servicetype","HTTP")

filter_params[1] = filtervalue("port","80")

result = lbvserver.get_filtered(ns_session1, filter_params)
```

Retrieving Statistics of Citrix ADC Resources

The Citrix ADC appliance collects statistics about the usage of its features and the corresponding resources. NITRO can retrieve these statistics. Not all Citrix ADC features and resources have statistic objects associated with them.

- `Using REST APIs through HTTP`

  To get statistics of a feature, the URL format must be: http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/stat/\<feature_name\>.

To get statistics of a resource, the URL format must be: http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/stat/\<resource_type\>/\<resource_name\>.

To get statistics of the services and service groups that are bound to a load balancing virtual server, the URL format must be: http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/stat/lbvserver/\<name\>?statbindings=yes*.

- `Request`

  `HTTP Method:` GET

`URL:` http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/stat/lbvserver/MyFirstLbVServer

`Request Headers:`

Cookie:NITRO_AUTH_TOKEN=\<tokenvalue\>

Accept:application/json

- `Response`

  `HTTP Status Code on Success:` 200 OK

`HTTP Status Code on Failure:`

4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix-ADC-specific errors). The response payload provides details of the error.

`Response Payload:`

```
{

    "lbvserver":

    [

        {

            "name":"MyFirstLbVServer",

            "establishedconn":0,

            "vslbhealth":0,

            "primaryipaddress":"0.0.0.0",

            ...
```

```
        }

    ]

}
```

To get statistics of the bound entities, use statbindings=yes. For example, to get the statistics of the services that are bound to a load balancing virtual server named MyFirstLbVServer.

- `Request`

  `HTTP Method:` GET

`URL:` http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/stat/lbvserver/MyFirstLbVServer?statbindings=yes

`Request Headers:`

Cookie:NITRO_AUTH_TOKEN=\<tokenvalue\>

Accept:application/json

- `Response`

  `HTTP Status code on success:` 200 OK

`HTTP Status code on failure:`

4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix-ADC-specific errors). The response payload provides details of the error.

`Response Header:`

Content-Type:application/json

`Response Payload:`

```
{

"lbvserver": [{

    "name": "MyFirstLbVServer",

    "sortorder": "descending",

    "vsvrsurgecount": "0",

    "establishedconn": "0",

    ...

    "service": [{

        "name": "s1",

        "throughput": "0",

        "throughputrate": 0,
```

```
        "avgsvrttfb": "0",

        "primaryipaddress": "1.2.3.5",

        "primaryport": 80,

        "servicetype": "HTTP",

        "state": "DOWN",

            "totalrequests": "0",

            "requestsrate": 0,

            ...

            }]
    }]

}
```

- `Using REST APIs through SDKs`

The Citrix ADC appliance collects statistics about the usage of its features and the corresponding resources. You can retrieve these statistics by using NITRO API. The statistics APIs are available in different packages from the configuration APIs.

For example, the API to retrieve statistics of the load balancing virtual server are available in com.citrix.netscaler.nitro.resource.stat.lb.

`Note.`

- For the python SDK, the package path is of the form nssrc.com.citrix.netscaler.....
- Not all Citrix ADC features and resources have statistic objects associated with them.

The following sample code retrieves the statistics of a load balancing virtual server and displays some of the statistics returned.

`Java - Sample code to get feature statistics`

```
lbvserver_stats stats =
lbvserver_stats.get(ns_session,"MyFirstLbVServer");



System.out.println(stats.get_curclntconnections());



System.out.println(stats.get_deferredreqrate());
```

`.NET - Sample code to get feature statistics`

```
lbvserver_stats stats =
lbvserver_stats.get(ns_session,"MyFirstLbVServer");



Console.WriteLine(stats.curclntconnections);



Console.WriteLine(stats.deferredreqrate);
```

`Python - Sample code to get feature statistics`

```
stats = lbvserver_stats.get(ns_session,"MyFirstLbVServer")



print(stats.curclntconnections)



print(stats.deferredreqrate)
```

Retrieving Statistics of Citrix ADC Resources

The Citrix ADC appliance collects statistics about the usage of its features and the corresponding resources. NITRO can retrieve these statistics. Not all Citrix ADC features and resources have statistic objects associated with them.

- `Using REST APIs through HTTP`

  To get statistics of a feature, the URL format must be: http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/stat/\<feature_name\>.

To get statistics of a resource, the URL format must be: http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/stat/\<resource_type\>/\<resource_name\>.

To get statistics of the services and service groups that are bound to a load balancing virtual server, the URL format must be: http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/stat/lbvserver/\<name\>?statbindings=yes*.

- `Request`

  `HTTP Method:` GET

`URL:` http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/stat/lbvserver/MyFirstLbVServer

`Request Headers:`

Cookie:NITRO_AUTH_TOKEN=\<tokenvalue\>

Accept:application/json

- `Response`

  `HTTP Status Code on Success:` 200 OK

`HTTP Status Code on Failure:`

4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix-ADC-specific errors). The response payload provides details of the error.

`Response Payload:`


```
{

    "lbvserver":

    [

        {

            "name":"MyFirstLbVServer",

            "establishedconn":0,

            "vslbhealth":0,

            "primaryipaddress":"0.0.0.0",

            ...
```

```
        }


    ]


}
```

To get statistics of the bound entities, use statbindings=yes. For example, to get the statistics of the services that are bound to a load balancing virtual server named MyFirstLbVServer.

- `Request`

  `HTTP Method:` GET

`URL:` http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/stat/lbvserver/MyFirstLbVServer?statbindings=yes

`Request Headers:`

Cookie:NITRO_AUTH_TOKEN=\<tokenvalue\>

Accept:application/json

- `Response`

  `HTTP Status code on success:` 200 OK

`HTTP Status code on failure:`

4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix-ADC-specific errors). The response payload provides details of the error.

`Response Header:`

Content-Type:application/json

`Response Payload:`

```
{

"lbvserver": [{

    "name": "MyFirstLbVServer",

    "sortorder": "descending",

    "vsvrsurgecount": "0",

    "establishedconn": "0",

    ...

    "service": [{

        "name": "s1",

        "throughput": "0",

        "throughputrate": 0,
```

```
        "avgsvrttfb": "0",

        "primaryipaddress": "1.2.3.5",

        "primaryport": 80,

        "servicetype": "HTTP",

        "state": "DOWN",

            "totalrequests": "0",

            "requestsrate": 0,

            ...

            }]
    }]

}
```

- `Using REST APIs through SDKs`

The Citrix ADC appliance collects statistics about the usage of its features and the corresponding resources. You can retrieve these statistics by using NITRO API. The statistics APIs are available in different packages from the configuration APIs.

For example, the API to retrieve statistics of the load balancing virtual server are available in com.citrix.netscaler.nitro.resource.stat.lb.

`Note:`

- For the python SDK, the package path is of the form nssrc.com.citrix.netscaler.....
- Not all Citrix ADC features and resources have statistic objects associated with them.

The following sample code retrieves the statistics of a load balancing virtual server and displays some of the statistics returned.

`Java - Sample code to get feature statistics`

```
lbvserver_stats stats =
lbvserver_stats.get(ns_session,"MyFirstLbVServer");



System.out.println(stats.get_curclntconnections());



System.out.println(stats.get_deferredreqrate());
```

`.NET - Sample code to get feature statistics`

```
lbvserver_stats stats =
lbvserver_stats.get(ns_session,"MyFirstLbVServer");



Console.WriteLine(stats.curclntconnections);



Console.WriteLine(stats.deferredreqrate);
```

`Python - Sample code to get feature statistics`

```
stats = lbvserver_stats.get(ns_session,"MyFirstLbVServer")



print(stats.curclntconnections)



print(stats.deferredreqrate)
```

Updating a Citrix ADC Resource

You can modify the properties of a resource after creating it.

Some properties in some Citrix ADC resources are not allowed to be modified after creation. The port number or the service type (protocol) of a load balancing virtual server or a service, are examples of such properties. Even though the update method appears to succeed, these properties retain their original values on the appliance.

- `Using REST APIs through HTTP`

  To update the details of an existing resource on the Citrix ADC appliance, specify the name of the resource in the URL, and in the request payload,within the specific resource object, specify the name and the updated details of the resource.

For example, to change the load balancing method to ROUNDROBIN and update the comment property for a load balancing virtual server named MyFirstLbVServer:

- `Request`

  `HTTP Method:` PUT

`URL:` http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/lbvserver/MyFirstLbVServer

`Request Headers:`

Cookie:NITRO_AUTH_TOKEN=<tokenvalue>

Content-Type:application/json

`Request Payload:`

```
{

    "lbvserver":

    {

        "name":"MyFirstLbVServer",

        "lbmethod":"ROUNDROBIN",

        "comment":"Updated comments"

    }

}
```

- `Response`

  `HTTP Status Code on Success:` 200 OK

`HTTP Status Code on Failure:`

4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix-ADC-specific errors). The response payload provides details of the error.

- `Using REST APIs through SDKs`

  To update the properties of a resource, instantiate the resource class, specify the name of the resource to be updated, configure the resource by updating its properties locally, and then upload the updated resource instance to the Citrix ADC appliance.

The following sample code updates the service type and load balancing method of a load balancing virtual server.

`Java - Sample code to update a Citrix ADC resource`

```
//Create an instance of the lbvserver class



lbvserver update_lb = new lbvserver();
```

```
//Specify the name of the lbvserver to be updated

update_lb.set_name("MyFirstLbVServer");

//Specify the updated service type and lb method

update_lb.set_servicetype("https");
update_lb.set_lbmethod("LEASTRESPONSETIME");

//Upload the resource to Citrix ADC

lbvserver.update(ns_session,update_lb);
```

`.NET - Sample code to update a Citrix ADC resource`

```
//Create an instance of the lbvserver class

lbvserver update_lb = new lbvserver();

//Specify the name of the lbvserver to be updated

update_lb.name = "MyFirstLbVServer";
```

```
//Specify the updated service type and lb method



update_lb.servicetype = "https";

update_lb.lbmethod = "LEASTRESPONSETIME";



//Upload the resource to Citrix ADC



lbvserver.update(ns_session, update_lb);
```

`Python - Sample code to update a Citrix ADC resource`

```
# Create an instance of the lbvserver class



update_lb = lbvserver()



# Specify the name of the lbvserver to be updated



update_lb.name = "MyFirstLbVServer"



# Specify the updated service type and lb method



update_lb.servicetype = "https"
```

```
update_lb.lbmethod = "LEASTRESPONSETIME"



# Upload the resource to the CItrix ADC



lbvserver.update(ns_session, update_lb)
```

Binding Citrix ADC Resources

Citrix ADC resources form relationships with each other through the process of binding. This is how services are associated with a load balancing virtual server, or how policies are bound to a load balancing virtual server. Each binding relationship is represented by its own object.

A binding resource has properties representing the name of each Citrix ADC resource in the binding relationship. It can also have other properties related to that relationship (for example, the weight of the binding between an lbvserver resource and a service resource).

- `Using REST APIs through HTTP`

  To bind one Citrix ADC resource to another, you must instantiate the appropriate binding class (for example, to bind a service to a load balancing virtual server, you must instantiate the lbvserver_service_binding class) and add it to the Citrix ADC configuration (by using the static add() method on this class).

To unbind a resource, use the DELETE method and specify an "args" query string parameter in the URL that contains the attribute name and value in the relationship resource that designates the secondary resource.

The following example binds a service named svc_prod to a load balancing virtual server named MyFirstLbVServer and specify a weight for the binding.

- `Request`

  `HTTP Method:` POST

`URL:` http://<Citrix-ADC-ip-address(NSIP)\>/nitro/v1/config/lbvserver_service_binding

`Request Headers:` Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Content-Type:application/json

`Request Payload:`

```
{

"lbvserver_service_binding":

    {

    "servicename":"svc_prod",

    "weight":"20",

    "name":"MyFirstLbVServer"

    }

}
```

- `Response`

  `HTTP Status Code on Success:` 201 Created

`HTTP Status Code on Failure:` 4xx \<string \> (for general HTTP errors) or 5xx \<string \> (for Citrix-ADC-specific errors). The response payload provides details of the error.

The following example binds a policy to a policy label.

- `Request`

  `HTTP Method:` POST

`URL:` http://<Citrix-ADC-ip-address(NSIP)\>/nitro/v1/config/authenticationpolicylabel_authenticationpolicy_binding

`Request Headers:`

Cookie:NITRO_AUTH_TOKEN=<tokenvalue>

Content-Type:application/json

`Request Payload:`

```
{

"authenticationpolicylabel_authenticationpolicy_binding":

    {

    "policyname":"p1",

    "priority":"100",

    "gotopriorityexpression":"END",

    "labelname":"pl1"

    }

}
```

- `Response`

  `HTTP Status Code on Success:` 201 Created

`HTTP Status Code on Failure:`

4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix-ADC-specific errors). The response payload provides details of the error.

The following example unbinds the service svc_prod from the load balancing virtual server MyFirstLbVServer.

- `Request`

  `HTTP Method:` DELETE

`URL:` http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/lbvserver_service_binding/MyFirstLbVServer?args=servicename:svc_prod

`Request Header:`

Cookie:NITRO_AUTH_TOKEN=\<tokenvalue\>

- `Response`

  `HTTP Status Code on Success:` 200 OK

`HTTP Status Code on Failure:`

4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix-ADC-specific errors). The response payload provides details of the error.

- `Using REST APIs through SDKs`

  To bind one Citrix ADC resource to another, you must instantiate the appropriate binding class (for example, to bind a service to a load balancing virtual server, you must instantiate the lbvserver_service_binding class) and add it to the Citrix ADC configuration (by using the static add() method on this class).

To unbind a resource from another, invoke the delete() method from the resource binding class, by passing the name of the two resources.

The following sample code binds a service to a load balancing virtual server, by specifying a certain weight for the binding.

`Java - Sample code for binding`

```
lbvserver_service_binding bindObj = new lbvserver_service_binding();

bindObj.set_name("MyFirstLbVServer");

bindObj.set_servicename("svc_prod");

bindObj.set_weight(20);

lbvserver_service_binding.add(ns_session,bindObj);
```

`.NET - Sample code for binding`

```
lbvserver_service_binding bindObj = new lbvserver_service_binding();

bindObj.name = "MyFirstLbVServer";

bindObj.servicename = "svc_prod";

bindObj.weight = 20;

lbvserver_service_binding.add(ns_session,bindObj);
```

`Python - Sample code for binding`

```
bindObj = lbvserver_service_binding()

bindObj.name = "MyFirstLbVServer"

bindObj.servicename = "svc_prod"

bindObj.weight = 20

lbvserver_service_binding.add(ns_session, bindObj)
```

The following code sample unbinds a service from a server.

`Java - Sample Code for unbinding`

```
lbvserver_service_binding bindObj = new lbvserver_service_binding();

bindObj.set_name("MyFirstLbVServer");

bindObj.set_servicename("svc_prod");

lbvserver_service_binding.delete(ns_session,bindObj);
```

`.NET - Sample code for unbinding`

```
lbvserver_service_binding bindObj = new lbvserver_service_binding();

bindObj.name("MyFirstLbVServer");

bindObj.servicename("svc_prod");

lbvserver_service_binding.delete(ns_session,bindObj);
```

`Python - Sample code for unbinding`

```
bindObj = lbvserver_service_binding()

bindObj.name = "MyFirstLbVServer"

bindObj.servicename = "svc_prod"
```

```
lbvserver_service_binding.delete(ns_session, bindObj)
```

Globally Binding Citrix ADC Resources

Some Citrix ADC resources can be bound globally to affect the whole system. For example, a compression policy can be bound to an load balancing virtual server, in which case the policy affects only the traffic on that load balancing virtual server. However, if bound globally, it can affect any traffic on the appliance, regardless of which virtual servers handle the traffic.

The names of NITRO resources that can be used to bind resources globally have the pattern *\<featurename \>global_\<resourcetype \>_binding*. For example, the object *aaaglobal_aaapreauthenticationpolicy_binding* is used to bind preauthentication policies globally.

- `Using REST APIs through HTTP`

  The following example binds the policy named preautpol1 globally at priority 200.

- `Request`

  `HTTP Method:` POST

`URL:`

http://<Citrix-ADC-ip-address(NSIP)\>/nitro/v1/config/aaaglobal_aaapreauthenticationpolicy_binding/preautpol1

`Request Headers:`

Cookie:NITRO_AUTH_TOKEN=<tokenvalue>

Content-Type:application/json

`Request Payload:`

```
{

"aaaglobal_aaapreauthenticationpolicy_binding":

    {

    "policy":"preautpol1",
```

```
    "priority":"200"

    }

}
```

- `Response`

  `HTTP Status Code on Success:` 200 OK

`HTTP Status Code on Failure:`

4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix-ADC-specific errors). The response payload provides details of the error.

The following example unbinds the policy named preautpol1.

- `Request`

  `HTTP Method:` DELETE

`URL:`

http://<Citrix-ADC-ip-address(NSIP)\>/nitro/v1/config/aaaglobal_aaapreauthenticationpolicy_binding?args=policy:preautpol1

`Request Header:`

Cookie:NITRO_AUTH_TOKEN=\<tokenvalue\>

- `Response`

  `HTTP Status Code on Success:` 200 OK

`HTTP Status Code on Failure:`

4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix-ADC-specific errors). The response payload provides details of the error.

- `Using REST APIs through SDKs`

  The following sample code creates a preauthentication action and a preauthentication policy that uses that action, and then binds the policy globally at priority 200.

Java - Sample code

```
aaapreauthenticationaction preauth_act1;

aaapreauthenticationpolicy preauth_pol1;

aaaglobal_aaapreauthenticationpolicy_binding glob_binding;

preauth_act1 = new aaapreauthenticationaction();

preauth_act1.set_name("preauth_act1");

preauth_act1.set_preauthenticationaction("ALLOW");

aaapreauthenticationaction.add(ns_session,preauth_act1);

preauth_pol1 = new aaapreauthenticationpolicy();

preauth_pol1.set_name("preauth_pol1");

preauth_pol1.set_rule("CLIENT.APPLICATION.PROCESS(antivirus.exe) EXISTS");

preauth_pol1.set_reqaction("preauth_act1");
```

```
aaapreauthenticationpolicy.add(ns_session,preauth_pol1);

 glob_binding = new aaaglobal_aaapreauthenticationpolicy_binding();

glob_binding.set_policy("preauth_pol1");

glob_binding.set_priority(200);

aaaglobal_aaapreauthenticationpolicy_binding.add(ns_session,glob_binding);
```

`.NET - Sample code`

```
aaapreauthenticationaction preauth_act1;

aaapreauthenticationpolicy preauth_pol1;

aaaglobal_aaapreauthenticationpolicy_binding glob_binding;

preauth_act1 = new aaapreauthenticationaction();

preauth_act1.name = "preauth_act1";
```

```
preauth_act1.preauthenticationaction = "ALLOW";

aaapreauthenticationaction.add(ns_session, preauth_act1);

preauth_pol1 = new aaapreauthenticationpolicy();

preauth_pol1.name = "preauth_pol1";

preauth_pol1.rule = "CLIENT.APPLICATION.PROCESS(antivirus.exe) EXISTS";

preauth_pol1.reqaction = "preauth_act1";

aaapreauthenticationpolicy.add(ns_session, preauth_pol1);

glob_binding = new aaaglobal_aaapreauthenticationpolicy_binding();

glob_binding.policy = "preauth_pol1";

glob_binding.priority = 200;

aaaglobal_aaapreauthenticationpolicy_binding.add(ns_session,glob_binding);
```

Python - Sample code

```
preauth_act1 = aaapreauthenticationaction()

preauth_act1.name = "preauth_act1"

preauth_act1.preauthenticationaction = "ALLOW"

aaapreauthenticationaction.add(ns_session, preauth_act1)

preauth_pol1 = aaapreauthenticationpolicy()

preauth_pol1.name = "preauth_pol1"

preauth_pol1.rule = "CLIENT.APPLICATION.PROCESS(antivirus.exe) EXISTS"

preauth_pol1.reqaction = "preauth_act1"

aaapreauthenticationpolicy.add(ns_session, preauth_pol1)

glob_binding = aaaglobal_aaapreauthenticationpolicy_binding()

glob_binding.policy = "preauth_pol1"
```

```
glob_binding.priority = 200


aaaglobal_aaapreauthenticationpolicy_binding.add(ns_session, glob_binding)
```

Deleting a Citrix ADC Resource

The usage of the delete operation depends on the unique identifiers (UIDs) of the resource being deleted.

Deleting Resource with Single UID. To delete a Citrix ADC resource that can be identified by a single identifier, specify the resource name. Deleting Resource with Multiple UIDs. To delete a Citrix ADC resource that is identified using multiple identifiers, use the "args" query parameter to specify the unique attributes along with their values.

- `Using REST APIs through HTTP`

  The following example deletes a load balancing virtual server named MyFirstLbVServer.

- `Request`

  `HTTP Method:` DELETE

`URL:` http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/lbvserver/MyFirstLbVServer

`Request Header:` Cookie:NITRO_AUTH_TOKEN=\<tokenvalue\>

- `Response`

  `HTTP Status Code on Success:` 200 OK

`HTTP Status Code on Failure:`

4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix-ADC-specific errors). The response payload provides details of the error.

The following example deletes a resource with multiple UIDs. Consider a SNIP (10.102.29.71) that belongs to two traffic domains (TDs) 123 and 110. To delete the SNIP from one of the traffic domains, specify the IP address and the relevant TD in the URL as follows:

- `Request`

  `HTTP Method:` DELETE

`URL:`

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/nsip?args=ipaddress:10.102.29.71,td:123

`Request Header:` Cookie:NITRO_AUTH_TOKEN=\<tokenvalue\>

- `Response`

  `HTTP Status Code on Success:` 200 OK

`HTTP Status Code on Failure:`

4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix-ADC-specific errors). The response payload provides details of the error.

- `Using REST APIs through SDKs`

  To delete an existing resource, invoke the static method delete() on the resource class, by passing the name of the resource.

The following sample code deletes a load balancing virtual server with name MyFirstLbVServer.

`Java - Sample code to delete a Citrix ADC resource`

```
lbvserver remove_lb = new lbvserver();

remove_lb.set_name("MyFirstLbVServer");

lbvserver.delete(ns_session, remove_lb);
```

`.NET - Sample code to delete a Citrix ADC resource`

```
lbvserver remove_lb = new lbvserver();

remove_lb.name("MyFirstLbVServer");

lbvserver.delete(ns_session, remove_lb);
```

`Python - Sample code to delete a Citrix ADC resource`

```
remove_lb = lbvserver()

remove_lb.name = "MyFirstLbVServer"

lbvserver.delete(ns_session, remove_lb)
```

Performing Bulk Operations

You can create, retrieve, update, and delete multiple resources simultaneously and thus minimize network traffic. For example, you can add multiple load balancing virtual servers in the same operation.

To account for the failure of some operations within the bulk operation, NITRO allows you to configure one of the following behaviors while establishing a connection with the appliance.

- `Exit`. When the first error is encountered, the execution stops. The commands that were executed before the error are committed.
- `Rollback`. When the first error is encountered, the execution stops. The commands that were executed before the error are rolled back. Rollback is only supported for add and bind commands.
- `Continue`. All the commands in the list are executed even if some commands fail.
- `Using REST APIs through HTTP`

  To perform a bulk operation, specify the required parameters in the same request payload. You must specify the behavior of the bulk operation in the request header using the X-NITRO-ONERROR parameter.

The following example adds two load balancing virtual servers in one operation. The operation continues even if one of the add operation fails.

- `Request`

  `HTTP Method:` POST

`URL:` http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/lbvserver

`Request Headers:`

Cookie:NITRO_AUTH_TOKEN=<tokenvalue>

Content-Type:application/json

`Request Payload:`

```
{

    "lbvserver":

    [

        {

        "name":"new_lbvserver1",

        "servicetype":"http"

        },



        {

        "name":"new_lbvserver2",

        "servicetype":"http"

        }

    ]

}
```

- `Response`

  `HTTP Status Code on Success:` 201 Created for the add operation and 200 OK for the update operation.

`HTTP Status Code on Failure:` 207 Multi Status with error details in the response payload.

The following example enables multiple load balancing virtual servers in the same operation.

- `Request`

  `HTTP Method:` POST

`URL:` http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/lbvserver?action=enable

`Request Headers:`

Cookie:NITRO_AUTH_TOKEN=\<tokenvalue\>

Content-Type:application/json

`Request Payload:`

```
{
            "lbvserver":
    [
        {
            "name":"v1",
        },


        {
            "name":"v2",
        }
    ]
}
```

- `Response`

  `HTTP Status Code on Success:` 201 Created for the add operation and 200 OK for the update operation.

`HTTP Status Code on Failure:` 207 Multi Status with error details in the response payload.

- `Using REST APIs through SDKs`

To perform a bulk operation, you instantiate an array of the resource class, configure the properties of all the instances locally, and then upload all the instances to the Citrix ADC with one command.

The following sample code specifies bulk operation behavior.

`Java - Sample code to specify bulk operation behavior`

```
nitro_service ns_session = new nitro_service("10.102.29.60","http");

ns_session.set_onerror(OnerrorEnum.CONTINUE);

ns_session.login("admin","verysecret");
```

`.NET - Sample code to specify bulk operation behavior`

```
nitro_service ns_session = new nitro_service("10.102.29.60","http");

ns_session.onerror = OnerrorEnum.CONTINUE;

ns_session.login("admin","verysecret");
```

`Python - Sample code to specify bulk operation behavior`

```
ns_session = nitro_service("10.102.29.60","http")

ns_session.onerror = OnerrorEnum.CONTINUE

ns_session.login("admin","verysecret")
```

The following sample code creates two load balancing virtual servers.

`Java - Sample code for bulk creation`

```
//Create an array of lbvserver instances



lbvserver[] lbs = new lbvserver[2];
```

```
//Specify properties of the first lbvserver


lbs[0] = new lbvserver();

lbs[0].set_name("lbvserv1");

lbs[0].set_servicetype("http");

lbs[0].set_ipv46("10.70.136.5");

lbs[0].set_port(80);


//Specify properties of the second lbvserver


lbs[1] = new lbvserver();

lbs[1].set_name("lbvserv2");

lbs[1].set_servicetype("https");

lbs[1].set_ipv46("10.70.136.5");

lbs[1].set_port(443);


//Upload the properties of the two lbvservers to the Citrix ADC


lbvserver.add(ns_session,lbs);
```

`.NET - Sample code for bulk creation`

```
//Create an array of lbvserver instances

lbvserver[] lbs = new lbvserver[2];



//Specify details of first lbvserver



lbs[0] = new lbvserver();

lbs[0].name = "lbvserv1";

lbs[0].servicetype = "http";

lbs[0].ipv46 = "10.70.136.5";

lbs[0].port = 80;



//Specify details of second lbvserver



lbs[1] = new lbvserver();

lbs[1].name = "lbvserv2";

lbs[1].servicetype = "https";

lbs[1].ipv46 = "10.70.136.5";

lbs[1].port = 443;



//upload the details of the lbvservers to the NITRO server
```

```
lbvserver.add(ns_session,lbs);
```

`Python - Sample code for bulk creation`

```
# Create an array of lbvserver instances

lbs = lbvserver[2]

# Specify properties of the first lbvserver

lbs[0] = lbvserver()
lbs[0].name = "lbvserv1"
lbs[0].servicetype = "http"
lbs[0].ipv46 = "10.70.136.5"
lbs[0].port = 80

# Specify properties of the second lbvserver

lbs[1] = lbvserver()
lbs[1].name = "lbvserv2"
lbs[1].servicetype = "https"
lbs[1].ipv46 = "10.70.136.5"
```

```
lbs[1].port = 443



# Upload the properties of the two lbvservers to the CItrix ADC



lbvserver.add(ns_session, lbs)
```

Getting the Count of Citrix ADC Resources

This topic covers getting the count of a type of Citrix ADC resource present in a Citrix ADC appliance by using REST APIs through HTTP.

- `Using REST APIs through HTTP`

  To get a count of a specific resource type, in the URL specify the count query parameter as "yes".

For example, to get a count of all the load balancing virtual servers.

- `Request`

  `HTTP Method:` GET

`URL:` http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/lbvserver?count=yes

`Request Headers:`

Cookie:NITRO_AUTH_TOKEN=<tokenvalue>

Accept:application/json

- `Response`

  `HTTP Status Code on Success:` 200 OK

`HTTP Status Code on Failure:`

4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix-ADC-specific errors). The response payload provides details of the error.

`Response Header:`

Content-Type:application/json

`Response Payload:`

```
{

    "lbvserver":

    [

        {

            "__count": 4

        }

    ]

}
```

Renaming a Citrix ADC Resource

This topic covers renaming a Citrix ADC resource by using REST APIs through HTTP.

- `Using REST APIs through HTTP`

  To change the name of an existing resource, specify the action as "rename" in the URL query string, and in the request payload, specify the existing name and the new name.

For example, to change the name of a load balancing virtual server from MyFirstLbVServer to MyServer:

- `Request`

  `HTTP Method:` POST

`URL:` http://<Citrix-ADC-ip-address(NSIP)\>/nitro/v1/config/lbvserver?action=rename

`Request Headers:`

Cookie:NITRO_AUTH_TOKEN=<tokenvalue>

Content-Type:application/json

`Request Payload:`

```
{
    "lbvserver":
    {
        "name":"MyFirstLbVServer",
        "newname":"MyServer"
    }
}
```

- `Response`

  `HTTP Status Code on Success:` 200 OK

`HTTP Status Code on Failure:`

4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix-ADC-specific errors). The response payload provides details of the error.

Performing File Operations

Citrix ADC operations such as configuring SSL certificates requires the input files to be available locally on the Citrix ADC appliance. NITRO allows you to perform file operations such as uploading files, retrieving files, retrieving file content, and deleting files of types: txt, cert, req, xml, lic, and key.

- `Notes`
  - File size must be less than or equal to 2 MB.
- Use the "BASE64" value for the fileencoding attribute in the request payload. This is the only valid encoding currently supported.
- The filelocation path must be URL encoded. For example, if the path is /nsconfig/ssl, encode the / and use the file location as %2Fnsconfig%2Fssl.

- When uploading a file, make sure that each directory of the file path has the 755 (read, write, execute) permission. For example, to upload a file to the "/nsconfig/ssl.html" directory, the following directories must have the 755 permission: flash (because the "/nsconfig" folder is actually a link to "/flash/nsconfig.html" directory)
    - nsconfig
    - ssl

Uploading a File

To upload a file to the Citrix ADC, specify a name for the file, the location where the file must be created on the Citrix ADC, and the content of the file.

- `Using REST APIs through HTTP`

  For example, to upload a file named cert1.crt in the /nsconfig/ssl/ directory:

- `Request`

  `HTTP Method:` POST

`URL:` http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/systemfile

`Request Headers:`

Cookie:NITRO_AUTH_TOKEN=<tokenvalue>

Content-Type:application/json

`Request Payload:`

```
{

        "systemfile":

        {

            "filename": "cert1.crt",

            "filelocation": "/nsconfig/ssl.html",

            "filecontent":"VGhpcyBpcyBteSBmaWxl",

            "fileencoding": "BASE64"
```

```
        }

}
```

- `Response`

  `HTTP Status Code on Success:` 200 OK

`HTTP Status Code on Failure:`

4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix-ADC-specific errors). The response payload provides details of the error.

Retrieving the Files

To retrieve the files from a specific Citrix ADC directory, specify the directory path in the URL.

- `Using REST APIs through HTTP`

  For example, to retrieve the files from the /nsconfig/ssl directory.

- `Request`

  `HTTP Method:` GET

`URL:`

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/systemfile?args=filelocation:%2Fnsconfig%2Fssl

`Request Header:`

Cookie:NITRO_AUTH_TOKEN=\<tokenvalue\>

Accept:application/json

- `Response`

  `HTTP Status Code on Success:` 200 OK

`HTTP Status Code on Failure:`

4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix-ADC-specific errors). The response payload provides details of the error.

`Response Header:`

Content-Type:application/json

`Response Payload:`

```
{

    "systemfile":

    [

            "filename": "ns-root.key",

            "filelocation": "/nsconfig/ssl",

            "fileaccesstime": "Tue Jan 14 19:27:01 2014",

            "filemodifiedtime": "Tue Nov  5 17:16:00 2013"

        },

        {

        {

            "filename": "ns-root.req",

            "filelocation": "/nsconfig/ssl",

            "fileaccesstime": "Tue Jan 14 19:27:01 2014",

            "filemodifiedtime": "Tue Nov  5 17:16:00 2013"

        }

    ]

}
```

Retrieving Contents of a Specific File

To retrieve the contents of a file, specify the filename and its directory path in the URL.

- `Using REST APIs through HTTP`

  For example, to retrieve the contents of the ns-root.key file from the /nsconfig/ssl directory.

- `Request`

  `HTTP Method:` GET

`URL:` http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/systemfile/ns-root.key?args=filelocation:%2Fnsconfig%2Fssl

`Request Header:`

Cookie:NITRO_AUTH_TOKEN=\<tokenvalue\>

Accept:application/json

- `Response`

  `HTTP Status Code on Success:`

200 OK

`HTTP Status Code on Failure:`

4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error.

`Response Header:`

Content-Type:application/json

`Response Payload:`

```
{

    "systemfile":

    [

        {

            "filename": "ns-root.key",

            "filelocation": "/nsconfig/ssl",
```

```
            "filecontent": "LS0tLS1CRUdJTiBSU0EgUFJJVkFU0tLQo=",

            "fileencoding": "BASE64",

            "fileaccesstime": "Tue Jan 14 19:27:01 2014",

            "filemodifiedtime": "Tue Nov  5 17:16:00 2013"

        }

    ]

}
```


Deleting a File

To delete a file from the Citrix ADC appliance, specify the filename and the directory path in the URL.

- `Using REST APIs through HTTP`

  For example, to delete the ns-root.key file from the /nsconfig/ssl directory.

`Request:`

`HTTP Method:` DELETE

`URL:` https://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/systemfile/ns-root.key?args=filelocation:%2Fnsconfig%2Fssl

`Request Header:`

Cookie:NITRO_AUTH_TOKEN=\<tokenvalue\>

- `Response`

  `HTTP Status Code on Success:` 200 OK

`HTTP Status Code on Failure:`

4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix-ADC-specific errors). The response payload provides details of the error.

Use Cases and Samples

This section covers some use case scenarios to certain resources and scenarios. More scenarios will be added in future updates to this section.

Managing a Citrix ADC Cluster

For managing clusters, you can add or remove a cluster instance or an individual node and perform a few other instance or node operations such as viewing instance or node properties. You can also configure the cluster IP address. Other cluster-management tasks include joining a Citrix ADC appliance to the cluster and configuring a linkset. For detailed information and best practices, see Clustering.

Cluster Instance Operations

This topic covers cluster instance operations by using REST APIs through HTTP or SDKs.

- `Using REST APIs through HTTP`

  All operations on a cluster instance must be performed on the clusterinstance object.

For example, to create a cluster instance with ID 1, connect to the Citrix ADC appliance that you are first adding to the cluster.

- `Request`

  `HTTP Method:` POST

`URL:` http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/clusterinstance

`Request Headers:`

Cookie:NITRO_AUTH_TOKEN=<tokenvalue>

Content-Type:application/json

`Request Payload:`

```
{

    "clusterinstance":

    {

    "clid":1,

    "preemption":"ENABLED"

    }

}
```

- `Response`

  `HTTP Status Code on Success:` 201 Created

`HTTP Status Code on Failure:` 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix-ADC-specific errors). The response payload provides details of the error.

- `Using REST APIs through SDKs`

  The com.citrix.netscaler.nitro.resource.config.cluster.clusterinstance class provides APIs to manage a cluster instance.

The following sample code creates a cluster instance with ID 1.

`Java - Sample code to create a cluster instance`

```
clusterinstance new_cl_inst_obj = new clusterinstance();



//Set the properties of the cluster instance locally



new_cl_inst_obj.set_clid(1);

new_cl_inst_obj.set_preemption("ENABLED");
```

```
//Upload the cluster instance



clusterinstance.add(ns_session,new_cl_inst_obj);
```

`.NET - Sample code to create a cluster instance`

```
clusterinstance new_cl_inst_obj = new clusterinstance();



//Set the properties of the cluster instance locally



new_cl_inst_obj.clid = 1;



new_cl_inst_obj.preemption = "ENABLED";



//Upload the cluster instance



clusterinstance.add(ns_session,new_cl_inst_obj);
```

`Python - Sample code to create a cluster instance`

```
new_cl_inst_obj = clusterinstance()



# Set the properties of the cluster instance locally
```

```
new_cl_inst_obj.clid = 1



# Upload the cluster instance



clusterinstance.add(ns_session, new_cl_inst_obj)
```

Cluster Node Operations

This topic covers cluster node operations by using REST APIs through HTTP or SDKs.

- `Using REST APIs through HTTP`

  All operations on a cluster node must be performed on the clusternode object. For example, to add a Citrix ADC appliance with NSIP address 10.102.29.60 to the cluster.

- `Request`

  `HTTP Method:` POST

`URL:` http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/clusternode

`Request Headers:`

Cookie:NITRO_AUTH_TOKEN=<tokenvalue>

Content-Type:application/json

`Request Payload:`

```
{

"clusternode":

    {

    "nodeid":1,

    "ipaddress":"10.102.29.60",
```

```
    "state":"ACTIVE",

    "backplane":"1/1/2"

    }

}
```

- `Response`

  `HTTP Status Code on Success:` 201 Created

`HTTP Status Code on Failure:` 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix-ADC-specific errors). The response payload provides details of the error.

- `Using REST APIs through SDKs`

The com.citrix.netscaler.nitro.resource.config.cluster.clusternode class provides APIs to manage cluster nodes.

The following sample code adds a cluster node with NSIP address 10.102.29.60.

`Java - Sample code to add a cluster node`

```
clusternode new_cl_node_obj = new clusternode();



//Set the properties of the cluster node locally



new_cl_node_obj.set_nodeid(0);



new_cl_node_obj.set_ipaddress("10.102.29.60");



new_cl_node_obj.set_state("ACTIVE");
```

```
new_cl_node_obj.set_backplane("0/1/1");


//Upload the cluster node


clusternode.add(ns_session,new_cl_node_obj);
```

`.NET - Sample code to add a cluster node`

```
clusternode new_cl_node_obj = new clusternode();


//Set the properties of the cluster node locally


new_cl_node_obj.nodeid = 0;


new_cl_node_obj.ipaddress = "10.102.29.60";


new_cl_node_obj.state = "ACTIVE";


new_cl_node_obj.backplane = "0/1/1";


//Upload the cluster node


clusternode.add(ns_session,new_cl_node_obj);
```

`Python - Sample code to add a cluster node`

```
new_cl_node_obj = clusternode()



# Set the properties of the cluster node locally



new_cl_node_obj.nodeid = 0



new_cl_node_obj.ipaddress = "10.102.29.60"



new_cl_node_obj.state = "ACTIVE"



new_cl_node_obj.backplane = "0/1/1"



# Upload the cluster node



clusternode.add(ns_session, new_cl_node_obj)
```

Add a Cluster IP Address

This topic covers adding a cluster IP address by using REST APIs through HTTP or SDKs.

- `Using REST APIs through HTTP`

  To define a cluster IP address, specify the required parameters in the nsip object. For example, to configure a cluster IP address.

- `Request`

`HTTP Method:` POST

`URL:` http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/nsip

`Request Headers:`

Cookie:NITRO_AUTH_TOKEN=<tokenvalue>

Content-Type:application/json

`Request Payload:`

```
{

"nsip":

    {

    "ipaddress":"10.102.29.61",

    "netmask":"255.255.255.255",

    "type":"CLIP"

    }

}
```

- `Response`

  `HTTP Status Code on Success:` 201 Created

`HTTP Status Code on Failure:` 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix-ADC-specific errors). The response payload provides details of the error.

- `Using REST APIs through SDKs`

  The com.citrix.netscaler.nitro.resource.config.ns.nsip class provides the add() API to configure an IP address. To configure the IP address as a cluster IP address, you must specify the type as CLIP.

The following sample code configures a cluster IP address on Citrix ADC appliance with IP address 10.102.29.60.

`Java - Sample code to add a cluster IP address`

```
nsip new_nsip_obj = new nsip();

//Set the properties locally

new_nsip_obj.set_ipaddress("10.102.29.61");

new_nsip_obj.set_netmask("255.255.255.255");

new_nsip_obj.set_type("CLIP");

//Upload the cluster node

nsip.add(ns_session,new_nsip_obj);
```

`.NET - Sample code to add a cluster IP address`

```
nsip new_nsip_obj = new nsip();

//Set the properties locally

new_nsip_obj.ipaddress = "10.102.29.61";
```

```
new_nsip_obj.netmask = "255.255.255.255";


new_nsip_obj.type = "CLIP";


//Upload the cluster node


nsip.add(ns_session,new_nsip_obj);
```

`Python - Sample code to add a cluster IP address`

```
new_nsip_obj = nsip()


# Set the properties locally


new_nsip_obj.ipaddress = "10.102.29.61"


new_nsip_obj.netmask = "255.255.255.255"


new_nsip_obj.type = "CLIP"


# Upload the cluster node


nsip.add(ns_session, new_nsip_obj)
```

Add a Spotted IP Address

This topic covers adding a spotted IP address by using REST APIs through HTTP or SDKs.

- `Using REST APIs through HTTP`

  To configure an IP address as spotted, specify the required parameters in the nsip object. This configuration must be done on the cluster IP address.

For example, to configure a spotted SNIP address on a node with ID 1.

- `Request`

  `HTTP Method:` POST

`URL:` http://<cluster-ip-address\>/nitro/v1/config/nsip

`Request Headers:`

Cookie:NITRO_AUTH_TOKEN=<tokenvalue>

Content-Type:application/json

`Request Payload:`

```
{

"nsip":

    {

    "ipaddress":"10.102.29.77",

    "netmask":"255.255.255.0",

    "type":"SNIP",

    "ownernode":1

    }

}
```

- `Response`

`HTTP Status Code on Success:` 201 Created

`HTTP Status Code on Failure:` 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix-ADC-specific errors). The response payload provides details of the error.

- `Using REST APIs through SDKs`

  The com.citrix.netscaler.nitro.resource.config.ns.nsip class provides the add() API to configure an IP address. To configure the IP address as spotted, you must specify the ID of the node that must own the IP address. This configuration must be done on the cluster IP address.

The following sample code configures a spotted SNIP address on a node with ID 1.

`Java - Sample code to configure a spotted IP address`

```
nsip new_nsip_obj = new nsip();



//Set the properties locally



new_nsip_obj.set_ipaddress("10.102.29.77");



new_nsip_obj.set_netmask("255.255.255.0");



new_nsip_obj.set_type("SNIP");



new_nsip_obj.set_ownernode(1);



//Upload the cluster node
```

```
nsip.add(ns_session,new_nsip_obj);
```

`.NET - Sample code to configure a spotted IP address`

```
nsip new_nsip_obj = new nsip();


//Set the properties locally


new_nsip_obj.ipaddress = "10.102.29.77";


new_nsip_obj.netmask = "255.255.255.0";


new_nsip_obj.type = "SNIP";


new_nsip_obj.ownernode = 1;


//Upload the cluster node


nsip.add(ns_session,new_nsip_obj);
```

`Python - Sample code to configure a spotted IP address`

```


# Add a spotted IP address


```

```
new_nsip_obj = nsip()


# Set the properties locally


new_nsip_obj.ipaddress = "10.102.29.77"


new_nsip_obj.netmask = "255.255.255.0"


new_nsip_obj.type = "SNIP"


new_nsip_obj.ownernode = 1


# Upload the cluster node


nsip.add(ns_session, new_nsip_obj)
```

Join Citrix ADC Appliance to Cluster

This topic covers adding a Citrix ADC appliance to a cluster by using REST APIs through HTTP or SDKs.

- `Using REST APIs through HTTP`

  To join an appliance to a cluster, specify the required parameters in the cluster object. For example, to join a Citrix ADC appliance to a cluster.

- `Request`

  `HTTP Method:` POST

`URL:` http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/cluster

`Request Headers:`

Cookie:NITRO_AUTH_TOKEN=<tokenvalue>

Content-Type:application/json

`Request Payload:`

```
{

"cluster":

    {

    "clip":"10.102.29.61",

    "password":"verysecret"

    }

}
```

- `Response`

  `HTTP Status Code on Success:` 201 Created

`HTTP Status Code on Failure:` 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix-ADC-specific errors). The response payload provides details of the error.

- `Using REST APIs through SDKs`

  The com.citrix.netscaler.nitro.resource.config.cluster.cluster class provides the join() API to join a Citrix ADC appliance to the cluster. You must specify the cluster IP address and the nsroot password of the configuration coordinator.

The following sample code joins a Citrix ADC appliance to a cluster.

`Java - Sample code to join an appliance to a cluster`

```
cluster new_cl_obj = new cluster();


//Set the properties of the cluster  locally


new_cl_obj.set_clip("10.102.29.61");


new_cl_obj.set_password("verysecret");


//Upload the cluster


cluster.add(ns_session,new_cl_obj);
```

`.NET - Sample code to join an appliance to a cluster`

```
cluster new_cl_obj = new cluster();


//Set the properties of the cluster locally


new_cl_obj.clip = "10.102.29.61";


new_cl_obj.password = "verysecret";


//Upload the cluster node
```

```
cluster.add(ns_session,new_cl_node_obj);
```

`Python - Sample code to join an appliance to a cluster`

```
new_cl_obj = cluster()

# Set the properties of the cluster locally

new_cl_obj.clip = "10.102.29.61"

new_cl_obj.password = "verysecret"

# Upload the cluster

cluster.add(ns_session, new_cl_obj)
```

Linkset Operations

This topic covers some linkset operations by using REST APIs through HTTP or SDKs.

- `Using REST APIs through HTTP`

  To configure a linkset, do the following:

Create a linkset by specifying the required parameters in the linkset object. For example, to add a linkset LS/1:

`Request:`

`HTTP Method:` POST

`URL:` http://<cluster-ip-address\>/nitro/v1/config/linkset

`Request Headers:`

Cookie:NITRO_AUTH_TOKEN=<tokenvalue>

Content-Type:application/json

- `Request Payload`

```
{ "linkset": { "id":"LS/1" } }
```

```
-   `Response`

    `HTTP Status Code on Success:` 201 Created

`HTTP Status Code on Failure:` 4xx &lt;string&gt; (for general HTTP
errors) or 5xx &lt;string&gt; (for Citrix-ADC-specific errors). The
response payload provides details of the error.

Bind the required interfaces to the linkset by specifying the interfaces
in the linkset_interface_binding object.For example, to bind interfaces
1/1/2 and 2/1/2 to linkset LS/1.

-   `Request`

    `HTTP Method:` PUT
```

```
`URL:` http://\&lt;cluster-ip-address\>/nitro/v1/config/
linkset_interface_binding/LS%2F1?action=bind

`Note.` The linkset name (LS/1), must be URL encoded as LS%2F1.



`Request Headers:`



Cookie:NITRO_AUTH_TOKEN=\&lt;tokenvalue>



Content-Type:application/json



`Request Payload:`



```json

{

"linkset_interface_binding":

    {

    "id":"LS/1",

    "ifnum":"1/1/2 2/1/2"

    }

}
```

- `Response`

  `HTTP Status Code on Success:` 200 Ok
```

`HTTP Status Code on Failure:` 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix-ADC-specific errors). The response payload provides details of the error.

- `Using REST APIs through SDKs`

  The com.citrix.netscaler.nitro.resource.config.network.linkset class provides the APIs to manage linksets.

To configure a linkset, do the following:

1. Add a linkset by invoking the add() method of the linkset class.
2. Bind the interfaces to the linkset using the add() method of the linkset_interface_binding class.

The following sample code creates a linkset LS/1 and bind interfaces 1/1/2 and 2/1/2 to it.

`Java - Sample code to configure linksets`

```
linkset new_linkset_obj = new linkset();



new_linkset_obj.set_id("LS/1");



linkset.add(ns_session,new_linkset_obj);



//Bind the interfaces to the linkset



linkset_interface_binding new_linkif_obj = new
linkset_interface_binding();



new_linkif_obj.set_id("LS/1");



new_linkif_obj.set_ifnum("1/1/2 2/1/2");
```

```
linkset_interface_binding.add(ns_session,new_linkif_obj);
```

`.NET - Sample code to configure linksets`

```
linkset new_linkset_obj = new linkset();

new_linkset_obj.id = "LS/1";

linkset.add(ns_session,new_linkset_obj);

//Bind the interfaces to the linkset

linkset_interface_binding new_linkif_obj = new
linkset_interface_binding();

new_linkif_obj.id = "LS/1";

new_linkif_obj.ifnum = "1/1/2 2/1/2";

linkset_interface_binding.add(ns_session,new_linkif_obj);
```

`Python - Sample code to configure linksets`

```
# Create a new linkset

new_linkset_obj = linkset()

new_linkset_obj.id = "LS/1"

linkset.add(ns_session, new_linkset_obj)

# Bind the interfaces to the linkset

new_linkif_obj = linkset_interface_binding()

new_linkif_obj.id = "LS/1"

new_linkif_obj.ifnum = "1/1/2 2/1/2"

linkset_interface_binding.add(ns_session, new_linkif_obj)
```

Configuring Admin Partitions

To create an admin partition, you must perform a set of operations on the default partition. To understand this procedure, let us consider a company that has two departments each of which has an application that requires the Citrix ADC functionality. The Citrix ADC admin wants to have a

different partition for each department so that there is isolation of users and configurations. For detailed information and best practices, see Admin Partitions.

Creating an Admin Partition

While creating an admin partition, you must also specify the system resources that must be allocated to that partition.

- `Using REST APIs through HTTP`

  The following example creates an admin partition named partition-dept1.

- `Request`

  `HTTP Method:` POST

`URL:` http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/nspartition

`Request Headers:`

Cookie:NITRO_AUTH_TOKEN=<tokenvalue>

Content-Type:application/json

`Request Payload:`

```
{

"nspartition":

{

    "partitionname":"partition-dept1",

    "maxbandwidth":"10240",

    "minbandwidth":"10240",

    "maxconn":"1024",

    "maxmemlimit":"10"

}
```

```
}
```

- `Response`

  `HTTP Status Code on Success:` 201 Created

`HTTP Status Code on Failure:` 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix-ADC-specific errors). The response payload provides details of the error.

- `Using REST APIs through SDKs`

  The following sample code creates an admin partition named partition-dept1.

`Java - Sample code to create an admin partition`

```
nspartition nspartitionObject = new nspartition();



nspartitionObject.set_partitionname("partition-dept1");



nspartitionObject.set_maxbandwidth(10240);



nspartitionObject.set_maxconn(1024);



nspartitionObject.set_maxmemlimit(10);



nspartitionObject.set_minbandwidth(1240);



base_response result = nspartition.add(nitroService, nspartitionObject);
```

`.NET - Sample code to create an admin partition`

```
nspartition nspartitionObject = new nspartition();


nspartitionObject.partitionname = "partition-dept1";


nspartitionObject.maxbandwidth = 10240;


nspartitionObject.maxconn = 1024;


nspartitionObject.maxmemlimit = 10;


nspartitionObject.minbandwidth = 1240;


base_response result = nspartition.add(nitroService, nspartitionObject);
```

`Python - Sample code to create an admin partition`

```
nspartitionObject = nspartition()


nspartitionObject.partitionname = "partition-dept1"


nspartitionObject.maxbandwidth = 10240


nspartitionObject.maxconn = 1024
```

```
nspartitionObject.maxmemlimit = 10


nspartitionObject.minbandwidth = 1240


result = nspartition.add(nitroService, nspartitionObject)
```

Associating Users with Partitions

Associate the appropriate users with the partition.

- `Using REST APIs through HTTP`

  The following example associates user1 to a partition named partition-dept1.

- `Request`

  `HTTP Method:` PUT

`URL:` http://<Citrix-ADC-ip-address(NSIP)\>/nitro/v1/config/systemuser_nspartition_binding/user1

`Request Headers:`

Cookie:NITRO_AUTH_TOKEN=<tokenvalue>

Content-Type:application/json

`Request Payload:`

```
{

    "systemuser_nspartition_binding":

    {

    "username":"user1",

    "partitionname":"partition-dept1"
```

```
    }

}
```

- `Response`

  `HTTP Status Code on Success:` 201 Created

`HTTP Status Code on Failure:` 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix-ADC-specific errors). The response payload provides details of the error.

- `Using REST APIs through SDKs`

  The following sample code associates "user1" to a partition named "partition-dept1".

`Java - Sample code for associating user with partition`

```
systemuser_nspartition_binding systemuser_nspartition_binding_object = new
systemuser_nspartition_binding();



systemuser_nspartition_binding_object.set_partitionname("partition-
dept1");



systemuser_nspartition_binding_object.set_username("user1");



base_response result = systemuser_nspartition_binding.add(nitroService,
systemuser_nspartition_binding_object);
```

`.NET - Sample code for associating user with partition`

```
systemuser_nspartition_binding systemuser_nspartition_binding_object = new
systemuser_nspartition_binding();



systemuser_nspartition_binding_object.partitionname = "partition-dept1";
```

```
systemuser_nspartition_binding_object.username = "user1";


base_response result = systemuser_nspartition_binding.add(nitroService,
systemuser_nspartition_binding_object);
```

`Python - Sample code for associating user with partition`

```
systemuser_nspartition_binding_object =  systemuser_nspartition_binding()


systemuser_nspartition_binding_object.partitionname = "partition-dept1"


systemuser_nspartition_binding_object.username = "user1"


result = systemuser_nspartition_binding.add(nitroService,
systemuser_nspartition_binding_object)
```

Specifying Command Policy for Partition Users

Associate an appropriate command policy to the admin partition user.

- `Using REST APIs through HTTP`

  The following example associates the command policy partition-admin to user1.

- `Request`

  `HTTP Method:` PUT

`URL:` http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/systemuser_systemcmdpolicy_binding/user1

`Request Headers:`

Cookie:NITRO_AUTH_TOKEN=<tokenvalue>

Content-Type:application/json

`Request Payload:`

```
{

    "systemuser_systemcmdpolicy_binding":

    {

    "username":"user1",

    "policyname":"partition-admin",

    "priority":"1"

    }

}
```

- `Response`

  `HTTP Status Code on Success:` 201 Created

`HTTP Status Code on Failure:` 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix-ADC-specific errors). The response payload provides details of the error.

- `Using REST APIs through SDKs`

  The following sample code associates the command policy "partition-admin" to "user1".

`Java - Sample code to specify command policy for partition user`

```
systemuser_systemcmdpolicy_binding binding_object = new
systemuser_systemcmdpolicy_binding();



binding_object.set_username("user1");
```

```
binding_object.set_policyname("partition-admin");



binding_object.set_priority(1);



base_response result =
systemuser_systemcmdpolicy_binding.add(nitroService,binding_object);
```

`.NET - Sample code to specify command policy for partition user`

```
systemuser_systemcmdpolicy_binding binding_object = new
systemuser_systemcmdpolicy_binding();



binding_object.username = "user1";



binding_object.policyname = "partition-admin";



binding_object.priority = 1;



base_response result =
systemuser_systemcmdpolicy_binding.add(nitroService,binding_object);
```

`Python - Sample code to specify command policy for partition user`

```
binding_object = systemuser_systemcmdpolicy_binding()



binding_object.username = "user1"
```

```
binding_object.policyname = "partition-admin"



binding_object.priority = 1



result =
systemuser_systemcmdpolicy_binding.add(nitroService,binding_object)
```

Specifying the Admin Partition VLAN or Bridgegroup

Specify the VLANs or bridgegroups to be associated with the partition. This step ensures network isolation of the traffic. Traffic received on the interfaces of the VLAN or bridgegroup is isolated from the traffic of other partitions.

- `Using REST APIs through HTTP`

  The following example specifies a VLAN for an admin partition.

- `Request`

  `HTTP Method:` PUT

`URL:` http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/nspartition_vlan_binding/partition-dept1

`Request Headers:`

Cookie:NITRO_AUTH_TOKEN=<tokenvalue>

Content-Type:application/json

`Request Payload:`

```
{

    "nspartition_vlan_binding":

    {
```

```
    "partitionname":"partition-dept1",

    "vlan":"2"

    }

}
```

- `Response`

  `HTTP Status Code on Success:` 201 Created

`HTTP Status Code on Failure:` 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix-ADC-specific errors). The response payload provides details of the error.

- `Using REST APIs through SDKs`

  The following sample code specifies a VLAN for an admin partition.

`Java - Sample Code to specify the VLAN`

```
nspartition_vlan_binding nspartition_vlan_binding_object = new
nspartition_vlan_binding();



nspartition_vlan_binding_object.set_vlan(2);



nspartition_vlan_binding_object.set_partitionname("partition-dept1");



base_response result = nspartition_vlan_binding.add(nitroService,
nspartition_vlan_binding_object);
```

`.NET - Sample code to specify the VLAN`

```
nspartition_vlan_binding nspartition_vlan_binding_object = new
nspartition_vlan_binding();



nspartition_vlan_binding_object.vlan = 2;



nspartition_vlan_binding_object.partitionname = "partition-dept1";



base_response result = nspartition_vlan_binding.add(nitroService,
nspartition_vlan_binding_object);
```

`Python - Sample code to specify the VLAN`

```
nspartition_vlan_binding_object = nspartition_vlan_binding()



nspartition_vlan_binding_object.vlan = 2



nspartition_vlan_binding_object.partitionname = "partition-dept1"



result = nspartition_vlan_binding.add(nitroService,
nspartition_vlan_binding_object)
```

Switching Partitions

If you are associated with multiple admin partitions, you can switch to the required partition.

- `Using REST APIs through HTTP`

  The following example shows how to switch from current partition to a partition named partition-dept2.

- `Request`

  `HTTP Method:` POST

`URL:` http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/nspartition?action=Switch

`Request Headers:`

Cookie:NITRO_AUTH_TOKEN=<tokenvalue>

Content-Type:application/json

`Request Payload:`

```
{

    "nspartition":

    {

    "partitionname":"partition-dept2"

    }

}
```

- `Response`

  `HTTP Status Code on Success:` 201 Created

`HTTP Status Code on Failure:` 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix-ADC-specific errors). The response payload provides details of the error.

- `Using REST APIs through SDKs`

  The following sample code switches from current partition to a partition named "partition-dept2".

`Java - Sample code to switch partitions`

```
nspartition nspartitionObject = new nspartition();
```

```
vnspartitionObject.set_partitionname("partition-dept2");



base_response result = nspartition.Switch(nitroService,
nspartitionObject);
```

`.NET - Sample code to switch partitions`

```
nspartition nspartitionObject = new nspartition();



nspartitionObject.partitionname = "partition-dept2";



base_response result = nspartition.Switch(nitroService,
nspartitionObject);
```

`Python - Sample code to switch partitions`

```
nspartitionObject = nspartition()



nspartitionObject.partitionname = "partition-dept2"



result = nspartition.Switch(nitroService, nspartitionObject)
```

Managing AppExpert Applications

The following sections talks about exporting and importing an AppExpert application using NITRO APIs.

Exporting an AppExpert Application

This topic covers exporting an AppExpert application by using REST APIs through HTTP or SDKs.

- `Using REST APIs through HTTP`

  To export an AppExpert application, specify the parameters needed for the export operation in the apptemplateinfo object. Optionally, you can specify basic information about the AppExpert application template, such as the author of the configuration, a summary of the template functionality, and the template version number, in the template_info object. This information is stored as part of the template file that is created.

For example, to export an AppExpert application named MyApp1.

- `Request`

  `HTTP Method:` POST

`URL:` http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/apptemplateinfo?action=export

`Request Headers:`

Cookie:NITRO_AUTH_TOKEN=<tokenvalue>

Content-Type:application/json

`Request Payload:`

```
{

    "apptemplateinfo":

    {

        "appname":"MyApp1",

        "apptemplatefilename":"BizAp.xml",

        "template_info":

        {

            "templateversion_major":"2",

            "templateversion_minor":"1",

            "author":"XYZ",
```

```
            "introduction":"Intro",

            "summary":"Summary"

        }

    }

}
```

- `Response`

  `HTTP Status Code on Success:` 200 OK

`HTTP Status Code on Failure:` 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix-ADC-specific errors). The response payload provides details of the error.

- `Using REST APIs through SDKs`

  To export an AppExpert application, you must do the following:

1. Instantiate the com.citrix.netscaler.nitro.resource.config.app.application class.

   `Note.` For the python SDK, the package path is of the form nssrc.com.citrix.netscaler.....

2. Configure the properties of the AppExpert locally.
3. Export the AppExpert application.

The following samples export an AppExpert application named MyApp1.

`JAVA - sample code to export an AppExpert application`

```
application myapp = new application();

myapp.set_appname("MyApp1");

myapp.set_apptemplatefilename("myapp_template");

application.export(ns_session,myapp);
```

`.NET - sample code to export an AppExpert application`

```
application myapp = new application();

myapp.appname = "MyApp1";

myapp.apptemplatefilename = "myapp_template";

application.export(ns_session,myapp);
```

`Python - sample code to export an AppExpert application`

```
myapp = application()

myapp.appname = "MyApp1"

myapp.apptemplatefilename = "myapp_template"

application.export(ns_session, myapp)
```

Importing an AppExpert Application

This topic covers importing an AppExpert application by using REST APIs through HTTP or SDKs.

- `Using REST APIs through HTTP`

  To import an AppExpert application, specify the parameters needed for the import operation in the apptemplateinfo object.

For example, to import an AppExpert application named MyApp1.

- `Request`

  `HTTP Method:` POST

`URL:` http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/apptemplateinfo?action=import

`Request Headers:`

Cookie:NITRO_AUTH_TOKEN=<tokenvalue>

Content-Type:application/json

`Request Payload:`

```
{

"apptemplateinfo":

{

    "apptemplatefilename":"BizAp.xml",

    "deploymentfilename":"BizAp_deployment.xml",

    "appname":"MyApp1"

}

}
```

- `Response`

  `HTTP Status Code on Success:` 200 OK

`HTTP Status Code on Failure:` 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix-ADC-specific errors). The response payload provides details of the error.

To import an AppExpert application by specifying different deployment settings.

- `Request`

  `HTTP Method:` POST

`URL:` http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/apptemplateinfo?action=import

`Request Headers:`

Cookie:NITRO_AUTH_TOKEN=<tokenvalue>

Content-Type:application/json

`Request Payload:`

```
{

    "apptemplateinfo":

    {
```

```
    "apptemplatefilename":"BizAp.xml",

    "appname":"Myapp2",

    "deploymentinfo":

    {

        "appendpoint":

        [

            {

                "ipv46":"11.2.3.8",

                "port":80,

                "servicetype":"HTTP"

            }

        ],

        "service":

        [

            {

                "ip":"12.3.3.15",

                "port":80,

                "servicetype":"SSL"

            },

            {

                "ip":"14.5.5.16",

                "port":443,
```

```
                "servicetype":"SSL"

            }

        ]

    }

    }

}
```

- `Response`

  `HTTP Status Code on Success:` 200 OK

`HTTP Status Code on Failure:` 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix-ADC-specific errors). The response payload provides details of the error.

- `Using REST APIs through SDKs`

  To import an AppExpert application, you must do the following:

1. Instantiate the com.citrix.netscaler.nitro.resource.config.app.application class.

   `Note`. For the python SDK, the package path is of the form nssrc.com.citrix.netscaler.....
2. Configure the properties of the AppExpert locally.
3. Import the AppExpert application.

The following samples import an AppExpert application named MyApp1.

`Java - sample code to import an AppExpert application`

```
application myapp = new application();

myapp.set_appname("MyApp1");

myapp.set_apptemplatefilename("myapp_template");

application.Import(ns_session,myapp);
```

`.NET - sample code to import an AppExpert application`

```
application myapp = new application();

myapp.appname = "MyApp1";

myapp.apptemplatefilename = "myapp_template";

application.Import(ns_session,myapp);
```

`Python - sample code to import an AppExpert application`

```
myapp = application()

myapp.appname = "MyApp1"

myapp.apptemplatefilename = "myapp_template"

application.Import(ns_session, myapp)
```

Automate Citrix ADC Upgrade and Downgrade with a Single API

You can use the "install" API to automate not just installation, but also an upgrade or a downgrade of the build on a Citrix ADC appliance. You can specify a local or remote location for the build file.

- `Using REST APIs through HTTP`

  For example, the following information describes a downgrade to Citrix ADC release 10.5 build 46, using a local build file:

- `Request`

  `HTTP Method:` POST

`URL:` http://<NSIP\>/nitro/v1/config/install

`Request Headers:`

Cookie:NITRO_AUTH_TOKEN=\<tokenvalue \>

Content-Type: application/json

`Request Payload:`

```
{

“install”:

{

        “url”: “file:///var/tagma/build_tagma_46_nc.tgz”

    }

}
```

- `Response`

  `HTTP status Code on Success:`

201 Created

209 Citrix ADC specific warning

`Note.` when “y” option is not specified and warning is enabled, API returns “1120 - The configuration must be saved and the system rebooted for these settings to take effect” message in X-NITRO-WARNING.

`HTTP Status Code on Failure:` 599 Citrix ADC specific error

- `Additional parameters available in the install API request payload`
  - “y”:“true” - This option enables reboot on successful loading of kernel.
- “L”:“true” - This option enables the callhome feature.

`Supported formats for the "url" parameter (specifies the location of the tar.gz file for the build)`

- http://:@host/path/to/file
- https://:@host/path/to/file
- sftp://:@host/path/to/file
- scp://:@host/path/to/file
- ftp://:@host/path/to/file
- file://path/to/file
- `Possible errors`
  - Installation failed.
- Installation failed.
- Installation failed.
- Installation failed.
- Installation failed.
- Installation failed.
- Installation failed.
- Installation failed.

Handle Multiple NITRO Calls in a Single Request

You can use the "macroapi" API to create, update, and delete multiple resources simultaneously, and thereby minimize network traffic. For example, multiple load-balancing virtual servers can be added in a single API.

To account for the failure of some operations within the bulk operation, NITRO allows configuring one of the following behaviors.

- `EXIT`. When the first error is encountered, execution stops. The commands that were executed before the error are committed.
- `ROLLBACK`. When the first error is encountered, execution stops. The commands that were executed before the error are rolled back. Rollback is supported for add and bind commands only.
- `CONTINUE`. All the commands in the request are executed even if some commands fail.

You must specify the behavior of the bulk operation in the request header, by using the X-NITRO-ONERROR parameter.

- `Advantages`

  Heterogeneous resources can be configured with a single API. For example, multiple load balancing virtual servers and multiple services can be created, and services can be bound to load balancing virtual servers in a single API.

- `Limitations`

  Only homogenous operation is supported in this API. For example, multiple load balancing virtual servers can be created but cannot be updated or deleted in the same API. “rollback” is supported only on “add” and “bind” operations.

- `Using REST APIs through HTTP`

  To add multiple load balancing resources in a single request:

- `Request`

  `HTTP Method:` POST

`URL:` http://<NSIP\>/nitro/v1/config/macroapi

`Request Headers:`

Content-Type: application/json

Cookie: NITRO_AUTH_TOKEN=<tokenvalue>

X-NITRO-ONERROR: exit

`Request Payload:`

```
{

"lbvserver":[

{"name":"lbv1","servicetype":"http"},

{"name":"lbv2","servicetype":"http"}
```

```
],

"serviceGroup": [

{ "servicegroupname": "sg1", "servicetype": "HTTP" },

{ "servicegroupname": "sg2", "servicetype": "HTTP" }

],

"lbvserver_servicegroup_binding":[

{ "name":"lbv1", "servicegroupname":"sg1" },

{ "name":"lbv2", "servicegroupname":"sg2" }

]

}
```

- `Response`

`HTTP Status Code on Success:` 201 Created for the add operation and 200 OK for the update operation.

`HTTP Status Code on Failure:` 207 Multi Status with error details in the response payload. For more information, see Error Handling.

For deleting multiple resources using macroapi, use POST HTTP method with query parameter “action=remove” in the URI.

Simplify Management Operations with an Idempotent API

You can add or update Citrix ADC resources seamlessly, with a single API. Previously, an attempt to add a resource that was already configured, or to update a resource that was not yet configured, caused an error.

If you enable the “idempotent” query parameter (“idempotent=yes”) in any POST request, NITRO executes the request in an idempotent manner. An idempotent HTTP method is an HTTP method that can be called many times without different results, and POST is desinged as a non-idempotent method.

`Note.` Use POST request with “idempotent” option if you are unsure whether the resource in the request exists on Citrix ADC or not.

This API hides the inconsistencies between parameter lists of POST and PUT operations. For some NITRO resources, certain parameters are accepted only on PUT not in POST, or vice versa. By using this idempotent API, you can overcome such challenges.

- `Limitations`

  If a resource is already configured and you try to add the same resource again, the resource is "updated," but the arguments already present are not unset. For example, if a load balancing virtual server named "V1" is configured to use the round robin load balancing method, and you try to ADD an lbvserver named "V1" without specifying a value for "lbmethod" in the request, the Citrix ADC appliance does not unset "lbmethod" to its default value of "leastconnection."

- `Using REST APIs through HTTP`

  In the following example, “preferredntpserver” is allowed only in PUT, but when given in POST request with idempotent=yes, NITRO internally adds the ntpserver and updates it with given properties.

- `Request`

`HTTP Method:` POST

`URL:` http://<NSIP\>/nitro/v1/config/ntpserver?idempotent=yes

`Request Headers:`

Cookie:NITRO_AUTH_TOKEN=<tokenvalue>

Content-Type: application/json

`Request Payload:`

```
{



“ntpserver”:{“servername”:“ntp1”,“minpoll”:“4,  “preferredntpserver”:
“yes”}



}
```

- `Response`

  `HTTP Status Code on Success:` 200 OK

`HTTP Status Code on Failure:` 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix-ADC-specific errors). The response payload provides details of the error.

Retrieve Bindings in Bulk

This topic covers retrieving bindings information in bulk by using REST APIs through HTTP.

- `Using REST APIs through HTTP`

  You can use a bulk GET API to fetch bindings of all the entities of a given entity type.

For example, you can fetch bindings of all the load balancing virtual servers in one call instead of by using multiple GET by "name" calls. In the examples below, the Citrix ADC appliance has the following configuration.

- add lb vserver lbv1 http

- add lb vserver lbv2 http
- add service svc1 10.20.30.40 http 80
- add servicegroup sg1 http
- bind lb vserver lbv1 svc1
- bind lb vserver lbv1 sg1
- bind lb vserver lbv2 svc1
- bind lb vserver lbv2 sg1

`Example - To fetch bindings of all lbvservers, in a single NITRO API`

- `Request`

  `HTTP Method:` GET

`URL:` http://<NSIP\>/nitro/v1/config/lbvserver_binding?bulkbindings=yes

`Request Headers:`

Cookie:NITRO_AUTH_TOKEN=<tokenvalue>

Accept: application/json

- `Response`

  `HTTP Status Code on Success:` 200 OK

`HTTP Status Code on Failure:` 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix-ADC-specific errors). The response payload provides details of the error.

`Response Header:` Content-Type:application/json

`Response Payload:`

```
{

"errorcode":0,

"message":"Done",
```

```
"severity":"NONE",

"lbvserver_binding":[

{

    "name":"lbv1",

    "lbvserver_service_binding":[

    {

        "name":"lbv1",

        "servicename":"svc1",

        "stateflag":"536936451",

        "ipv46":"10.20.30.40",

        "port":80,
```

```
        "servicetype":"HTTP",

        "curstate":"DOWN",

        "weight":"1",

        "dynamicweight":"0",

        "cookieipport":"",

        "vserverid":"mcw1",

        "vsvrbindsvcip":"10.20.30.40",

        "vsvrbindsvcport":80,

        "preferredlocation":""

    }

    ],
```

```
    "lbvserver_servicegroup_binding":[

    {

        "name":"lbv1",

        "servicegroupname":"sg1",

        "stateflag":"536936464",

        "servicename":"sg1"

    }

    ]

},

{

    "name":"lbv2",
```

```
"lbvserver_service_binding":[

{

    "name":"lbv2",

    "servicename":"svc1",

    "stateflag":"536936451",

    "ipv46":"10.20.30.40",

    "port":80,

    "servicetype":"HTTP",

    "curstate":"DOWN",

    "weight":"1",

    "dynamicweight":"0",
```

```
        "cookieipport":"",

        "vserverid":"mcw2",

        "vsvrbindsvcip":"10.20.30.40",

        "vsvrbindsvcport":80,

        "preferredlocation":""

    }

    ],

    "lbvserver_servicegroup_binding":[

    {

        "name":"lbv2",

        "servicegroupname":"sg1",
```

```
        "stateflag":"536936464",

        "servicename":"sg1"

    }

    ]

}

]

}
```

`Example - To fetch only “service” bindings of all lbvservers`

- `Request`

  `HTTP Method:` GET

`URL:` http://<NSIP\>/nitro/v1/config/lbvserver_service_binding?bulkbindings=yes

`Request Header:` Content-Type:application/json

- `Response`

  `Response Payload:`

```
{

"errorcode":0,

"message":"Done",

"severity":"NONE",

"lbvserver_service_binding":[

    {

        "name":"lbv1",

        "servicename":"svc1",

        "stateflag":"536936451",

        "ipv46":"10.20.30.40",

        "port":80,
```

```
        "servicetype":"HTTP",

        "curstate":"DOWN",

        "weight":"1",

        "dynamicweight":"0",

        "cookieipport":"",

        "vserverid":"mcw1",

        "vsvrbindsvcip":"10.20.30.40",

        "vsvrbindsvcport":80,

        "preferredlocation":""

    },

    {
```

```
        "name":"lbv2",

        "servicename":"svc1",

        "stateflag":"536936451",

        "ipv46":"10.20.30.40",

        "port":80,

        "servicetype":"HTTP",

        "curstate":"DOWN",

        "weight":"1",

        "dynamicweight":"0",

        "cookieipport":"",

        "vserverid":"mcw2",
```

```
        "vsvrbindsvcip":"10.20.30.40",


        "vsvrbindsvcport":80,


        "preferredlocation":""


    }


    ]


}
```

View Individual Counter Information

This topic viewing individual counter information by using REST APIs through HTTP.

- `Using REST APIs through HTTP`

  To view global counters that are not otherwise shown by the Citrix ADC CLI or the GUI, you can now use the following URL format.

`URL`: http://<NSIP\>/nitro/v1/stat/nsglobalcntr?args=counters:\<counter1\>;\<counter2\>

Previously, these counter values could be viewed only through the “nsconmsg” Shell command.

`Note.` You can view only global counters of type ULONG and only up to 12 counters in a single request.

- `Example`

This example shows how to view the individual counters http_tot_Requests and http_tot_Responses. Enter the details in the system or tool that you are using to generate HTTP or HTTPS requests.

- `Request`

  `HTTP Method:` GET

`URL:` http://<NSIP\>/nitro/v1/stat/nsglobalcntr?args=counters:http_tot_Requests;http_tot_Responses

- `Response`

```
{

"errorcode": 0,



"message": "Done",



"severity": "NONE",



"nsglobalcntr":

    {

    "http_tot_Requests": "5783",

    "http_tot_Responses": "5783"

    }

}
```

Prevent XSS and CSRF Attacks by Disabling Basic Authentication

As an administrator or a root user, you can now prevent users from making API calls after using basic authentication (such as one-time credentials) to log on. You can use this feature to prevent Cross-Site Scripting (XSS), Cross-Site Request Forgery (CSRF), and other types of attacks.

- `Procedure for Disabling Basic Authentication`

  Use the following formats to enter the details in the system or tool that you are using to generate HTTP or HTTPS requests.

- `Using REST APIs through HTTP`

  - `Request`

  `URL:` http://<Citrix-ADC-ip-address(NSIP)\>/nitro/v1/config/systemparameter

`HTTP Method:` PUT

`Request Headers:`

Cookie:NITRO_AUTH_TOKEN=<tokenvalue>

Content-Type:application/json

`Request Payload:`

```
{

    "systemparameter":{ "basicauth":"disabled",}



}
```

- `Response`

  `HTTP Status Code on Success:` 200 OK

After you disable the basic authentication, access to Citrix ADC through the one-time password is denied and an error message is displayed.

- `Example`

This example shows what happens if you make any API call after basic authentication has been disabled.

- `Request`

  `URL:` http://10.102.201.159/nitro/v1/config/lbvserver

`HTTP Method:` GET

After you make a GET call, the a logon screen appears. If you click `Log In` after entering the logon credentials, then you get the following response, which shows an error message.

- `Response`

```
{"errorcode": 1244,"message": "Authentication Failed: Use of Basic
Authentication is Disabled.","severity": "ERROR"

}
```

Error and Exception Handling

This section covers error handling support for NITRO APIs and exception handling support for SDKs.

Error Handling for NITRO APIs

In case of a failed request, NITRO provides the required information through the HTTP status code and in the response header and response payload.

- Error in a Single Resource Operation
- Error in a Bulk Operations
- Warnings
- `Error in a Single Resource Operation`

  The response of a single erroneous operation is as follows:
- `Response`

  `HTTP Status Code:` 4xx \<string\> (for general HTTP errors) or 5xx \<string\> (for Citrix-ADC-specific errors)

`Response Header:` Content-Type:application/json

`Response Payload:`

```
{

    errorcode: <Error code>,

    message: "<Error message>",

    severity: "ERROR"

}
```

- `Error in a Bulk Operation`

When there is a failure in one of the bulk operations, the response payload gives a combination of success and failure (depends on the value set for X-NITRO-ONERROR in the request header).

- `Response`

  `HTTP Status Code:` 207 Multi Status

`Response Header:` Content-Type:application/json

`Response Payload when X-NITRO-ONERROR is set to continue:`

When the first operation fails, the request is not terminated. The response payload shows the error details of the failed operation and the success status of the other operations.

```
{

    "errorcode": 1243,

    "message": "Bulk operation failed",

    "severity": "ERROR",

    "response":

    [

        {

            "errorcode": 273,

            "message": "Resource already exists",

            "severity": "ERROR"

        },

        {

            "errorcode": 0,

            "message": "Done",

            "severity": "NONE"
```

```
        }
    ]
}
```

`Response Payload when X-NITRO-ONERROR is set to exit:`

When the first operation fails, the request is terminated. The response payload only shows the error details of the failed operation.

```
{
    "errorcode": 1243,
    "message": "Bulk operation failed",
    "severity": "ERROR",
    "response":
    [
        {
            "errorcode": 273,
            "message": "Resource already exists",
            "severity": "ERROR"
        }
    ]
}
```

- `Warnings in NITRO Operations`

Warnings can be captured by specifying the "warning" query parameter as "yes" when performing any NITRO operation. For example, to get warnings while connecting to the Citrix ADC appliance, the URL is as follows:

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/llbvserver?warning=yes

If there are any warnings, the response is as follows:

- `Response`

  `HTTP Status Code:` 209 X-NITRO-WARNING

`Response Header:` X-NITRO-WARNING →1067 - Feature(s) not enabled

Exception Handling Support for SDKs

The status of a NITRO request is captured in the com.citrix.netscaler.nitro.exception.nitro_exception class. This class provides the following details of the exception:

- `Session ID`. The session in which the exception occurred.
- `Severity`. The severity of the exception: error or warning. By default, only errors are captured. To capture warnings, you must set the warning flag to true, while connecting to the appliance.
- `Error code`. The status of the NITRO request. An error code of 0 indicates that the NITRO request is successful. A non-zero error code indicates an error in processing the NITRO request.
- `Error message`. Provides a brief description of the exception.

For a list of error codes, see the errorlisting.html file available in the \<NITRO_SDK_HOME\>/doc/api_reference folder.

API Reference

This documentation provides details of all operations that can be performed on the Citrix ADC appliance by using the REST API.

- Configuration
- Statistics
- Error Messages

Configuration

The configuration API is arranged in a logical groups of objects. Choose an appropriate group to find more details.

NAME	DESCRIPTION
AAA	Authentication, authorization, and accounting service configuration.
Analytics	Analytics configuration.
App	Application
Appflow	AppFlow configuration.
Application Firewall	Application Firewall configuration.
Appqoe	Application Level Quality of Experience configuration
Audit	Audit configuration.
Authentication	Authentication configuration.
Authorization	Authorization configuration. Authorization services check which resources users are authorized to access, and grant permissions accordingly.
Autoscale	AutoScale configuration.
Azure	Azure configuration
Basic	Basic system configuration.
Bfd	Bidirectional Forwarding Detection

NAME	DESCRIPTION
Ca	Content Accelerator Configuration. The Content Accelerator feature is used to cache static and dynamic web application data.
Integrated Caching	Integrated Caching Configuration. The Integrated Caching feature is used to cache static and dynamic web application data.
Cloud	Citrix ADC as SD proxy Configuration and cloud discovery commands.
Cluster	Cluster commands.
Compression	Compression configuration. The systems feature for compressing HTTP responses to compression-aware browsers.
CI	Content Inspection
Cache Redirection	Cache redirection configuration. The cache redirection feature can transparently redirect cacheable HTTP requests to a cache and send non-cacheable or dynamic HTTP requests to the origin server. A cache stores or caches frequently requested web content and serves such web content to a client on behalf of the origin servers, alleviating the load on the origin server farm.
Content Switching	Content Switching configuration. Content Switching feature that enables you to direct traffic to servers on the basis of content.
Db	All the commands associated with database user
Domain Name Service	Domain Name Service(DNS) configuration.
HTTP DoS Protection	HTTP DOS protection configuration.

NAME	DESCRIPTION
Front-end-optimization	Front end optimization configuration. The systems feature to optimize Web content for performance.
Filter	Filter configuration. Protection from malicious attacks can be achieved for web sites at the content level with the systems content filtering feature. When this feature is enabled, the system inspects every incoming request with the user-configured rules based on any HTTP URL or headers.
Global Server Load Balancing	Global Server Load Balancing (GSLB) configuration. GSLB feature ensures that client requests are directed to a best performing site available in a global enterprise and distributed Internet environment.
High Availability	High-availability commands.
Ica	ICA configuration.
Ipsec	IPSEC configuration.
Ipsecalg	IPSECALG configuration.
Load Balancing	Load Balancing configuration. The load balancing methods manage the selection of an appropriate physical server in a server farm.
LLDP	Link Layer Discovery Protocol.
Lsn	Large Scale NAT commands
Network	Networking commands.
NS	System/Global level configuration.
NTP	NTP configuration.
Pcp	Port Control Protocol Configuration.

NAME	DESCRIPTION
Policy	Policy configuration.
Priority Queuing	Priority Queuing configuration. Priority Queuing ensures that key customer web sites get the timely responses they need. System provides priority-based queuing, which provides different levels of service to customers.
Protocol	Protocol Configuration.
Rdp	RDP configuration.
Reputation	Reputation services configuration.
Responder	Responder configuration.
Rewrite	Rewrite configuration.
Rise	RISE feature related configuration.
Router	Router configuration.
Sure Connect	SureConnect configuration. SureConnect allows the system to deliver alternate content to clients when the system is under heavy load for the original URL. The original content will be served to the client, automatically, after an estimated delay period.
Smpp	All the commands associated with SMPP.
SNMP	SNMP(Simple Network Management Protocol) configuration.
Spillover	Spillover policies and actions.
SSL	SSL Configuration
Stream	Stream configuration.
Subscriber	Subscriber configuration

NAME	DESCRIPTION
System	System configuration.
Traffic Management	TM session/policy configuration.
Transform	URL Transformation configuration.
Tunnel	SSL VPN Tunnel Configuration.
Ulfd	ULFD server configuration.
Urlfiltering	URL Filtering feature is used control access to webpages based on category.
User	User protocol configuration.
Utility	Utilities.
VideoOptimization	Video optimization feature is used to show (i) the stats of different media types that are being served by the Citrix ADC and (ii) the details of optimization applied on ABR videos
SSL VPN	Virtual Private Network configuration.
Wf	Web Front configuration.
WebInterface	Web Interface configuration.
Xm	XenMobile deployment configuration.

AAA

AAA

Authentication, authorization, and accounting service configuration.

NAME	DESCRIPTION
aaacertparams	Configuration for certificate parameter

NAME	DESCRIPTION
aaaglobal_aaapreauthenticationpolicy_binding	Binding object showing the aaapreauthenticationpolicy that can be bound to aaaglobal
aaaglobal_authenticationnegotiateaction_binding	Binding object showing the authenticationnegotiateaction that can be bound to aaaglobal
aaaglobal_binding	Binding object showing the resources that can be bound to aaaglobal
aaagroup	Configuration for AAA group
aaagroup_aaauser_binding	Binding object showing the aaauser that can be bound to aaagroup
aaagroup_auditnslogpolicy_binding	Binding object showing the auditnslogpolicy that can be bound to aaagroup
aaagroup_auditsyslogpolicy_binding	Binding object showing the auditsyslogpolicy that can be bound to aaagroup
aaagroup_authorizationpolicy_binding	Binding object showing the authorizationpolicy that can be bound to aaagroup
aaagroup_binding	Binding object showing the resources that can be bound to aaagroup
aaagroup_intranetip6_binding	Binding object showing the intranetip6 that can be bound to aaagroup
aaagroup_intranetip_binding	Binding object showing the intranetip that can be bound to aaagroup
aaagroup_tmsessionpolicy_binding	Binding object showing the tmsessionpolicy that can be bound to aaagroup
aaagroup_vpnintranetapplication_binding	Binding object showing the vpnintranetapplication that can be bound to aaagroup

NAME	DESCRIPTION
aaagroup_vpnsessionpolicy_binding	Binding object showing the vpnsessionpolicy that can be bound to aaagroup
aaagroup_vpntrafficpolicy_binding	Binding object showing the vpntrafficpolicy that can be bound to aaagroup
aaagroup_vpnurl_binding	Binding object showing the vpnurl that can be bound to aaagroup
aaagroup_vpnurlpolicy_binding	Binding object showing the vpnurlpolicy that can be bound to aaagroup
aaakcdaccount	Configuration for Kerberos constrained delegation account
aaaldapparams	Configuration for LDAP parameter
aaaotpparameter	Configuration for AAA otpparameter
aaaparameter	Configuration for AAA parameter
aaapreauthenticationaction	Configuration for pre authentication action
aaapreauthenticationparameter	Configuration for pre authentication parameter
aaapreauthenticationpolicy	Configuration for pre authentication policy
aaapreauthenticationpolicy_aaaglobal_binding	Binding object showing the aaaglobal that can be bound to aaapreauthenticationpolicy
aaapreauthenticationpolicy_binding	Binding object showing the resources that can be bound to aaapreauthenticationpolicy
aaapreauthenticationpolicy_vpnvserver_binding	Binding object showing the vpnvserver that can be bound to aaapreauthenticationpolicy
aaaradiusparams	Configuration for RADIUS parameter
aaasession	Configuration for active connection
aaassoprofile	Configuration for aaa sso profile

NAME	DESCRIPTION
aaatacacsparams	Configuration for tacacs parameters
aaauser	Configuration for AAA user
aaauser_aaagroup_binding	Binding object showing the aaagroup that can be bound to aaauser
aaauser_auditnslogpolicy_binding	Binding object showing the auditnslogpolicy that can be bound to aaauser
aaauser_auditsyslogpolicy_binding	Binding object showing the auditsyslogpolicy that can be bound to aaauser
aaauser_authorizationpolicy_binding	Binding object showing the authorizationpolicy that can be bound to aaauser
aaauser_binding	Binding object showing the resources that can be bound to aaauser
aaauser_intranetip6_binding	Binding object showing the intranetip6 that can be bound to aaauser
aaauser_intranetip_binding	Binding object showing the intranetip that can be bound to aaauser
aaauser_tmsessionpolicy_binding	Binding object showing the tmsessionpolicy that can be bound to aaauser
aaauser_vpnintranetapplication_binding	Binding object showing the vpnintranetapplication that can be bound to aaauser
aaauser_vpnsessionpolicy_binding	Binding object showing the vpnsessionpolicy that can be bound to aaauser
aaauser_vpntrafficpolicy_binding	Binding object showing the vpntrafficpolicy that can be bound to aaauser
aaauser_vpnurl_binding	Binding object showing the vpnurl that can be bound to aaauser

NAME	DESCRIPTION
aaauser_vpnurlpolicy_binding	Binding object showing the vpnurlpolicy that can be bound to aaauser

aaacertparams

Configuration for certificate parameter resource.

Properties

(click to see Operations)

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
usernamefield	<String>	Read-write	Client certificate field that contains the username, in the format <field>:<subfield>. .
groupnamefield	<String>	Read-write	Client certificate field that specifies the group, in the format <field>:<subfield>.
defaultauthentication group	<String>	Read-write	This is the default group that is chosen when the authentication succeeds in addition to extracted groups. Maximum length = 64
twofactor	<String>	Read-only	The state of the two-factor authentication. Possible values = ON, OFF

Operations

(click to see Properties)

UPDATE| UNSET| GET (ALL)

Some options that you can use for each operations:

- **Getting warnings in response:**NITRO allows you to get warnings in an operation by specifying the "warning" query parameter as "yes". For example, to get warnings while connecting to the Citrix ADC appliance, the URL is as follows:

  http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/login?warning=yes

  If any, the warnings are displayed in the response payload with the HTTP code "209 X-NITRO-WARNING".

- **Authenticated access for individual NITRO operations:**NITRO allows you to logon to the Citrix ADC appliance to perform individual operations. You can use this option instead of creating a NITRO session (using the login object) and then using that session to perform all operations,

  To do this, you must specify the username and password in the request header of the NITRO request as follows:

  X-NITRO-USER:<username>

  X-NITRO-PASS:<password>

  Note: In such cases, make sure that the request header DOES not include the following:

  Cookie:NITRO_AUTH_TOKEN=<tokenvalue>

`*Note:`*

Mandatory parameters are marked in redand placeholder content is marked in <green>.

update

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/aaacertparams

HTTP Method:PUT

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Content-Type:application/json

Request Payload:

```
{"aaacertparams":{

"usernamefield":<String_value>,

"groupnamefield":<String_value>,

"defaultauthenticationgroup":<String_value>

}}
```

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error

unset

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/aaacertparams?**action=unset**

HTTP Method:POST

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Content-Type:application/json

Request Payload:

```
{"aaacertparams":{

"usernamefield":true,

"groupnamefield":true,

"defaultauthenticationgroup":true

}}
```

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error

get (all)

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/aaacertparams

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "aaacertparams": [ {

"twofactor":<String_value>,

"usernamefield":<String_value>,

"groupnamefield":<String_value>,

"defaultauthenticationgroup":<String_value>

}]}
```

aaaglobal_aaapreauthenticationpolicy_binding

Binding object showing the aaapreauthenticationpolicy that can be bound to aaaglobal.

Properties

(click to see Operations)

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
priority	<Double>	Read-write	Priority of the bound policy.
policy	<String>	Read-write	Name of the policy to be unbound. Minimum length = 1
bindpolicytype	<Double>	Read-only	Bound policy type.
__count	<Double>	Read-only	count parameter

Operations

(click to see Properties)

ADD:| DELETE:| GET| GET (ALL)| COUNT

Some options that you can use for each operations:

- **Getting warnings in response:**NITRO allows you to get warnings in an operation by specifying the "warning" query parameter as "yes". For example, to get warnings while connecting to the Citrix ADC appliance, the URL is as follows:

  http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/login?warning=yes

  If any, the warnings are displayed in the response payload with the HTTP code "209 X-NITRO-WARNING".
- **Authenticated access for individual NITRO operations:**NITRO allows you to logon to the Citrix ADC appliance to perform individual operations. You can use this option instead of creating a NITRO session (using the login object) and then using that session to perform all operations,

  To do this, you must specify the username and password in the request header of the NITRO request as follows:

X-NITRO-USER:<username>

X-NITRO-PASS:<password>

Note: In such cases, make sure that the request header DOES not include the following:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue>

`*Note:`*

Mandatory parameters are marked in redand placeholder content is marked in <green>.

add

URL:http://<Citrix-ADC-IP-address(NSIP)/nitro/v1/config/aaaglobal_aaapreauthenticationpolicy_binding

HTTP Method:PUT

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Content-Type:application/json

Request Payload:

```
{
"aaaglobal_aaapreauthenticationpolicy_binding":{
"policy":<String_value>,
"priority":<Double_value>
}}
```

Response:

HTTP Status Code on Success: 201 Created HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error

delete

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/aaaglobal_aaapreauthenticationpolicy_binding

Query-parameters:

args

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/aaaglobal_aaapreauthenticationpolicy_binding?**args=policy:<String_value>**

HTTP Method:DELETE

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue>

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error

get

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/aaaglobal_aaapreauthenticationpolicy_binding

Query-parameters:

filter

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/aaaglobal_aaapreauthenticationpolicy_binding?**filter=property-name1:property-value1,property-name2:property-value2**

Use this query-parameter to get the filtered set of aaaglobal_aaapreauthenticationpolicy_binding resources configured on Citrix ADC. Filtering can be done on any of the properties of the resource.

pagination

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/aaaglobal_aaapreauthenticationpolicy_binding?**pagesize=#no;pageno=#no**

Use this query-parameter to get the aaaglobal_aaapreauthenticationpolicy_binding resources in chunks.

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "aaaglobal_aaapreauthenticationpolicy_binding": [ {

"priority":<Double_value>,

"policy":<String_value>,

"bindpolicytype":<Double_value>

}]}
```

get (all)

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/aaaglobal_aaapreauthenticationpolicy_binding

Query-parameters:

bulkbindings

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/aaaglobal_aaapreauthenticationpolicy_binding?**bulkbindings=yes**

NITRO allows you to fetch bindings in bulk.

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "aaaglobal_aaapreauthenticationpolicy_binding": [ {

"priority":<Double_value>,

"policy":<String_value>,

"bindpolicytype":<Double_value>

}]}
```

count

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/aaaglobal_aaapreauthenticationpolicy_binding?**count=yes**

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "aaaglobal_aaapreauthenticationpolicy_binding": [ { "__count":
"#no"} ] }
```

aaaglobal_authenticationnegotiateaction_binding

Binding object showing the authenticationnegotiateaction that can be bound to aaaglobal.

Properties

(click to see Operations)

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
windowsprofile	<String>	Read-write	Name of the negotiate profile to be bound. Minimum length = 1 Maximum length = 32
__count	<Double>	Read-write	count parameter

Operations

(click to see Properties)

ADD:| DELETE:| GET| GET (ALL)| COUNT

Some options that you can use for each operations:

- **Getting warnings in response:**NITRO allows you to get warnings in an operation by specifying the "warning" query parameter as "yes". For example, to get warnings while connecting to the Citrix ADC appliance, the URL is as follows:

  http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/login?warning=yes

  If any, the warnings are displayed in the response payload with the HTTP code "209 X-NITRO-WARNING".

- **Authenticated access for individual NITRO operations:**NITRO allows you to logon to the Citrix ADC appliance to perform individual operations. You can use this option instead of creating a NITRO session (using the login object) and then using that session to perform all operations,

  To do this, you must specify the username and password in the request header of the NITRO request as follows:

  X-NITRO-USER:<username>

  X-NITRO-PASS:<password>

  Note: In such cases, make sure that the request header DOES not include the following:

  Cookie:NITRO_AUTH_TOKEN=<tokenvalue>

`*Note:`*

Mandatory parameters are marked in redand placeholder content is marked in <green>.

add

URL:http://<Citrix-ADC-IP-address(NSIP)/nitro/v1/config/aaaglobal_authenticationnegotiateaction_binding

HTTP Method:PUT

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Content-Type:application/json

Request Payload:

```
{
```

```
"aaaglobal_authenticationnegotiateaction_binding":{

"windowsprofile":<String_value>

}}
```

Response:

HTTP Status Code on Success: 201 Created HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error

delete

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/aaaglobal_authenticationnegotiateaction_binding

Query-parameters:

args

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/aaaglobal_authenticationnegotiateaction_binding?**args=windowsprofile:<String_value>**

HTTP Method:DELETE

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue>

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error

get

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/aaaglobal_authenticationnegotiateaction_binding

Query-parameters:

filter

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/aaaglobal_authenticationnegotiateaction_binding?**filter=property-name1:property-value1,property-name2:property-value2**

Use this query-parameter to get the filtered set of aaaglobal_authenticationnegotiateaction_binding resources configured on Citrix ADC. Filtering can be done on any of the properties of the resource.

pagination

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/aaaglobal_authenticationnegotiateaction_binding?**pagesize=#no;pageno=#no**

Use this query-parameter to get the aaaglobal_authenticationnegotiateaction_binding resources in chunks.

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "aaaglobal_authenticationnegotiateaction_binding": [ {

"windowsprofile":<String_value>

}]}
```

get (all)

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/aaaglobal_authenticationnegotiateaction_binding

Query-parameters:

bulkbindings

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/aaaglobal_authenticationnegotiateaction_binding?**bulkbindings=yes**

NITRO allows you to fetch bindings in bulk.

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "aaaglobal_authenticationnegotiateaction_binding": [ {

"windowsprofile":<String_value>

}]}
```

count

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/aaaglobal_authenticationnegotiateaction_binding?**count=yes**

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "aaaglobal_authenticationnegotiateaction_binding": [ { "__count":
"#no"} ] }
```

aaaglobal_binding

Binding object which returns the resources bound to aaaglobal.

Properties

(click to see Operations)

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
aaaglobal_authenticationnegotiateaction_binding	<aaaglobal_authenticationnegotiateaction_binding[]>	Read-only	authenticationnegotiateaction that can be bound to aaaglobal.
aaaglobal_aaapreauthenticationpolicy_binding	<aaaglobal_aaapreauthenticationpolicy_binding[]>	Read-only	aaapreauthentication policy that can be bound to aaaglobal.

Operations

(click to see Properties)

GET| GET (ALL)

Some options that you can use for each operations:

- **Getting warnings in response:**NITRO allows you to get warnings in an operation by specifying the "warning" query parameter as "yes". For example, to get warnings while connecting to the Citrix ADC appliance, the URL is as follows:

  http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/login?warning=yes

  If any, the warnings are displayed in the response payload with the HTTP code "209 X-NITRO-WARNING".

- **Authenticated access for individual NITRO operations:**NITRO allows you to logon to the Citrix ADC appliance to perform individual operations. You can use this option instead of creating a NITRO session (using the login object) and then using that session to perform all operations,

  To do this, you must specify the username and password in the request header of the NITRO request as follows:

  X-NITRO-USER:<username>

  X-NITRO-PASS:<password>

  Note: In such cases, make sure that the request header DOES not include the following:

  Cookie:NITRO_AUTH_TOKEN=<tokenvalue>

`*Note:`*

Mandatory parameters are marked in redand placeholder content is marked in <green>.

get

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/aaaglobal_binding

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "aaaglobal_binding": [ {

"aaaglobal_authenticationnegotiateaction_binding":<aaaglobal_authenticatio
nnegotiateaction_binding[]_value>,

"aaaglobal_aaapreauthenticationpolicy_binding":<aaaglobal_aaapreauthentica
tionpolicy_binding[]_value>

}]}
```

get (all)

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/aaaglobal_binding

Query-parameters:

bulkbindings

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/aaaglobal_binding?**bulkbindings=yes**

NITRO allows you to fetch bindings in bulk.

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "aaaglobal_binding": [ {

"aaaglobal_authenticationnegotiateaction_binding":<aaaglobal_authenticatio
nnegotiateaction_binding[]_value>,

"aaaglobal_aaapreauthenticationpolicy_binding":<aaaglobal_aaapreauthentica
tionpolicy_binding[]_value>

}]}
```

aaagroup

Configuration for AAA group resource.

Properties

(click to see Operations)

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
groupname	<String>	Read-write	Name for the group. Must begin with a letter, number, or the underscore character (_), and must consist only of letters, numbers, and the hyphen (-), period (.) pound (#), space ( ), at sign (@), equals (=), colon (:), and underscore characters. Cannot be changed after the group is added. The following requirement applies

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
			only to the Citrix ADC CLI: If the name includes one or more spaces, enclose the name in double or single quotation marks (for example, "my aaa group" or 'my aaa group'). Minimum length = 1
weight	<Double>	Read-write	Weight of this group with respect to other configured aaa groups (lower the number higher the weight). Default value: 0 Minimum value = 0 Maximum value = 65535
loggedin	<Boolean>	Read-write	Display only the group members who are currently logged in.
__count	<Double>	Read-only	count parameter

Operations

(click to see Properties)

ADD| DELETE| GET (ALL)| GET| COUNT

Some options that you can use for each operations:

- **Getting warnings in response:**NITRO allows you to get warnings in an operation by specifying the "warning" query parameter as "yes". For example, to get warnings while connecting to the Citrix ADC appliance, the URL is as follows:

  http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/login?warning=yes

  If any, the warnings are displayed in the response payload with the HTTP code "209 X-NITRO-WARNING".

- **Authenticated access for individual NITRO operations:**NITRO allows you to logon to the Citrix ADC appliance to perform individual operations. You can use this option instead of creating a NITRO session (using the login object) and then using that session to perform all operations,

  To do this, you must specify the username and password in the request header of the NITRO request as follows:

  X-NITRO-USER:<username>

  X-NITRO-PASS:<password>

  Note: In such cases, make sure that the request header DOES not include the following:

  Cookie:NITRO_AUTH_TOKEN=<tokenvalue>

`*Note:`*

Mandatory parameters are marked in redand placeholder content is marked in <green>.

add

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/aaagroup

HTTP Method:POST

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Content-Type:application/json

Request Payload:

```
{"aaagroup":{

<b>"groupname":<String_value>,

</b>"weight":<Double_value>

}}
```

Response:

HTTP Status Code on Success: 201 Created HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error

delete

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/aaagroup/groupname_value<String>

HTTP Method:DELETE

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue>

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error

get (all)

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/aaagroup

Query-parameters:

args

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/aaagroup?
args=groupname:<String_value>,loggedin:<Boolean_value>

Use this query-parameter to get aaagroup resources based on additional properties.

attrs

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/aaagroup?**attrs=property-name1,property-name2**

Use this query parameter to specify the resource details that you want to retrieve.

filter

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/aaagroup?**filter=property-name1:property-val1,property-name2:property-val2**

Use this query-parameter to get the filtered set of aaagroup resources configured on Citrix ADC. Filtering can be done on any of the properties of the resource.

view

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/aaagroup?**view=summary**

Note: By default, the retrieved results are displayed in detail view (?view=detail).

pagination

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/aaagroup?**pagesize=#no;pageno=#no**

Use this query-parameter to get the aaagroup resources in chunks.

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "aaagroup": [ {

groupname:<String_value>,loggedin:<Boolean_value>"weight":<Double_value>

}]}
```

get

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/aaagroup/groupname_value<String>

Query-parameters:

attrs

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/aaagroup/groupname_value<String>?**attrs=property-name1,property-name2**

Use this query parameter to specify the resource details that you want to retrieve.

view

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/aaagroup/groupname_value<String>?**view=summary**

Note: By default, the retrieved results are displayed in detail view (?view=detail).

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "aaagroup": [ {

groupname:<String_value>,loggedin:<Boolean_value>"weight":<Double_value>

}]}
```

count

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/aaagroup?**count=yes**

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "aaagroup": [ { "__count": "#no"} ] }
```

aaagroup_aaauser_binding

Binding object showing the aaauser that can be bound to aaagroup.

Properties

(click to see Operations)

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
gotopriorityexpression	<String>	Read-write	Expression or other value specifying the next policy to evaluate if the current policy evaluates to TRUE. Specify one of the following values: * NEXT - Evaluate the policy with the next higher priority number. * END - End policy evaluation. * USE_INVOCATION_RESULT - Applicable if this policy invokes another policy label. If the final goto in the invoked policy label has a value of END, the evaluation stops. If the final goto is anything other than END, the current policy label performs a NEXT. * An expression that evaluates to a number. If you specify an expression, the number to which it evaluates determines the next policy to evaluate, as follows: * If the expression evaluates to a higher numbered priority, the policy with that priority is evaluated

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
			next. * If the expression evaluates to the priority of the current policy, the policy with the next higher numbered priority is evaluated next. * If the expression evaluates to a number that is larger than the largest numbered priority, policy evaluation ends. An UNDEF event is triggered if: * The expression is invalid. * The expression evaluates to a priority number that is numerically lower than the current policy's priority. * The expression evaluates to a priority number that is between the current policy's priority number (say, 30) and the highest priority number (say, 100), but does not match any configured priority number (for example, the expression evaluates to the number 85). This example assumes that the priority number increments by 10 for

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
			every successive policy, and therefore a priority number of 85 does not exist in the policy label.
username	<String>	Read-write	The user name.
groupname	<String>	Read-write	Name of the group that you are binding. Minimum length = 1
__count	<Double>	Read-write	count parameter

Operations

(click to see Properties)

ADD:| DELETE:| GET| GET (ALL)| COUNT

Some options that you can use for each operations:

- **Getting warnings in response:**NITRO allows you to get warnings in an operation by specifying the "warning" query parameter as "yes". For example, to get warnings while connecting to the Citrix ADC appliance, the URL is as follows:

  http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/login?warning=yes

  If any, the warnings are displayed in the response payload with the HTTP code "209 X-NITRO-WARNING".

- **Authenticated access for individual NITRO operations:**NITRO allows you to logon to the Citrix ADC appliance to perform individual operations. You can use this option instead of creating a NITRO session (using the login object) and then using that session to perform all operations,

  To do this, you must specify the username and password in the request header of the NITRO request as follows:

  X-NITRO-USER:<username>

  X-NITRO-PASS:<password>

Note: In such cases, make sure that the request header DOES not include the following:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue>

`*Note:` *

Mandatory parameters are marked in redand placeholder content is marked in <green>.

add

URL:http://<Citrix-ADC-IP-address(NSIP)/nitro/v1/config/aaagroup_aaauser_binding

HTTP Method:PUT

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Content-Type:application/json

Request Payload:

```
{

"aaagroup_aaauser_binding":{

<b>"groupname":<String_value>,

</b>"username":<String_value>,

"gotopriorityexpression":<String_value>

}}
```

Response:

HTTP Status Code on Success: 201 Created HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error

delete

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/aaagroup_aaauser_binding/groupname_value<String>

Query-parameters:

args

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/aaagroup_aaauser_binding/groupname_value<String>?**args=username:<String_value>**

HTTP Method:DELETE

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue>

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error

get

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/aaagroup_aaauser_binding/groupname_value<String>

Query-parameters:

filter

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/aaagroup_aaauser_binding/groupname_value<String>?**filter=property-name1:property-value1,property-name2:property-value2**

Use this query-parameter to get the filtered set of aaagroup_aaauser_binding resources configured on Citrix ADC. Filtering can be done on any of the properties of the resource.

pagination

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/aaagroup_aaauser_binding/groupname_value<String>?**pagesize=#no;pageno=#no**

Use this query-parameter to get the aaagroup_aaauser_binding resources in chunks.

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "aaagroup_aaauser_binding": [ {

"gotopriorityexpression":<String_value>,

"username":<String_value>,

"groupname":<String_value>

}]}
```

get (all)

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/aaagroup_aaauser_binding

Query-parameters:

bulkbindings

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/aaagroup_aaauser_binding?**bulkbindings=yes**

NITRO allows you to fetch bindings in bulk.

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "aaagroup_aaauser_binding": [ {

"gotopriorityexpression":<String_value>,

"username":<String_value>,

"groupname":<String_value>

}]}
```

count

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/aaagroup_aaauser_binding/groupname_value<String>?**count=yes**

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{"aaagroup_aaauser_binding": [ { "__count": "#no"} ] }
```

aaagroup_auditnslogpolicy_binding

Binding object showing the auditnslogpolicy that can be bound to aaagroup.

Properties

(click to see Operations)

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
priority	<Double>	Read-write	Integer specifying the priority of the policy. A lower number indicates a higher priority. Policies are evaluated in the order of their priority numbers. Maximum value for default syntax policies is 2147483647 and for classic policies is 64000. Minimum value = 0 Maximum value = 2147483647
gotopriorityexpression	<String>	Read-write	Expression or other value specifying the next policy to evaluate if the current policy evaluates to TRUE. Specify one of the following values: * NEXT - Evaluate the

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
			policy with the next higher priority number. * END - End policy evaluation. * USE_INVOCATION_RESULT - Applicable if this policy invokes another policy label. If the final goto in the invoked policy label has a value of END, the evaluation stops. If the final goto is anything other than END, the current policy label performs a NEXT. * An expression that evaluates to a number. If you specify an expression, the number to which it evaluates determines the next policy to evaluate, as follows: * If the expression evaluates to a higher numbered priority, the policy with that priority is evaluated next. * If the expression evaluates to the priority of the current policy, the policy with the next higher numbered priority is evaluated next. * If the expression evaluates

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
			to a number that is larger than the largest numbered priority, policy evaluation ends. An UNDEF event is triggered if: * The expression is invalid. * The expression evaluates to a priority number that is numerically lower than the current policy's priority. * The expression evaluates to a priority number that is between the current policy's priority number (say, 30) and the highest priority number (say, 100), but does not match any configured priority number (for example, the expression evaluates to the number 85). This example assumes that the priority number increments by 10 for every successive policy, and therefore a priority number of 85 does not exist in the policy label.
policy	<String>	Read-write	The policy name.

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
type	<String>	Read-write	Bindpoint to which the policy is bound. Default value: REQUEST Possible values = REQUEST, UDP_REQUEST, DNS_REQUEST, ICMP_REQUEST
groupname	<String>	Read-write	Name of the group that you are binding. Minimum length = 1
acttype	<Double>	Read-only	.
__count	<Double>	Read-only	count parameter

Operations

(click to see Properties)

ADD:| DELETE:| GET| GET (ALL)| COUNT

Some options that you can use for each operations:

- **Getting warnings in response:**NITRO allows you to get warnings in an operation by specifying the "warning" query parameter as "yes". For example, to get warnings while connecting to the Citrix ADC appliance, the URL is as follows:

  http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/login?warning=yes

  If any, the warnings are displayed in the response payload with the HTTP code "209 X-NITRO-WARNING".

- **Authenticated access for individual NITRO operations:**NITRO allows you to logon to the Citrix ADC appliance to perform individual operations. You can use this option instead of creating a NITRO session (using the login object) and then using that session to perform all operations,

To do this, you must specify the username and password in the request header of the NITRO request as follows:

X-NITRO-USER:<username>

X-NITRO-PASS:<password>

Note: In such cases, make sure that the request header DOES not include the following:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue>

`*Note:`*

Mandatory parameters are marked in redand placeholder content is marked in <green>.

add

URL:http://<Citrix-ADC-IP-address(NSIP)/nitro/v1/config/aaagroup_auditnslogpolicy_binding

HTTP Method:PUT

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Content-Type:application/json

Request Payload:

```
{

"aaagroup_auditnslogpolicy_binding":{

<b>"groupname":<String_value>,

</b>"policy":<String_value>,

"priority":<Double_value>,

"type":<String_value>,

"gotopriorityexpression":<String_value>

}}
```

Response:

HTTP Status Code on Success: 201 Created HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error

delete

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/aaagroup_auditnslogpolicy_binding/groupname_value<String>

Query-parameters:

args

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/aaagroup_auditnslogpolicy_binding/groupname_value<String>?**args=policy:<String_value>,type:<String_value>**

HTTP Method:DELETE

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue>

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error

get

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/aaagroup_auditnslogpolicy_binding/groupname_value<String>

Query-parameters:

filter

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/aaagroup_auditnslogpolicy_binding/groupname_value<String>?**filter=property-name1:property-value1,property-name2:property-value2**

Use this query-parameter to get the filtered set of aaagroup_auditnslogpolicy_binding resources configured on Citrix ADC. Filtering can be done on any of the properties of the resource.

pagination

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/aaagroup_auditnslogpolicy_binding/groupname_value<String>?**pagesize=#no;pageno=#no**

Use this query-parameter to get the aaagroup_auditnslogpolicy_binding resources in chunks.

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "aaagroup_auditnslogpolicy_binding": [ {

"priority":<Double_value>,

"gotopriorityexpression":<String_value>,

"policy":<String_value>,

"type":<String_value>,

"groupname":<String_value>,

"acttype":<Double_value>

}]}
```

get (all)

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/aaagroup_auditnslogpolicy_binding

Query-parameters:

bulkbindings

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/aaagroup_auditnslogpolicy_binding?**bulkbindings=yes**

NITRO allows you to fetch bindings in bulk.

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "aaagroup_auditnslogpolicy_binding": [ {

"priority":<Double_value>,

"gotopriorityexpression":<String_value>,

"policy":<String_value>,

"type":<String_value>,

"groupname":<String_value>,
```

```
"acttype":<Double_value>

}]}
```

count

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/aaagroup_auditnslogpolicy_binding/groupname_value<String>?**count=yes**

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{"aaagroup_auditnslogpolicy_binding": [ { "__count": "#no"} ] }
```

aaagroup_auditsyslogpolicy_binding

Binding object showing the auditsyslogpolicy that can be bound to aaagroup.

Properties

(click to see Operations)

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
priority	<Double>	Read-write	Integer specifying the priority of the

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
			policy. A lower number indicates a higher priority. Policies are evaluated in the order of their priority numbers. Maximum value for default syntax policies is 2147483647 and for classic policies is 64000. Minimum value = 0 Maximum value = 2147483647
gotopriorityexpression	<String>	Read-write	Expression or other value specifying the next policy to evaluate if the current policy evaluates to TRUE. Specify one of the following values: * NEXT - Evaluate the policy with the next higher priority number. * END - End policy evaluation. * USE_INVOCATION_RESULT - Applicable if this policy invokes another policy label. If the final goto in the invoked policy label has a value of END, the evaluation stops. If the final goto is anything other than END, the current

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
			policy label performs a NEXT. * An expression that evaluates to a number. If you specify an expression, the number to which it evaluates determines the next policy to evaluate, as follows: * If the expression evaluates to a higher numbered priority, the policy with that priority is evaluated next. * If the expression evaluates to the priority of the current policy, the policy with the next higher numbered priority is evaluated next. * If the expression evaluates to a number that is larger than the largest numbered priority, policy evaluation ends. An UNDEF event is triggered if: * The expression is invalid. * The expression evaluates to a priority number that is numerically lower than the current policy's priority. * The expression evaluates

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
			to a priority number that is between the current policy's priority number (say, 30) and the highest priority number (say, 100), but does not match any configured priority number (for example, the expression evaluates to the number 85). This example assumes that the priority number increments by 10 for every successive policy, and therefore a priority number of 85 does not exist in the policy label.
policy	<String>	Read-write	The policy name.
type	<String>	Read-write	Bindpoint to which the policy is bound. Default value: REQUEST Possible values = REQUEST, UDP_REQUEST, DNS_REQUEST, ICMP_REQUEST
groupname	<String>	Read-write	Name of the group that you are binding. Minimum length = 1

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
acttype	<Double>	Read-only	.
__count	<Double>	Read-only	count parameter

Operations

(click to see Properties)

ADD:| DELETE:| GET| GET (ALL)| COUNT

Some options that you can use for each operations:

- **Getting warnings in response:**NITRO allows you to get warnings in an operation by specifying the "warning" query parameter as "yes". For example, to get warnings while connecting to the Citrix ADC appliance, the URL is as follows:

  http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/login?warning=yes

  If any, the warnings are displayed in the response payload with the HTTP code "209 X-NITRO-WARNING".
- **Authenticated access for individual NITRO operations:**NITRO allows you to logon to the Citrix ADC appliance to perform individual operations. You can use this option instead of creating a NITRO session (using the login object) and then using that session to perform all operations,

  To do this, you must specify the username and password in the request header of the NITRO request as follows:

  X-NITRO-USER:<username>

  X-NITRO-PASS:<password>

  Note: In such cases, make sure that the request header DOES not include the following:

  Cookie:NITRO_AUTH_TOKEN=<tokenvalue>

`*Note:`*

Mandatory parameters are marked in redand placeholder content is marked in <green>.

add

URL:http://<Citrix-ADC-IP-address(NSIP)/nitro/v1/config/aaagroup_auditsyslogpolicy_binding

HTTP Method:PUT

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Content-Type:application/json

Request Payload:

```
{

"aaagroup_auditsyslogpolicy_binding":{

<b>"groupname":<String_value>,

</b>"policy":<String_value>,

"priority":<Double_value>,

"type":<String_value>,

"gotopriorityexpression":<String_value>

}}
```

Response:

HTTP Status Code on Success: 201 Created HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error

delete

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/aaagroup_auditsyslogpolicy_binding/groupname_value<String>

Query-parameters:

args

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/aaagroup_auditsyslogpolicy_binding/groupname_value<String>?**args=policy:<String_value>,type:<String_value>**

HTTP Method:DELETE

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue>

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error

get

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/aaagroup_auditsyslogpolicy_binding/groupname_value<String>

Query-parameters:

filter

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/aaagroup_auditsyslogpolicy_binding/groupname_value<String>?**filter=property-name1:property-value1,property-name2:property-value2**

Use this query-parameter to get the filtered set of aaagroup_auditsyslogpolicy_binding resources configured on Citrix ADC. Filtering can be done on any of the properties of the resource.

pagination

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/aaagroup_auditsyslogpolicy_binding/groupname_value<String>?**pagesize=#no;pageno=#no**

Use this query-parameter to get the aaagroup_auditsyslogpolicy_binding resources in chunks.

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "aaagroup_auditsyslogpolicy_binding": [ {

"priority":<Double_value>,

"gotopriorityexpression":<String_value>,

"policy":<String_value>,

"type":<String_value>,

"groupname":<String_value>,

"acttype":<Double_value>

}]}
```

get (all)

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/aaagroup_auditsyslogpolicy_binding

Query-parameters:

bulkbindings

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/aaagroup_auditsyslogpolicy_binding?**bulkbindings=yes**

NITRO allows you to fetch bindings in bulk.

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "aaagroup_auditsyslogpolicy_binding": [ {

"priority":<Double_value>,

"gotopriorityexpression":<String_value>,

"policy":<String_value>,

"type":<String_value>,

"groupname":<String_value>,

"acttype":<Double_value>

}]}
```

count

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/aaagroup_auditsyslogpolicy_binding/groupname_value<String>?**count=yes**

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{"aaagroup_auditsyslogpolicy_binding": [ { "__count": "#no"} ] }
```

aaagroup_authorizationpolicy_binding

Binding object showing the authorizationpolicy that can be bound to aaagroup.

Properties

(click to see Operations)

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
priority	<Double>	Read-write	Integer specifying the priority of the policy. A lower number indicates a higher priority. Policies are evaluated in the order of their priority numbers. Maximum value for default syntax policies is 2147483647 and for classic policies is 64000. Minimum value = 0 Maximum value = 2147483647

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
gotopriorityexpression	<String>	Read-write	Expression specifying the priority of the next policy which will get evaluated if the current policy rule evaluates to TRUE.
policy	<String>	Read-write	The policy name.
type	<String>	Read-write	Bindpoint to which the policy is bound. Default value: REQUEST Possible values = REQUEST, UDP_REQUEST, DNS_REQUEST, ICMP_REQUEST
groupname	<String>	Read-write	Name of the group that you are binding. Minimum length = 1
acttype	<Double>	Read-only	.
__count	<Double>	Read-only	count parameter

Operations

(click to see Properties)

ADD:| DELETE:| GET| GET (ALL)| COUNT

Some options that you can use for each operations:

- **Getting warnings in response:**NITRO allows you to get warnings in an operation by specifying the "warning" query parameter as "yes". For example, to get warnings while connecting to the Citrix ADC appliance, the URL is as follows:

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/login?warning=yes

If any, the warnings are displayed in the response payload with the HTTP code "209 X-NITRO-WARNING".

- **Authenticated access for individual NITRO operations:**NITRO allows you to logon to the Citrix ADC appliance to perform individual operations. You can use this option instead of creating a NITRO session (using the login object) and then using that session to perform all operations,

  To do this, you must specify the username and password in the request header of the NITRO request as follows:

  X-NITRO-USER:<username>

  X-NITRO-PASS:<password>

  Note: In such cases, make sure that the request header DOES not include the following:

  Cookie:NITRO_AUTH_TOKEN=<tokenvalue>

`*Note:`*

Mandatory parameters are marked in redand placeholder content is marked in <green>.

add

URL:http://<Citrix-ADC-IP-address(NSIP)/nitro/v1/config/aaagroup_authorizationpolicy_binding

HTTP Method:PUT

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Content-Type:application/json

Request Payload:

```
{

"aaagroup_authorizationpolicy_binding":{

<b>"groupname":<String_value>,

</b>"policy":<String_value>,
```

```
"priority":<Double_value>,

"type":<String_value>,

"gotopriorityexpression":<String_value>

}}
```

Response:

HTTP Status Code on Success: 201 Created HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error

delete

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/aaagroup_authorizationpolicy_binding/groupname_value<String>

Query-parameters:

args

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/aaagroup_authorizationpolicy_binding/groupname_value<String>?**args=policy:<String_value>,type:<String_value>**

HTTP Method:DELETE

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue>

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error

get

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/aaagroup_authorizationpolicy_binding/groupname_value<String>

Query-parameters:

filter

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/aaagroup_authorizationpolicy_binding/groupname_value<String>?**filter=property-name1:property-value1,property-name2:property-value2**

Use this query-parameter to get the filtered set of aaagroup_authorizationpolicy_binding resources configured on Citrix ADC. Filtering can be done on any of the properties of the resource.

pagination

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/aaagroup_authorizationpolicy_binding/groupname_value<String>?**pagesize=#no;pageno=#no**

Use this query-parameter to get the aaagroup_authorizationpolicy_binding resources in chunks.

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "aaagroup_authorizationpolicy_binding": [ {

"priority":<Double_value>,

"gotopriorityexpression":<String_value>,

"policy":<String_value>,
```

```
"type":<String_value>,

"groupname":<String_value>,

"acttype":<Double_value>

}]}
```

get (all)

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/aaagroup_authorizationpolicy_binding

Query-parameters:

bulkbindings

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/aaagroup_authorizationpolicy_binding?**bulkbindings=yes**

NITRO allows you to fetch bindings in bulk.

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "aaagroup_authorizationpolicy_binding": [ {

"priority":<Double_value>,
```

```
"gotopriorityexpression":<String_value>,

"policy":<String_value>,

"type":<String_value>,

"groupname":<String_value>,

"acttype":<Double_value>

}]}
```

count

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/aaagroup_authorizationpolicy_binding/groupname_value<String>?**count=yes**

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{"aaagroup_authorizationpolicy_binding": [ { "__count": "#no"} ] }
```

aaagroup_binding

Binding object which returns the resources bound to aaagroup.

Properties

(click to see Operations)

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
groupname	<String>	Read-write	Name of the group. Minimum length = 1
aaagroup_auditnslogpolicy_binding	<aaagroup_auditnslogpolicy_binding[]>	Read-only	auditnslogpolicy that can be bound to aaagroup.
aaagroup_intranetip6_binding	<aaagroup_intranetip6_binding[]>	Read-only	intranetip6 that can be bound to aaagroup.
aaagroup_vpnsessionpolicy_binding	<aaagroup_vpnsessionpolicy_binding[]>	Read-only	vpnsessionpolicy that can be bound to aaagroup.
aaagroup_vpnurlpolicy_binding	<aaagroup_vpnurlpolicy_binding[]>	Read-only	vpnurlpolicy that can be bound to aaagroup.
aaagroup_intranetip_binding	<aaagroup_intranetip_binding[]>	Read-only	intranetip that can be bound to aaagroup.
aaagroup_aaauser_binding	<aaagroup_aaauser_binding[]>	Read-only	aaauser that can be bound to aaagroup.
aaagroup_vpntrafficpolicy_binding	<aaagroup_vpntrafficpolicy_binding[]>	Read-only	vpntrafficpolicy that can be bound to aaagroup.
aaagroup_vpnintranetapplication_binding	<aaagroup_vpnintranetapplication_binding[]>	Read-only	vpnintranetapplication that can be bound to aaagroup.
aaagroup_authorizationpolicy_binding	<aaagroup_authorizationpolicy_binding[]>	Read-only	authorizationpolicy that can be bound to aaagroup.

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
aaagroup_auditsyslogpolicy_binding	<aaagroup_auditsyslogpolicy_binding[]>	Read-only	auditsyslogpolicy that can be bound to aaagroup.
aaagroup_vpnurl_binding	<aaagroup_vpnurl_binding[]>	Read-only	vpnurl that can be bound to aaagroup.
aaagroup_tmsessionpolicy_binding	<aaagroup_tmsessionpolicy_binding[]>	Read-only	tmsessionpolicy that can be bound to aaagroup.

Operations

(click to see Properties)

GET| GET (ALL)

Some options that you can use for each operations:

- **Getting warnings in response:**NITRO allows you to get warnings in an operation by specifying the "warning" query parameter as "yes". For example, to get warnings while connecting to the Citrix ADC appliance, the URL is as follows:

  http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/login?warning=yes

  If any, the warnings are displayed in the response payload with the HTTP code "209 X-NITRO-WARNING".

- **Authenticated access for individual NITRO operations:**NITRO allows you to logon to the Citrix ADC appliance to perform individual operations. You can use this option instead of creating a NITRO session (using the login object) and then using that session to perform all operations,

  To do this, you must specify the username and password in the request header of the NITRO request as follows:

  X-NITRO-USER:<username>

  X-NITRO-PASS:<password>

  Note: In such cases, make sure that the request header DOES not include the following:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue>

`*Note:` *

Mandatory parameters are marked in redand placeholder content is marked in <green>.

get

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/aaagroup_binding/groupname_value<String>

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "aaagroup_binding": [ {

"groupname":<String_value>,

"aaagroup_vpntrafficpolicy_binding":<aaagroup_vpntrafficpolicy_binding[]_value>,

"aaagroup_authorizationpolicy_binding":<aaagroup_authorizationpolicy_binding[]_value>,

"aaagroup_intranetip_binding":<aaagroup_intranetip_binding[]_value>,

"aaagroup_vpnurlpolicy_binding":<aaagroup_vpnurlpolicy_binding[]_value>,

"aaagroup_intranetip6_binding":<aaagroup_intranetip6_binding[]_value>,
```

```
"aaagroup_auditnslogpolicy_binding":<aaagroup_auditnslogpolicy_binding[]_value>,

"aaagroup_vpnurl_binding":<aaagroup_vpnurl_binding[]_value>,

"aaagroup_auditsyslogpolicy_binding":<aaagroup_auditsyslogpolicy_binding[]_value>,

"aaagroup_aaauser_binding":<aaagroup_aaauser_binding[]_value>,

"aaagroup_tmsessionpolicy_binding":<aaagroup_tmsessionpolicy_binding[]_value>,

"aaagroup_vpnsessionpolicy_binding":<aaagroup_vpnsessionpolicy_binding[]_value>,

"aaagroup_vpnintranetapplication_binding":<aaagroup_vpnintranetapplication_binding[]_value>

}]}
```

get (all)

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/aaagroup_binding

Query-parameters:

bulkbindings

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/aaagroup_binding?**bulkbindings=yes**

NITRO allows you to fetch bindings in bulk.

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "aaagroup_binding": [ {

"groupname":<String_value>,

"aaagroup_vpntrafficpolicy_binding":<aaagroup_vpntrafficpolicy_binding[]_v
alue>,

"aaagroup_authorizationpolicy_binding":<aaagroup_authorizationpolicy_bindi
ng[]_value>,

"aaagroup_intranetip_binding":<aaagroup_intranetip_binding[]_value>,

"aaagroup_vpnurlpolicy_binding":<aaagroup_vpnurlpolicy_binding[]_value>,

"aaagroup_intranetip6_binding":<aaagroup_intranetip6_binding[]_value>,

"aaagroup_auditnslogpolicy_binding":<aaagroup_auditnslogpolicy_binding[]_v
alue>,

"aaagroup_vpnurl_binding":<aaagroup_vpnurl_binding[]_value>,

"aaagroup_auditsyslogpolicy_binding":<aaagroup_auditsyslogpolicy_binding[]
_value>,

"aaagroup_aaauser_binding":<aaagroup_aaauser_binding[]_value>,

"aaagroup_tmsessionpolicy_binding":<aaagroup_tmsessionpolicy_binding[]_val
ue>,

"aaagroup_vpnsessionpolicy_binding":<aaagroup_vpnsessionpolicy_binding[]_v
alue>,

"aaagroup_vpnintranetapplication_binding":<aaagroup_vpnintranetapplication
```

```
_binding[]_value>

}]}
```

aaagroup_intranetip6_binding

Binding object showing the intranetip6 that can be bound to aaagroup.

Properties

(click to see Operations)

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
intranetip6	<String>	Read-write	The Intranet IP6(s) bound to the group.
gotopriorityexpression	<String>	Read-write	Expression or other value specifying the next policy to evaluate if the current policy evaluates to TRUE. Specify one of the following values: * NEXT - Evaluate the policy with the next higher priority number. * END - End policy evaluation. * USE_INVOCATION_RESULT - Applicable if this policy invokes another policy label. If the final goto in the invoked policy label has a value of END, the evaluation stops. If the final goto is anything other than END, the current policy label performs

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
			a NEXT. * An expression that evaluates to a number. If you specify an expression, the number to which it evaluates determines the next policy to evaluate, as follows: * If the expression evaluates to a higher numbered priority, the policy with that priority is evaluated next. * If the expression evaluates to the priority of the current policy, the policy with the next higher numbered priority is evaluated next. * If the expression evaluates to a number that is larger than the largest numbered priority, policy evaluation ends. An UNDEF event is triggered if: * The expression is invalid. * The expression evaluates to a priority number that is numerically lower than the current policy's priority. * The expression evaluates to a priority number

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
			that is between the current policy's priority number (say, 30) and the highest priority number (say, 100), but does not match any configured priority number (for example, the expression evaluates to the number 85). This example assumes that the priority number increments by 10 for every successive policy, and therefore a priority number of 85 does not exist in the policy label.
numaddr	<Double>	Read-write	Numbers of ipv6 address bound starting with intranetip6.
groupname	<String>	Read-write	Name of the group that you are binding. Minimum length = 1
__count	<Double>	Read-write	count parameter

Operations

(click to see Properties)

ADD:| DELETE:| GET| GET (ALL)| COUNT

Some options that you can use for each operations:

- **Getting warnings in response:**NITRO allows you to get warnings in an operation by specifying the "warning" query parameter as "yes". For example, to get warnings while connecting to the Citrix ADC appliance, the URL is as follows:

  http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/login?warning=yes

  If any, the warnings are displayed in the response payload with the HTTP code "209 X-NITRO-WARNING".

- **Authenticated access for individual NITRO operations:**NITRO allows you to logon to the Citrix ADC appliance to perform individual operations. You can use this option instead of creating a NITRO session (using the login object) and then using that session to perform all operations,

  To do this, you must specify the username and password in the request header of the NITRO request as follows:

  X-NITRO-USER:<username>

  X-NITRO-PASS:<password>

  Note: In such cases, make sure that the request header DOES not include the following:

  Cookie:NITRO_AUTH_TOKEN=<tokenvalue>

`*Note:`*

Mandatory parameters are marked in redand placeholder content is marked in <green>.

add

URL:http://<Citrix-ADC-IP-address(NSIP)/nitro/v1/config/aaagroup_intranetip6_binding

HTTP Method:PUT

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Content-Type:application/json

Request Payload:

```
{
```

```
"aaagroup_intranetip6_binding":{

<b>"groupname":<String_value>,

</b>"gotopriorityexpression":<String_value>,

"intranetip6":<String_value>,

"numaddr":<Double_value>

}}
```

Response:

HTTP Status Code on Success: 201 Created HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error

delete

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/aaagroup_intranetip6_binding/groupname_value<String>

Query-parameters:

args

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/aaagroup_intranetip6_binding/groupname_value<String>?**args=intranetip6:<String_value>,numaddr:<Double_value>**

HTTP Method:DELETE

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue>

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error

get

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/aaagroup_intranetip6_binding/groupname_value<String>

Query-parameters:

filter

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/aaagroup_intranetip6_binding/groupname_value<String>?**filter=property-name1:property-value1,property-name2:property-value2**

Use this query-parameter to get the filtered set of aaagroup_intranetip6_binding resources configured on Citrix ADC. Filtering can be done on any of the properties of the resource.

pagination

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/aaagroup_intranetip6_binding/groupname_value<String>?**pagesize=#no;pageno=#no**

Use this query-parameter to get the aaagroup_intranetip6_binding resources in chunks.

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "aaagroup_intranetip6_binding": [ {

"intranetip6":<String_value>,
```

```
"gotopriorityexpression":<String_value>,

"numaddr":<Double_value>,

"groupname":<String_value>

}]}
```

get (all)

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/aaagroup_intranetip6_binding

Query-parameters:

bulkbindings

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/aaagroup_intranetip6_binding?**bulkbindings=yes**

NITRO allows you to fetch bindings in bulk.

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "aaagroup_intranetip6_binding": [ {

"intranetip6":<String_value>,
```

```
"gotopriorityexpression":<String_value>,

"numaddr":<Double_value>,

"groupname":<String_value>

}]}
```

count

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/aaagroup_intranetip6_binding/groupname_value<String>?**count=yes**

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{"aaagroup_intranetip6_binding": [ { "__count": "#no"} ] }
```

aaagroup_intranetip_binding

Binding object showing the intranetip that can be bound to aaagroup.

Properties

(click to see Operations)

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
gotopriorityexpression	<String>	Read-write	Expression or other value specifying the next policy to evaluate if the current policy evaluates to TRUE. Specify one of the following values: * NEXT - Evaluate the policy with the next higher priority number. * END - End policy evaluation. * USE_INVOCATION_RESULT - Applicable if this policy invokes another policy label. If the final goto in the invoked policy label has a value of END, the evaluation stops. If the final goto is anything other than END, the current policy label performs a NEXT. * An expression that evaluates to a number. If you specify an expression, the number to which it evaluates determines the next policy to evaluate, as follows: * If the expression evaluates to a higher numbered priority, the policy with that priority is evaluated

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
			next. * If the expression evaluates to the priority of the current policy, the policy with the next higher numbered priority is evaluated next. * If the expression evaluates to a number that is larger than the largest numbered priority, policy evaluation ends. An UNDEF event is triggered if: * The expression is invalid. * The expression evaluates to a priority number that is numerically lower than the current policy's priority. * The expression evaluates to a priority number that is between the current policy's priority number (say, 30) and the highest priority number (say, 100), but does not match any configured priority number (for example, the expression evaluates to the number 85). This example assumes that the priority number increments by 10 for

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
			every successive policy, and therefore a priority number of 85 does not exist in the policy label.
intranetip	<String>	Read-write	The Intranet IP(s) bound to the group.
groupname	<String>	Read-write	Name of the group that you are binding. Minimum length = 1
netmask	<String>	Read-write	The netmask for the Intranet IP.
__count	<Double>	Read-write	count parameter

Operations

(click to see Properties)

ADD:| DELETE:| GET| GET (ALL)| COUNT

Some options that you can use for each operations:

- **Getting warnings in response:**NITRO allows you to get warnings in an operation by specifying the "warning" query parameter as "yes". For example, to get warnings while connecting to the Citrix ADC appliance, the URL is as follows:

  http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/login?warning=yes

  If any, the warnings are displayed in the response payload with the HTTP code "209 X-NITRO-WARNING".

- **Authenticated access for individual NITRO operations:**NITRO allows you to logon to the Citrix ADC appliance to perform individual operations. You can use this option instead of creating a NITRO session (using the login object) and then using that session to perform all operations,

  To do this, you must specify the username and password in the request header of the NITRO request as follows:

X-NITRO-USER:<username>

X-NITRO-PASS:<password>

Note: In such cases, make sure that the request header DOES not include the following:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue>

`*Note:`*

Mandatory parameters are marked in redand placeholder content is marked in <green>.

add

URL:http://<Citrix-ADC-IP-address(NSIP)/nitro/v1/config/aaagroup_intranetip_binding

HTTP Method:PUT

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Content-Type:application/json

Request Payload:

```
{
"aaagroup_intranetip_binding":{
<b>"groupname":<String_value>,
</b>"intranetip":<String_value>,
"netmask":<String_value>,
"gotopriorityexpression":<String_value>
}}
```

Response:

HTTP Status Code on Success: 201 Created HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error

delete

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/aaagroup_intranetip_binding/groupname_value<String>

Query-parameters:

args

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/aaagroup_intranetip_binding/groupname_value<String>?**args=intranetip:<String_value>,netmask:<String_value>**

HTTP Method:DELETE

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue>

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error

get

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/aaagroup_intranetip_binding/groupname_value<String>

Query-parameters:

filter

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/aaagroup_intranetip_binding/groupname_value<String>?**filter=property-name1:property-value1,property-name2:property-value2**

Use this query-parameter to get the filtered set of aaagroup_intranetip_binding resources configured on Citrix ADC. Filtering can be done on any of the properties of the resource.

pagination

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/aaagroup_intranetip_binding/groupname_value<String>?**pagesize=#no;pageno=#no**

Use this query-parameter to get the aaagroup_intranetip_binding resources in chunks.

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "aaagroup_intranetip_binding": [ {

"gotopriorityexpression":<String_value>,

"intranetip":<String_value>,

"groupname":<String_value>,

"netmask":<String_value>

}]}
```

get (all)

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/aaagroup_intranetip_binding

Query-parameters:

bulkbindings

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/aaagroup_intranetip_binding?**bulkbindings=yes**

NITRO allows you to fetch bindings in bulk.

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "aaagroup_intranetip_binding": [ {

"gotopriorityexpression":<String_value>,

"intranetip":<String_value>,

"groupname":<String_value>,

"netmask":<String_value>

}]}
```

count

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/aaagroup_intranetip_binding/groupname_value<String>?**count=yes**

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{"aaagroup_intranetip_binding": [ { "__count": "#no"} ] }
```

aaagroup_tmsessionpolicy_binding

Binding object showing the tmsessionpolicy that can be bound to aaagroup.

Properties

(click to see Operations)

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
priority	<Double>	Read-write	Integer specifying the priority of the policy. A lower number indicates a higher priority. Policies are evaluated in the order of their priority numbers. Maximum value for default syntax policies is 2147483647 and for classic policies is 64000. Minimum value = 0

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
			Maximum value = 2147483647
gotopriorityexpression	<String>	Read-write	Expression specifying the priority of the next policy which will get evaluated if the current policy rule evaluates to TRUE.
policy	<String>	Read-write	The policy name.
type	<String>	Read-write	Bindpoint to which the policy is bound. Default value: REQUEST Possible values = REQUEST, UDP_REQUEST, DNS_REQUEST, ICMP_REQUEST
groupname	<String>	Read-write	Name of the group that you are binding. Minimum length = 1
acttype	<Double>	Read-only	.
__count	<Double>	Read-only	count parameter

Operations

(click to see Properties)

ADD:| DELETE:| GET| GET (ALL)| COUNT

Some options that you can use for each operations:

- **Getting warnings in response:**NITRO allows you to get warnings in an operation by specifying the "warning" query parameter as "yes". For example, to get warnings while connecting to the Citrix ADC appliance, the URL is as follows:

  http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/login?warning=yes

  If any, the warnings are displayed in the response payload with the HTTP code "209 X-NITRO-WARNING".

- **Authenticated access for individual NITRO operations:**NITRO allows you to logon to the Citrix ADC appliance to perform individual operations. You can use this option instead of creating a NITRO session (using the login object) and then using that session to perform all operations,

  To do this, you must specify the username and password in the request header of the NITRO request as follows:

  X-NITRO-USER:<username>

  X-NITRO-PASS:<password>

  Note: In such cases, make sure that the request header DOES not include the following:

  Cookie:NITRO_AUTH_TOKEN=<tokenvalue>

`*Note:`*

Mandatory parameters are marked in redand placeholder content is marked in <green>.

add

URL:http://<Citrix-ADC-IP-address(NSIP)/nitro/v1/config/aaagroup_tmsessionpolicy_binding

HTTP Method:PUT

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Content-Type:application/json

Request Payload:

```
{
```

```
"aaagroup_tmsessionpolicy_binding":{

<b>"groupname":<String_value>,

</b>"policy":<String_value>,

"priority":<Double_value>,

"type":<String_value>,

"gotopriorityexpression":<String_value>

}}
```

Response:

HTTP Status Code on Success: 201 Created HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error

delete

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/aaagroup_tmsessionpolicy_binding/groupname_value<String>

Query-parameters:

args

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/aaagroup_tmsessionpolicy_binding/groupname_value<String>?**args=policy:<String_value>,type:<String_value>**

HTTP Method:DELETE

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue>

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error

get

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/aaagroup_tmsessionpolicy_binding/groupname_value<String>

Query-parameters:

filter

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/aaagroup_tmsessionpolicy_binding/groupname_value<String>?**filter=property-name1:property-value1,property-name2:property-value2**

Use this query-parameter to get the filtered set of aaagroup_tmsessionpolicy_binding resources configured on Citrix ADC. Filtering can be done on any of the properties of the resource.

pagination

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/aaagroup_tmsessionpolicy_binding/groupname_value<String>?**pagesize=#no;pageno=#no**

Use this query-parameter to get the aaagroup_tmsessionpolicy_binding resources in chunks.

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "aaagroup_tmsessionpolicy_binding": [ {

"priority":<Double_value>,

"gotopriorityexpression":<String_value>,

"policy":<String_value>,

"type":<String_value>,

"groupname":<String_value>,

"acttype":<Double_value>

}]}
```

get (all)

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/aaagroup_tmsessionpolicy_binding

Query-parameters:

bulkbindings

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/aaagroup_tmsessionpolicy_binding?**bulkbindings=yes**

NITRO allows you to fetch bindings in bulk.

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "aaagroup_tmsessionpolicy_binding": [ {

"priority":<Double_value>,

"gotopriorityexpression":<String_value>,

"policy":<String_value>,

"type":<String_value>,

"groupname":<String_value>,

"acttype":<Double_value>

}]}
```

count

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/aaagroup_tmsessionpolicy_binding/groupname_value<String>?**count=yes**

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{"aaagroup_tmsessionpolicy_binding": [ { "__count": "#no"} ] }
```

aaagroup_vpnintranetapplication_binding

Binding object showing the vpnintranetapplication that can be bound to aaagroup.

Properties

(click to see Operations)

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
gotopriorityexpression	<String>	Read-write	Expression or other value specifying the next policy to evaluate if the current policy evaluates to TRUE. Specify one of the following values: * NEXT - Evaluate the policy with the next higher priority number. * END - End policy evaluation. * USE_INVOCATION_RESULT - Applicable if this policy invokes another policy label. If the final goto in the invoked policy label has a value of END, the evaluation stops. If the final goto is anything other than END, the current policy label performs a NEXT. * An expression that

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
			evaluates to a number. If you specify an expression, the number to which it evaluates determines the next policy to evaluate, as follows: * If the expression evaluates to a higher numbered priority, the policy with that priority is evaluated next. * If the expression evaluates to the priority of the current policy, the policy with the next higher numbered priority is evaluated next. * If the expression evaluates to a number that is larger than the largest numbered priority, policy evaluation ends. An UNDEF event is triggered if: * The expression is invalid. * The expression evaluates to a priority number that is numerically lower than the current policy's priority. * The expression evaluates to a priority number that is between the current policy's

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
			priority number (say, 30) and the highest priority number (say, 100), but does not match any configured priority number (for example, the expression evaluates to the number 85). This example assumes that the priority number increments by 10 for every successive policy, and therefore a priority number of 85 does not exist in the policy label.
groupname	<String>	Read-write	Name of the group that you are binding. Minimum length = 1
intranetapplication	<String>	Read-write	Bind the group to the specified intranet VPN application.
acttype	<Double>	Read-only	.
__count	<Double>	Read-only	count parameter

Operations

(click to see Properties)

ADD:| DELETE:| GET| GET (ALL)| COUNT

Some options that you can use for each operations:

- **Getting warnings in response:**NITRO allows you to get warnings in an operation by specifying the "warning" query parameter as "yes". For example, to get warnings while connecting to the Citrix ADC appliance, the URL is as follows:

  http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/login?warning=yes

  If any, the warnings are displayed in the response payload with the HTTP code "209 X-NITRO-WARNING".

- **Authenticated access for individual NITRO operations:**NITRO allows you to logon to the Citrix ADC appliance to perform individual operations. You can use this option instead of creating a NITRO session (using the login object) and then using that session to perform all operations,

  To do this, you must specify the username and password in the request header of the NITRO request as follows:

  X-NITRO-USER:<username>

  X-NITRO-PASS:<password>

  Note: In such cases, make sure that the request header DOES not include the following:

  Cookie:NITRO_AUTH_TOKEN=<tokenvalue>

`*Note:`*

Mandatory parameters are marked in redand placeholder content is marked in <green>.

add

URL:http://<Citrix-ADC-IP-address(NSIP)/nitro/v1/config/aaagroup_vpnintranetapplication_binding

HTTP Method:PUT

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Content-Type:application/json

Request Payload:

```
{
```

```
"aaagroup_vpnintranetapplication_binding":{

<b>"groupname":<String_value>,

</b>"intranetapplication":<String_value>,

"gotopriorityexpression":<String_value>

}}
```

Response:

HTTP Status Code on Success: 201 Created HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error

delete

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/aaagroup_vpnintranetapplication_binding/groupname_value<String>

Query-parameters:

args

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/aaagroup_vpnintranetapplication_binding/groupname_value<String>?**args=intranetapplication:<String_value>**

HTTP Method:DELETE

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue>

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error

get

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/aaagroup_vpnintranetapplication_binding/groupname_value<String>

Query-parameters:

filter

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/aaagroup_vpnintranetapplication_binding/groupname_value<String>?**filter=property-name1:property-value1,property-name2:property-value2**

Use this query-parameter to get the filtered set of aaagroup_vpnintranetapplication_binding resources configured on Citrix ADC. Filtering can be done on any of the properties of the resource.

pagination

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/aaagroup_vpnintranetapplication_binding/groupname_value<String>?**pagesize=#no;pageno=#no**

Use this query-parameter to get the aaagroup_vpnintranetapplication_binding resources in chunks.

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "aaagroup_vpnintranetapplication_binding": [ {

"gotopriorityexpression":<String_value>,
```

```
"groupname":<String_value>,

"intranetapplication":<String_value>,

"acttype":<Double_value>

}]}
```

get (all)

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/aaagroup_vpnintranetapplication_binding

Query-parameters:

bulkbindings

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/aaagroup_vpnintranetapplication_binding?**bulkbindings=yes**

NITRO allows you to fetch bindings in bulk.

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "aaagroup_vpnintranetapplication_binding": [ {

"gotopriorityexpression":<String_value>,
```

```
"groupname":<String_value>,

"intranetapplication":<String_value>,

"acttype":<Double_value>

}]}
```

count

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/aaagroup_vpnintranetapplication_binding/groupname_value<String>?**count=yes**

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{"aaagroup_vpnintranetapplication_binding": [ { "__count": "#no"} ] }
```

aaagroup_vpnsessionpolicy_binding

Binding object showing the vpnsessionpolicy that can be bound to aaagroup.

Properties

(click to see Operations)

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
priority	<Double>	Read-write	Integer specifying the priority of the policy. A lower number indicates a higher priority. Policies are evaluated in the order of their priority numbers. Maximum value for default syntax policies is 2147483647 and for classic policies is 64000. Minimum value = 0 Maximum value = 2147483647
gotopriorityexpression	<String>	Read-write	Expression specifying the priority of the next policy which will get evaluated if the current policy rule evaluates to TRUE.
policy	<String>	Read-write	The policy name.
type	<String>	Read-write	Bindpoint to which the policy is bound. Default value: REQUEST Possible values = REQUEST, UDP_REQUEST, DNS_REQUEST, ICMP_REQUEST

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
groupname	<String>	Read-write	Name of the group that you are binding. Minimum length = 1
acttype	<Double>	Read-only	.
__count	<Double>	Read-only	count parameter

Operations

(click to see Properties)

ADD:| DELETE:| GET| GET (ALL)| COUNT

Some options that you can use for each operations:

- **Getting warnings in response:**NITRO allows you to get warnings in an operation by specifying the "warning" query parameter as "yes". For example, to get warnings while connecting to the Citrix ADC appliance, the URL is as follows:

  http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/login?warning=yes

  If any, the warnings are displayed in the response payload with the HTTP code "209 X-NITRO-WARNING".

- **Authenticated access for individual NITRO operations:**NITRO allows you to logon to the Citrix ADC appliance to perform individual operations. You can use this option instead of creating a NITRO session (using the login object) and then using that session to perform all operations,

  To do this, you must specify the username and password in the request header of the NITRO request as follows:

  X-NITRO-USER:<username>

  X-NITRO-PASS:<password>

  Note: In such cases, make sure that the request header DOES not include the following:

  Cookie:NITRO_AUTH_TOKEN=<tokenvalue>

`*Note:` *

Mandatory parameters are marked in redand placeholder content is marked in <green>.

add

URL:http://<Citrix-ADC-IP-address(NSIP)/nitro/v1/config/aaagroup_vpnsessionpolicy_binding

HTTP Method:PUT

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Content-Type:application/json

Request Payload:

```
{

"aaagroup_vpnsessionpolicy_binding":{

<b>"groupname":<String_value>,

</b>"policy":<String_value>,

"priority":<Double_value>,

"type":<String_value>,

"gotopriorityexpression":<String_value>

}}
```

Response:

HTTP Status Code on Success: 201 Created HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error

delete

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/aaagroup_vpnsessionpolicy_binding/groupname_value<String>

Query-parameters:

args

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/aaagroup_vpnsessionpolicy_binding/groupname_value<String>?**args=policy:<String_value>,type:<String_value>**

HTTP Method:DELETE

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue>

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error

get

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/aaagroup_vpnsessionpolicy_binding/groupname_value<String>

Query-parameters:

filter

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/aaagroup_vpnsessionpolicy_binding/groupname_value<String>?**filter=property-name1:property-value1,property-name2:property-value2**

Use this query-parameter to get the filtered set of aaagroup_vpnsessionpolicy_binding resources configured on Citrix ADC. Filtering can be done on any of the properties of the resource.

pagination

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/aaagroup_vpnsessionpolicy_binding/groupname_value<String>?**pagesize=#no;pageno=#no**

Use this query-parameter to get the aaagroup_vpnsessionpolicy_binding resources in chunks.

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "aaagroup_vpnsessionpolicy_binding": [ {

"priority":<Double_value>,

"gotopriorityexpression":<String_value>,

"policy":<String_value>,

"type":<String_value>,

"groupname":<String_value>,

"acttype":<Double_value>

}]}
```

get (all)

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/aaagroup_vpnsessionpolicy_binding

Query-parameters:

bulkbindings

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/aaagroup_vpnsessionpolicy_binding?**bulkbindings=yes**

NITRO allows you to fetch bindings in bulk.

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "aaagroup_vpnsessionpolicy_binding": [ {

"priority":<Double_value>,

"gotopriorityexpression":<String_value>,

"policy":<String_value>,

"type":<String_value>,

"groupname":<String_value>,

"acttype":<Double_value>

}]}
```

count

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/aaagroup_vpnsessionpolicy_binding/groupname_value<String>?**count=yes**

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{"aaagroup_vpnsessionpolicy_binding": [ { "__count": "#no"} ] }
```

aaagroup_vpntrafficpolicy_binding

Binding object showing the vpntrafficpolicy that can be bound to aaagroup.

Properties

(click to see Operations)

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
priority	<Double>	Read-write	Integer specifying the priority of the policy. A lower number indicates a higher priority. Policies are evaluated in the order of their priority numbers. Maximum value for default syntax policies is 2147483647 and for classic policies is 64000. Minimum value = 0

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
			Maximum value = 2147483647
gotopriorityexpression	<String>	Read-write	Expression or other value specifying the next policy to evaluate if the current policy evaluates to TRUE. Specify one of the following values: * NEXT - Evaluate the policy with the next higher priority number. * END - End policy evaluation. * USE_INVOCATION_RESULT - Applicable if this policy invokes another policy label. If the final goto in the invoked policy label has a value of END, the evaluation stops. If the final goto is anything other than END, the current policy label performs a NEXT. * An expression that evaluates to a number. If you specify an expression, the number to which it evaluates determines the next policy to evaluate, as follows: * If the expression evaluates to a higher

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
			numbered priority, the policy with that priority is evaluated next. * If the expression evaluates to the priority of the current policy, the policy with the next higher numbered priority is evaluated next. * If the expression evaluates to a number that is larger than the largest numbered priority, policy evaluation ends. An UNDEF event is triggered if: * The expression is invalid. * The expression evaluates to a priority number that is numerically lower than the current policy's priority. * The expression evaluates to a priority number that is between the current policy's priority number (say, 30) and the highest priority number (say, 100), but does not match any configured priority number (for example, the expression evaluates to the number 85). This example

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
			assumes that the priority number increments by 10 for every successive policy, and therefore a priority number of 85 does not exist in the policy label.
policy	<String>	Read-write	The policy name.
type	<String>	Read-write	Bindpoint to which the policy is bound. Default value: REQUEST Possible values = REQUEST, UDP_REQUEST, DNS_REQUEST, ICMP_REQUEST
groupname	<String>	Read-write	Name of the group that you are binding. Minimum length = 1
acttype	<Double>	Read-only	.
__count	<Double>	Read-only	count parameter

Operations

(click to see Properties)

ADD:| DELETE:| GET| GET (ALL)| COUNT

Some options that you can use for each operations:

- **Getting warnings in response:**NITRO allows you to get warnings in an operation by specifying the "warning" query parameter as "yes". For example, to get warnings while connecting to the Citrix ADC appliance, the URL is as follows:

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/login?warning=yes

If any, the warnings are displayed in the response payload with the HTTP code "209 X-NITRO-WARNING".

- **Authenticated access for individual NITRO operations:**NITRO allows you to logon to the Citrix ADC appliance to perform individual operations. You can use this option instead of creating a NITRO session (using the login object) and then using that session to perform all operations,

  To do this, you must specify the username and password in the request header of the NITRO request as follows:

  X-NITRO-USER:<username>

  X-NITRO-PASS:<password>

  Note: In such cases, make sure that the request header DOES not include the following:

  Cookie:NITRO_AUTH_TOKEN=<tokenvalue>

`*Note:`*

Mandatory parameters are marked in redand placeholder content is marked in <green>.

add

URL:http://<Citrix-ADC-IP-address(NSIP)/nitro/v1/config/aaagroup_vpntrafficpolicy_binding

HTTP Method:PUT

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Content-Type:application/json

Request Payload:

```
{

"aaagroup_vpntrafficpolicy_binding":{

<b>"groupname":<String_value>,

</b>"policy":<String_value>,
```

```
"priority":<Double_value>,

"type":<String_value>,

"gotopriorityexpression":<String_value>

}}
```

Response:

HTTP Status Code on Success: 201 Created HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error

delete

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/aaagroup_vpntrafficpolicy_binding/groupname_value<String>

Query-parameters:

args

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/aaagroup_vpntrafficpolicy_binding/groupname_value<String>?**args=policy:<String_value>,type:<String_value>**

HTTP Method:DELETE

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue>

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error

get

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/aaagroup_vpntrafficpolicy_binding/groupname_value<String>

Query-parameters:

filter

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/aaagroup_vpntrafficpolicy_binding/groupname_value<String>?**filter=property-name1:property-value1,property-name2:property-value2**

Use this query-parameter to get the filtered set of aaagroup_vpntrafficpolicy_binding resources configured on Citrix ADC. Filtering can be done on any of the properties of the resource.

pagination

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/aaagroup_vpntrafficpolicy_binding/groupname_value<String>?**pagesize=#no;pageno=#no**

Use this query-parameter to get the aaagroup_vpntrafficpolicy_binding resources in chunks.

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "aaagroup_vpntrafficpolicy_binding": [ {

"priority":<Double_value>,

"gotopriorityexpression":<String_value>,

"policy":<String_value>,
```

```
"type":<String_value>,

"groupname":<String_value>,

"acttype":<Double_value>

}]}
```

get (all)

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/aaagroup_vpntrafficpolicy_binding

Query-parameters:

bulkbindings

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/aaagroup_vpntrafficpolicy_binding?**bulkbindings=yes**

NITRO allows you to fetch bindings in bulk.

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "aaagroup_vpntrafficpolicy_binding": [ {

"priority":<Double_value>,
```

```
"gotopriorityexpression":<String_value>,

"policy":<String_value>,

"type":<String_value>,

"groupname":<String_value>,

"acttype":<Double_value>

}]}
```

count

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/aaagroup_vpntrafficpolicy_binding/groupname_value<String>?**count=yes**

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{"aaagroup_vpntrafficpolicy_binding": [ { "__count": "#no"} ] }
```

aaagroup_vpnurl_binding

Binding object showing the vpnurl that can be bound to aaagroup.

Properties

(click to see Operations)

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
urlname	<String>	Read-write	The intranet url.
gotopriorityexpression	<String>	Read-write	Expression or other value specifying the next policy to evaluate if the current policy evaluates to TRUE. Specify one of the following values: * NEXT - Evaluate the policy with the next higher priority number. * END - End policy evaluation. * USE_INVOCATION_RESULT - Applicable if this policy invokes another policy label. If the final goto in the invoked policy label has a value of END, the evaluation stops. If the final goto is anything other than END, the current policy label performs a NEXT. * An expression that evaluates to a number. If you specify an expression, the number to which it evaluates determines the next policy to evaluate, as

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
			follows: * If the expression evaluates to a higher numbered priority, the policy with that priority is evaluated next. * If the expression evaluates to the priority of the current policy, the policy with the next higher numbered priority is evaluated next. * If the expression evaluates to a number that is larger than the largest numbered priority, policy evaluation ends. An UNDEF event is triggered if: * The expression is invalid. * The expression evaluates to a priority number that is numerically lower than the current policy's priority. * The expression evaluates to a priority number that is between the current policy's priority number (say, 30) and the highest priority number (say, 100), but does not match any configured priority number (for example, the

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
			expression evaluates to the number 85). This example assumes that the priority number increments by 10 for every successive policy, and therefore a priority number of 85 does not exist in the policy label.
groupname	<String>	Read-write	Name of the group that you are binding. Minimum length = 1
acttype	<Double>	Read-only	.
__count	<Double>	Read-only	count parameter

Operations

(click to see Properties)

ADD:| DELETE:| GET| GET (ALL)| COUNT

Some options that you can use for each operations:

- **Getting warnings in response:**NITRO allows you to get warnings in an operation by specifying the "warning" query parameter as "yes". For example, to get warnings while connecting to the Citrix ADC appliance, the URL is as follows:

  http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/login?warning=yes

  If any, the warnings are displayed in the response payload with the HTTP code "209 X-NITRO-WARNING".

- **Authenticated access for individual NITRO operations:**NITRO allows you to logon to the Citrix ADC appliance to perform individual operations. You can use this option instead of creating a NITRO session (using the login object) and then using that session to perform all operations,

To do this, you must specify the username and password in the request header of the NITRO request as follows:

X-NITRO-USER:<username>

X-NITRO-PASS:<password>

Note: In such cases, make sure that the request header DOES not include the following:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue>

`*Note:`*

Mandatory parameters are marked in redand placeholder content is marked in <green>.

add

URL:http://<Citrix-ADC-IP-address(NSIP)/nitro/v1/config/aaagroup_vpnurl_binding

HTTP Method:PUT

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Content-Type:application/json

Request Payload:

```
{

"aaagroup_vpnurl_binding":{

<b>"groupname":<String_value>,

</b>"urlname":<String_value>,

"gotopriorityexpression":<String_value>

}}
```

Response:

HTTP Status Code on Success: 201 Created HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error

delete

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/aaagroup_vpnurl_binding/groupname_value<String>

Query-parameters:

args

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/aaagroup_vpnurl_binding/groupname_value<String>?**args=urlname:<String_value>**

HTTP Method:DELETE

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue>

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error

get

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/aaagroup_vpnurl_binding/groupname_value<String>

Query-parameters:

filter

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/aaagroup_vpnurl_binding/groupname_value<String>?**filter=property-name1:property-value1,property-name2:property-value2**

Use this query-parameter to get the filtered set of aaagroup_vpnurl_binding resources configured on Citrix ADC. Filtering can be done on any of the properties of the resource.

pagination

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/aaagroup_vpnurl_binding/groupname_value<String>?**pagesize=#no;pageno=#no**

Use this query-parameter to get the aaagroup_vpnurl_binding resources in chunks.

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "aaagroup_vpnurl_binding": [ {

"urlname":<String_value>,

"gotopriorityexpression":<String_value>,

"groupname":<String_value>,

"acttype":<Double_value>

}]}
```

get (all)

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/aaagroup_vpnurl_binding

Query-parameters:

bulkbindings

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/aaagroup_vpnurl_binding?**bulkbindings=yes**

NITRO allows you to fetch bindings in bulk.

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "aaagroup_vpnurl_binding": [ {

"urlname":<String_value>,

"gotopriorityexpression":<String_value>,

"groupname":<String_value>,

"acttype":<Double_value>

}]}
```

count

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/aaagroup_vpnurl_binding/groupname_value<String>?**count=yes**

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{"aaagroup_vpnurl_binding": [ { "__count": "#no"} ] }
```

aaakcdaccount

Configuration for Kerberos constrained delegation account resource.

Properties

(click to see Operations)

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
kcdaccount	<String>	Read-write	The name of the KCD account. Minimum length = 1
keytab	<String>	Read-write	The path to the keytab file. If specified other parameters in this command need not be given.
realmstr	<String>	Read-write	Kerberos Realm.
delegateduser	<String>	Read-write	Username that can perform kerberos constrained delegation.
kcdpassword	<String>	Read-write	Password for Delegated User.

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
usercert	<String>	Read-write	SSL Cert (including private key) for Delegated User.
cacert	<String>	Read-write	CA Cert for UserCert or when doing PKINIT backchannel.
userrealm	<String>	Read-write	Realm of the user.
enterpriserealm	<String>	Read-write	Enterprise Realm of the user. This should be given only in certain KDC deployments where KDC expects Enterprise username instead of Principal Name.
servicespn	<String>	Read-write	Service SPN. When specified, this will be used to fetch kerberos tickets. If not specified, Citrix ADC will construct SPN using service fqdn.
principle	<String>	Read-only	SPN extracted from keytab file.
kcdspn	<String>	Read-only	Host SPN extracted from keytab file.
__count	<Double>	Read-only	count parameter

Operations

(click to see Properties)

ADD| DELETE| UPDATE| UNSET| GET (ALL)| GET| COUNT

Some options that you can use for each operations:

- **Getting warnings in response:**NITRO allows you to get warnings in an operation by specifying the "warning" query parameter as "yes". For example, to get warnings while connecting to the Citrix ADC appliance, the URL is as follows:

  http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/login?warning=yes

  If any, the warnings are displayed in the response payload with the HTTP code "209 X-NITRO-WARNING".

- **Authenticated access for individual NITRO operations:**NITRO allows you to logon to the Citrix ADC appliance to perform individual operations. You can use this option instead of creating a NITRO session (using the login object) and then using that session to perform all operations,

  To do this, you must specify the username and password in the request header of the NITRO request as follows:

  X-NITRO-USER:<username>

  X-NITRO-PASS:<password>

  Note: In such cases, make sure that the request header DOES not include the following:

  Cookie:NITRO_AUTH_TOKEN=<tokenvalue>

`*Note:`*

Mandatory parameters are marked in redand placeholder content is marked in <green>.

add

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/aaakcdaccount

HTTP Method:POST

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Content-Type:application/json

Request Payload:

```
{"aaakcdaccount":{

<b>"kcdaccount":<String_value>,

</b>"keytab":<String_value>,

"realmstr":<String_value>,

"delegateduser":<String_value>,

"kcdpassword":<String_value>,

"usercert":<String_value>,

"cacert":<String_value>,

"userrealm":<String_value>,

"enterpriserealm":<String_value>,

"servicespn":<String_value>

}}
```

Response:

HTTP Status Code on Success: 201 Created HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error

delete

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/aaakcdaccount/kcdaccount_value<String>

HTTP Method:DELETE

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue>

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error

update

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/aaakcdaccount

HTTP Method:PUT

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Content-Type:application/json

Request Payload:

```
{"aaakcdaccount":{

<b>"kcdaccount":<String_value>,

</b>"keytab":<String_value>,

"realmstr":<String_value>,

"delegateduser":<String_value>,

"kcdpassword":<String_value>,

"usercert":<String_value>,

"cacert":<String_value>,

"userrealm":<String_value>,

"enterpriserealm":<String_value>,

"servicespn":<String_value>

}}
```

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error

unset

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/aaakcdaccount?**action=unset**

HTTP Method:POST

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Content-Type:application/json

Request Payload:

```
{"aaakcdaccount":{

<b>"kcdaccount":<String_value>,

</b>"keytab":true,

"usercert":true,

"cacert":true,

"userrealm":true,

"enterpriserealm":true,

"servicespn":true

}}
```

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error

get (all)

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/aaakcdaccount

Query-parameters:

attrs

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/aaakcdaccount?**attrs=property-name1,property-name2**

Use this query parameter to specify the resource details that you want to retrieve.

filter

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/aaakcdaccount?**filter=property-name1:property-val1,property-name2:property-val2**

Use this query-parameter to get the filtered set of aaakcdaccount resources configured on Citrix ADC. Filtering can be done on any of the properties of the resource.

view

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/aaakcdaccount?**view=summary**

Note: By default, the retrieved results are displayed in detail view (?view=detail).

pagination

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/aaakcdaccount?**pagesize=#no;pageno=#no**

Use this query-parameter to get the aaakcdaccount resources in chunks.

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "aaakcdaccount": [ {

"kcdaccount":<String_value>,

"keytab":<String_value>,

"principle":<String_value>,

"kcdspn":<String_value>,

"realmstr":<String_value>,

"delegateduser":<String_value>,

"kcdpassword":<String_value>,

"usercert":<String_value>,

"cacert":<String_value>,

"userrealm":<String_value>,

"enterpriserealm":<String_value>,

"servicespn":<String_value>

}]}
```

get

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/aaakcdaccount/kcdaccount_value<String>

Query-parameters:

attrs

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/aaakcdaccount/kcdaccount_value<String>?**attrs=property-name1,property-name2**

Use this query parameter to specify the resource details that you want to retrieve.

view

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/aaakcdaccount/kcdaccount_value<String>?**view=summary**

Note: By default, the retrieved results are displayed in detail view (?view=detail).

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "aaakcdaccount": [ {

"kcdaccount":<String_value>,

"keytab":<String_value>,

"principle":<String_value>,

"kcdspn":<String_value>,

"realmstr":<String_value>,
```

```
"delegateduser":<String_value>,

"kcdpassword":<String_value>,

"usercert":<String_value>,

"cacert":<String_value>,

"userrealm":<String_value>,

"enterpriserealm":<String_value>,

"servicespn":<String_value>

}]}
```

count

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/aaakcdaccount?**count=yes**

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "aaakcdaccount": [ { "__count": "#no"} ] }
```

aaaldapparams

Configuration for LDAP parameter resource.

Properties

(click to see Operations)

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
serverip	<String>	Read-write	IP address of your LDAP server.
serverport	<Integer>	Read-write	Port number on which the LDAP server listens for connections. Default value: 389 Minimum value = 1
authtimeout	<Double>	Read-write	Maximum number of seconds that the Citrix ADC waits for a response from the LDAP server. Default value: 3 Minimum value = 1
ldapbase	<String>	Read-write	Base (the server and location) from which LDAP search commands should start. If the LDAP server is running locally, the default value of base is dc=netscaler, dc=com.

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
ldapbinddn	<String>	Read-write	Complete distinguished name (DN) string used for binding to the LDAP server.
ldapbinddnpassword	<String>	Read-write	Password for binding to the LDAP server. Minimum length = 1
ldaploginname	<String>	Read-write	Name attribute that the Citrix ADC uses to query the external LDAP server or an Active Directory.
searchfilter	<String>	Read-write	String to be combined with the default LDAP user search string to form the value to use when executing an LDAP search. For example, the following values: vpnallowed=true, ldaploginame=""sa maccount"" when combined with the user-supplied username ""bob"", yield the following LDAP search string:

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
			""(; (vpnallowed=true) (samaccount=bob)"" . Minimum length = 1
groupattrname	<String>	Read-write	Attribute name used for group extraction from the LDAP server.
subattributename	<String>	Read-write	Subattribute name used for group extraction from the LDAP server.
sectype	<String>	Read-write	Type of security used for communications between the Citrix ADC and the LDAP server. For the PLAINTEXT setting, no encryption is required. Default value: PLAINTEXT Possible values = PLAINTEXT, TLS, SSL
svrtype	<String>	Read-write	The type of LDAP server. Default value: AAA_LDAP_SERVER_TYPE_DEFAULT

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
			Possible values = AD, NDS
ssonameattribute	<String>	Read-write	Attribute used by the Citrix ADC to query an external LDAP server or Active Directory for an alternative username. This alternative username is then used for single sign-on (SSO).
passwdchange	<String>	Read-write	Accept password change requests. Default value: DISABLED Possible values = ENABLED, DISABLED
nestedgroupextraction	<String>	Read-write	Queries the external LDAP server to determine whether the specified group belongs to another group. Default value: OFF Possible values = ON, OFF
maxnestinglevel	<Double>	Read-write	Number of levels up to which the system can query nested LDAP groups.

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
			Default value: 2 Minimum value = 2
groupnameidentifier	<String>	Read-write	LDAP-group attribute that uniquely identifies the group. No two groups on one LDAP server can have the same group name identifier.
groupsearchattribute	<String>	Read-write	LDAP-group attribute that designates the parent group of the specified group. Use this attribute to search for a group's parent group.
groupsearchsubattribute	<String>	Read-write	LDAP-group subattribute that designates the parent group of the specified group. Use this attribute to search for a group's parent group.
groupsearchfilter	<String>	Read-write	Search-expression that can be specified for sending group-search requests to the LDAP server.
defaultauthenticationgroup	<String>	Read-write	This is the default group that is chosen when the authentication

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
			succeeds in addition to extracted groups. Maximum length = 64
groupauthname	<String>	Read-only	To associate AAA users with an AAA group, use the command "bind AAA group ... -username ...". You can bind different policies to each AAA group. Use the command "bind AAA group ... -policy ...".

Operations

(click to see Properties)

UPDATE| UNSET| GET (ALL)

Some options that you can use for each operations:

- **Getting warnings in response:**NITRO allows you to get warnings in an operation by specifying the "warning" query parameter as "yes". For example, to get warnings while connecting to the Citrix ADC appliance, the URL is as follows:

  http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/login?warning=yes

  If any, the warnings are displayed in the response payload with the HTTP code "209 X-NITRO-WARNING".

- **Authenticated access for individual NITRO operations:**NITRO allows you to logon to the Citrix ADC appliance to perform individual operations. You can use this option instead of

creating a NITRO session (using the login object) and then using that session to perform all operations,

To do this, you must specify the username and password in the request header of the NITRO request as follows:

X-NITRO-USER:<username>

X-NITRO-PASS:<password>

Note: In such cases, make sure that the request header DOES not include the following:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue>

`*Note:`*

Mandatory parameters are marked in redand placeholder content is marked in <green>.

update

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/aaaldapparams

HTTP Method:PUT

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Content-Type:application/json

Request Payload:

```
{"aaaldapparams":{

"serverip":<String_value>,

"serverport":<Integer_value>,

"authtimeout":<Double_value>,

"ldapbase":<String_value>,

"ldapbinddn":<String_value>,

"ldapbinddnpassword":<String_value>,
```

```
"ldaploginname":<String_value>,

"searchfilter":<String_value>,

"groupattrname":<String_value>,

"subattributename":<String_value>,

"sectype":<String_value>,

"svrtype":<String_value>,

"ssonameattribute":<String_value>,

"passwdchange":<String_value>,

"nestedgroupextraction":<String_value>,

"maxnestinglevel":<Double_value>,

"groupnameidentifier":<String_value>,

"groupsearchattribute":<String_value>,

"groupsearchsubattribute":<String_value>,

"groupsearchfilter":<String_value>,

"defaultauthenticationgroup":<String_value>

}}
```

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error

unset

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/aaaldapparams?**action=unset**

HTTP Method:POST

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Content-Type:application/json

Request Payload:

```
{"aaaldapparams":{

"serverip":true,

"serverport":true,

"authtimeout":true,

"ldapbase":true,

"ldapbinddn":true,

"ldapbinddnpassword":true,

"ldaploginname":true,

"searchfilter":true,

"groupattrname":true,

"subattributename":true,

"sectype":true,

"svrtype":true,

"ssonameattribute":true,

"passwdchange":true,

"nestedgroupextraction":true,

"maxnestinglevel":true,
```

```
"groupnameidentifier":true,

"groupsearchattribute":true,

"groupsearchsubattribute":true,

"groupsearchfilter":true,

"defaultauthenticationgroup":true

}}
```

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error

get (all)

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/aaaldapparams

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "aaaldapparams": [ {
```

```
"serverip":<String_value>,

"serverport":<Integer_value>,

"authtimeout":<Double_value>,

"ldapbinddn":<String_value>,

"ldaploginname":<String_value>,

"ldapbase":<String_value>,

"sectype":<String_value>,

"svrtype":<String_value>,

"ssonameattribute":<String_value>,

"searchfilter":<String_value>,

"groupattrname":<String_value>,

"subattributename":<String_value>,

"groupauthname":<String_value>,

"passwdchange":<String_value>,

"nestedgroupextraction":<String_value>,

"maxnestinglevel":<Double_value>,

"groupnameidentifier":<String_value>,

"groupsearchattribute":<String_value>,

"groupsearchsubattribute":<String_value>,

"groupsearchfilter":<String_value>,

"defaultauthenticationgroup":<String_value>

}]}
```

aaaparameter

Configuration for AAA parameter resource.

Properties

(click to see Operations)

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
enablestaticpagecaching	<String>	Read-write	The default state of VPN Static Page caching. If nothing is specified, the default value is set to YES. Default value: YES Possible values = YES, NO
enableenhancedauthfeedback	<String>	Read-write	Enhanced auth feedback provides more information to the end user about the reason for an authentication failure. The default value is set to NO. Default value: NO Possible values = YES, NO
defaultauthtype	<String>	Read-write	The default authentication server type. Default value: LOCAL

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
			Possible values = LOCAL, LDAP, RADIUS, TACACS, CERT
maxaaausers	<Double>	Read-write	Maximum number of concurrent users allowed to log on to VPN simultaneously. Minimum value = 1
maxloginattempts	<Double>	Read-write	Maximum Number of login Attempts. Minimum value = 1
failedlogintimeout	<Double>	Read-write	Number of minutes an account will be locked if user exceeds maximum permissible attempts. Minimum value = 1
aaadnatip	<String>	Read-write	Source IP address to use for traffic that is sent to the authentication server.
enablesessionstickiness	<String>	Read-write	Enables/Disables stickiness to authentication servers. Default value: NO Possible values = YES, NO
aaasessionloglevel	<String>	Read-write	Audit log level, which specifies the types of events to log for cli

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
			executed commands. Available values function as follows: * EMERGENCY - Events that indicate an immediate crisis on the server. * ALERT - Events that might require action. * CRITICAL - Events that indicate an imminent server crisis. * ERROR - Events that indicate some type of error. * WARNING - Events that require action in the near future. * NOTICE - Events that the administrator should know about. * INFORMATIONAL - All but low-level events.

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
			* DEBUG - All events, in extreme detail. Default value: DEFAULT_LOGLEVEL_AAA Possible values = EMERGENCY, ALERT, CRITICAL, ERROR, WARNING, NOTICE, INFORMATIONAL, DEBUG
aaadloglevel	<String>	Read-write	AAAD log level, which specifies the types of AAAD events to log in nsvpn.log. Available values function as follows: * EMERGENCY - Events that indicate an immediate crisis on the server. * ALERT - Events that might require action. * CRITICAL - Events that indicate an imminent server crisis.

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
			* ERROR - Events that indicate some type of error. * WARNING - Events that require action in the near future. * NOTICE - Events that the administrator should know about. * INFORMATIONAL - All but low-level events. * DEBUG - All events, in extreme detail. Default value: INFORMATIONAL Possible values = EMERGENCY, ALERT, CRITICAL, ERROR, WARNING, NOTICE, INFORMATIONAL, DEBUG
dynaddr	<String>	Read-write	Set by the DHCP client when the IP address was fetched dynamically.

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
			Default value: OFF Possible values = ON, OFF
ftmode	<String>	Read-write	First time user mode determines which configuration options are shown by default when logging in to the GUI. This setting is controlled by the GUI. Default value: ON Possible values = ON, HA, OFF
maxsamldeflatesize	<Double>	Read-write	This will set the maximum deflate size in case of SAML Redirect binding.
persistentloginattempts	<String>	Read-write	Persistent storage of unsuccessful user login attempts. Default value: DISABLED Possible values = ENABLED, DISABLED
pwdexpirynotificationdays	<Double>	Read-write	This will set the threshold time in days for password expiry notification. Default value is 0, which means no notification is sent.

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
maxkbquestions	<Double>	Read-write	This will set maximum number of Questions to be asked for KB Validation. Default value is 2, Max Value is 6. Minimum value = 2 Maximum value = 6

Operations

(click to see Properties)

UPDATE| UNSET| GET (ALL)

Some options that you can use for each operations:

- **Getting warnings in response:**NITRO allows you to get warnings in an operation by specifying the "warning" query parameter as "yes". For example, to get warnings while connecting to the Citrix ADC appliance, the URL is as follows:

  http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/login?warning=yes

  If any, the warnings are displayed in the response payload with the HTTP code "209 X-NITRO-WARNING".

- **Authenticated access for individual NITRO operations:**NITRO allows you to logon to the Citrix ADC appliance to perform individual operations. You can use this option instead of creating a NITRO session (using the login object) and then using that session to perform all operations,

  To do this, you must specify the username and password in the request header of the NITRO request as follows:

  X-NITRO-USER:<username>

  X-NITRO-PASS:<password>

  Note: In such cases, make sure that the request header DOES not include the following:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue>

`*Note:` *

Mandatory parameters are marked in redand placeholder content is marked in <green>.

update

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/aaaparameter

HTTP Method:PUT

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Content-Type:application/json

Request Payload:

```
{"aaaparameter":{

"enablestaticpagecaching":<String_value>,

"enableenhancedauthfeedback":<String_value>,

"defaultauthtype":<String_value>,

"maxaaausers":<Double_value>,

"maxloginattempts":<Double_value>,

"failedlogintimeout":<Double_value>,

"aaadnatip":<String_value>,

"enablesessionstickiness":<String_value>,

"aaasessionloglevel":<String_value>,

"aaadloglevel":<String_value>,

"dynaddr":<String_value>,

"ftmode":<String_value>,
```

```
"maxsamldeflatesize":<Double_value>,

"persistentloginattempts":<String_value>,

"pwdexpirynotificationdays":<Double_value>,

"maxkbquestions":<Double_value>

}}
```

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error

unset

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/aaaparameter?**action=unset**

HTTP Method:POST

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Content-Type:application/json

Request Payload:

```
{"aaaparameter":{

"enablestaticpagecaching":true,

"enableenhancedauthfeedback":true,

"defaultauthtype":true,

"maxaaausers":true,

"aaadnatip":true,
```

```
"maxloginattempts":true,

"enablesessionstickiness":true,

"maxsamldeflatesize":true,

"persistentloginattempts":true,

"pwdexpirynotificationdays":true,

"maxkbquestions":true,

"aaasessionloglevel":true,

"aaadloglevel":true,

"dynaddr":true,

"ftmode":true

}}
```

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error

get (all)

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/aaaparameter

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "aaaparameter": [ {

"enablestaticpagecaching":<String_value>,

"enableenhancedauthfeedback":<String_value>,

"defaultauthtype":<String_value>,

"maxaaausers":<Double_value>,

"aaadnatip":<String_value>,

"maxloginattempts":<Double_value>,

"failedlogintimeout":<Double_value>,

"enablesessionstickiness":<String_value>,

"aaasessionloglevel":<String_value>,

"aaadloglevel":<String_value>,

"dynaddr":<String_value>,

"ftmode":<String_value>,

"maxsamldeflatesize":<Double_value>,

"persistentloginattempts":<String_value>,

"pwdexpirynotificationdays":<Double_value>,

"maxkbquestions":<Double_value>
```

```
}]}
```

aaapreauthenticationaction

Configuration for pre authentication action resource.

Properties

(click to see Operations)

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
name	<String>	Read-write	Name for the preauthentication action. Must begin with a letter, number, or the underscore character (_), and must consist only of letters, numbers, and the hyphen (-), period (.) pound (#), space ( ), at (@), equals (=), colon (:), and underscore characters. Cannot be changed after preauthentication action is created. The following requirement applies only to the Citrix ADC CLI: If the name includes one or more spaces, enclose the name in double or single quotation marks (for example,

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
			"my aaa action" or 'my aaa action'). Minimum length = 1
preauthenticationaction	<String>	Read-write	Allow or deny logon after endpoint analysis (EPA) results. Possible values = ALLOW, DENY
killprocess	<String>	Read-write	String specifying the name of a process to be terminated by the endpoint analysis (EPA) tool.
deletefiles	<String>	Read-write	String specifying the path(s) and name(s) of the files to be deleted by the endpoint analysis (EPA) tool.
defaultepagroup	<String>	Read-write	This is the default group that is chosen when the EPA check succeeds. Maximum length = 64
__count	<Double>	Read-only	count parameter

Operations

(click to see Properties)

ADD| DELETE| UPDATE| UNSET| GET (ALL)| GET| COUNT

Some options that you can use for each operations:

- **Getting warnings in response:**NITRO allows you to get warnings in an operation by specifying the "warning" query parameter as "yes". For example, to get warnings while connecting to the Citrix ADC appliance, the URL is as follows:

  http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/login?warning=yes

  If any, the warnings are displayed in the response payload with the HTTP code "209 X-NITRO-WARNING".

- **Authenticated access for individual NITRO operations:**NITRO allows you to logon to the Citrix ADC appliance to perform individual operations. You can use this option instead of creating a NITRO session (using the login object) and then using that session to perform all operations,

  To do this, you must specify the username and password in the request header of the NITRO request as follows:

  X-NITRO-USER:<username>

  X-NITRO-PASS:<password>

  Note: In such cases, make sure that the request header DOES not include the following:

  Cookie:NITRO_AUTH_TOKEN=<tokenvalue>

`*Note:`*

Mandatory parameters are marked in redand placeholder content is marked in <green>.

add

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/aaapreauthenticationaction

HTTP Method:POST

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Content-Type:application/json

Request Payload:

```
{"aaapreauthenticationaction":{
```

```
<b>"name":<String_value>,

</b>"preauthenticationaction":<String_value>,

"killprocess":<String_value>,

"deletefiles":<String_value>,

"defaultepagroup":<String_value>

}}
```

Response:

HTTP Status Code on Success: 201 Created HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error

delete

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/aaapreauthenticationaction/name_value<String>

HTTP Method:DELETE

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue>

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error

update

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/aaapreauthenticationaction

HTTP Method:PUT

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Content-Type:application/json

Request Payload:

```
{"aaapreauthenticationaction":{

<b>"name":<String_value>,

</b>"preauthenticationaction":<String_value>,

"killprocess":<String_value>,

"deletefiles":<String_value>,

"defaultepagroup":<String_value>

}}
```

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error

unset

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/aaapreauthenticationaction?**action=unset**

HTTP Method:POST

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Content-Type:application/json

Request Payload:

```
{"aaapreauthenticationaction":{
```

```
<b>"name":<String_value>,

</b>"killprocess":true,

"deletefiles":true,

"defaultepagroup":true

}}
```

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error

get (all)

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/aaapreauthenticationaction

Query-parameters:

attrs

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/aaapreauthenticationaction?**attrs=property-name1,property-name2**

Use this query parameter to specify the resource details that you want to retrieve.

filter

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/aaapreauthenticationaction?**filter=property-name1:property-val1,property-name2:property-val2**

Use this query-parameter to get the filtered set of aaapreauthenticationaction resources configured on Citrix ADC. Filtering can be done on any of the properties of the resource.

view

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/aaapreauthenticationaction?**view=summary**

Note: By default, the retrieved results are displayed in detail view (?view=detail).

pagination

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/aaapreauthenticationaction?**pagesize=#no;pageno=#no**

Use this query-parameter to get the aaapreauthenticationaction resources in chunks.

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "aaapreauthenticationaction": [ {

"name":<String_value>,

"preauthenticationaction":<String_value>,

"killprocess":<String_value>,

"deletefiles":<String_value>,

"defaultepagroup":<String_value>

}]}
```

get

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/aaapreauthenticationaction/name_value<String>

Query-parameters:

attrs

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/aaapreauthenticationaction/name_value<String>?**attrs=property-name1,property-name2**

Use this query parameter to specify the resource details that you want to retrieve.

view

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/aaapreauthenticationaction/name_value<String>?**view=summary**

Note: By default, the retrieved results are displayed in detail view (?view=detail).

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "aaapreauthenticationaction": [ {

"name":<String_value>,

"preauthenticationaction":<String_value>,

"killprocess":<String_value>,

"deletefiles":<String_value>,

"defaultepagroup":<String_value>
```

```
}]}
```

count

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/aaapreauthenticationaction?**count=yes**

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "aaapreauthenticationaction": [ { "__count": "#no"} ] }
```

aaapreauthenticationparameter

Configuration for pre authentication parameter resource.

Properties

(click to see Operations)

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
preauthenticationaction	<String>	Read-write	Deny or allow login on the basis of end point analysis results.

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
			Possible values = ALLOW, DENY
rule	<String>	Read-write	Name of the Citrix ADC named rule, or an expression, to be evaluated by the EPA tool.
killprocess	<String>	Read-write	String specifying the name of a process to be terminated by the EPA tool.
deletefiles	<String>	Read-write	String specifying the path(s) to and name(s) of the files to be deleted by the EPA tool, as a string of between 1 and 1023 characters.

Operations

(click to see Properties)

UPDATE| UNSET| GET (ALL)

Some options that you can use for each operations:

- **Getting warnings in response:**NITRO allows you to get warnings in an operation by specifying the "warning" query parameter as "yes". For example, to get warnings while connecting to the Citrix ADC appliance, the URL is as follows:

  http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/login?warning=yes

  If any, the warnings are displayed in the response payload with the HTTP code "209 X-NITRO-WARNING".
- **Authenticated access for individual NITRO operations:**NITRO allows you to logon to the Citrix ADC appliance to perform individual operations. You can use this option instead of

creating a NITRO session (using the login object) and then using that session to perform all operations,

To do this, you must specify the username and password in the request header of the NITRO request as follows:

X-NITRO-USER:<username>

X-NITRO-PASS:<password>

Note: In such cases, make sure that the request header DOES not include the following:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue>

`*Note:` *

Mandatory parameters are marked in redand placeholder content is marked in <green>.

update

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/aaapreauthenticationparameter

HTTP Method:PUT

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Content-Type:application/json

Request Payload:

```
{"aaapreauthenticationparameter":{

"preauthenticationaction":<String_value>,

"rule":<String_value>,

"killprocess":<String_value>,

"deletefiles":<String_value>

}}
```

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error

unset

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/aaapreauthenticationparameter?**action=unset**

HTTP Method:POST

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Content-Type:application/json

Request Payload:

```
{"aaapreauthenticationparameter":{

"rule":true,

"preauthenticationaction":true,

"killprocess":true,

"deletefiles":true

}}
```

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error

get (all)

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/aaapreauthenticationparameter

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "aaapreauthenticationparameter": [ {

"preauthenticationaction":<String_value>,

"rule":<String_value>,

"killprocess":<String_value>,

"deletefiles":<String_value>

}]}
```

aaapreauthenticationpolicy

Configuration for pre authentication policy resource.

Properties

(click to see Operations)

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
name	<String>	Read-write	Name for the preauthentication policy. Must begin with a letter, number, or the underscore

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
			character (_), and must consist only of letters, numbers, and the hyphen (-), period (.) pound (#), space ( ), at sign (@), equals (=), colon (:), and underscore characters. Cannot be changed after the preauthentication policy is created. The following requirement applies only to the Citrix ADC CLI: If the name includes one or more spaces, enclose the name in double or single quotation marks (for example, "my policy" or 'my policy'). Minimum length = 1
rule	<String>	Read-write	Name of the Citrix ADC named rule, or an expression, defining connections that match the policy.
reqaction	<String>	Read-write	Name of the action that the policy is to invoke when a connection matches the policy.

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
			Minimum length = 1
hits	<Double>	Read-only	No of hits.
__count	<Double>	Read-only	count parameter

Operations

(click to see Properties)

ADD| DELETE| UPDATE| GET (ALL)| GET| COUNT

Some options that you can use for each operations:

- **Getting warnings in response:**NITRO allows you to get warnings in an operation by specifying the "warning" query parameter as "yes". For example, to get warnings while connecting to the Citrix ADC appliance, the URL is as follows:

  http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/login?warning=yes

  If any, the warnings are displayed in the response payload with the HTTP code "209 X-NITRO-WARNING".

- **Authenticated access for individual NITRO operations:**NITRO allows you to logon to the Citrix ADC appliance to perform individual operations. You can use this option instead of creating a NITRO session (using the login object) and then using that session to perform all operations,

  To do this, you must specify the username and password in the request header of the NITRO request as follows:

  X-NITRO-USER:<username>

  X-NITRO-PASS:<password>

  Note: In such cases, make sure that the request header DOES not include the following:

  Cookie:NITRO_AUTH_TOKEN=<tokenvalue>

`*Note:`*

Mandatory parameters are marked in redand placeholder content is marked in <green>.

add

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/aaapreauthenticationpolicy

HTTP Method:POST

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Content-Type:application/json

Request Payload:

```
{"aaapreauthenticationpolicy":{

<b>"name":<String_value>,

</b><b>"rule":<String_value>,

</b>"reqaction":<String_value>

}}
```

Response:

HTTP Status Code on Success: 201 Created HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error

delete

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/aaapreauthenticationpolicy/name_value<String>

HTTP Method:DELETE

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue>

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error

update

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/aaapreauthenticationpolicy

HTTP Method:PUT

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Content-Type:application/json

Request Payload:

```
{"aaapreauthenticationpolicy":{

<b>"name":<String_value>,

</b>"rule":<String_value>,

"reqaction":<String_value>

}}
```

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error

get (all)

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/aaapreauthenticationpolicy

Query-parameters:

attrs

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/aaapreauthenticationpolicy?**attrs=property-name1,property-name2**

Use this query parameter to specify the resource details that you want to retrieve.

filter

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/aaapreauthenticationpolicy?**filter=property-name1:property-val1,property-name2:property-val2**

Use this query-parameter to get the filtered set of aaapreauthenticationpolicy resources configured on Citrix ADC. Filtering can be done on any of the properties of the resource.

view

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/aaapreauthenticationpolicy?**view=summary**

Note: By default, the retrieved results are displayed in detail view (?view=detail).

pagination

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/aaapreauthenticationpolicy?**pagesize=#no;pageno=#no**

Use this query-parameter to get the aaapreauthenticationpolicy resources in chunks.

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "aaapreauthenticationpolicy": [ {

"name":<String_value>,
```

```
"rule":<String_value>,

"reqaction":<String_value>,

"hits":<Double_value>

}]}
```

get

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/aaapreauthenticationpolicy/name_value<String>

Query-parameters:

attrs

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/aaapreauthenticationpolicy/name_value<String>?**attrs=property-name1,property-name2**

Use this query parameter to specify the resource details that you want to retrieve.

view

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/aaapreauthenticationpolicy/name_value<String>?**view=summary**

Note: By default, the retrieved results are displayed in detail view (?view=detail).

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "aaapreauthenticationpolicy": [ {

"name":<String_value>,

"rule":<String_value>,

"reqaction":<String_value>,

"hits":<Double_value>

}]}
```

count

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/aaapreauthenticationpolicy?**count=yes**

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "aaapreauthenticationpolicy": [ { "__count": "#no"} ] }
```

aaapreauthenticationpolicy_aaaglobal_binding

Binding object showing the aaaglobal that can be bound to aaapreauthenticationpolicy.

Properties

(click to see Operations)

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
name	<String>	Read-write	Name of the preauthentication policy whose properties you want to view. Minimum length = 1
boundto	<String>	Read-write	The entity name to which policy is bound.
priority	<Double>	Read-only	.
activepolicy	<Double>	Read-only	.
__count	<Double>	Read-only	count parameter

Operations

(click to see Properties)

GET| GET (ALL)| COUNT

Some options that you can use for each operations:

- **Getting warnings in response:**NITRO allows you to get warnings in an operation by specifying the "warning" query parameter as "yes". For example, to get warnings while connecting to the Citrix ADC appliance, the URL is as follows:

  http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/login?warning=yes

  If any, the warnings are displayed in the response payload with the HTTP code "209 X-NITRO-WARNING".

- **Authenticated access for individual NITRO operations:**NITRO allows you to logon to the Citrix ADC appliance to perform individual operations. You can use this option instead of creating a NITRO session (using the login object) and then using that session to perform all operations,

  To do this, you must specify the username and password in the request header of the NITRO request as follows:

  X-NITRO-USER:<username>

  X-NITRO-PASS:<password>

  Note: In such cases, make sure that the request header DOES not include the following:

  Cookie:NITRO_AUTH_TOKEN=<tokenvalue>

`*Note:`*

Mandatory parameters are marked in redand placeholder content is marked in <green>.

get

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/aaapreauthenticationpolicy_aaaglobal_binding/name_value<String>

Query-parameters:

filter

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/aaapreauthenticationpolicy_aaaglobal_binding/name_value<String>?**filter=property-name1:property-value1,property-name2:property-value2**

Use this query-parameter to get the filtered set of aaapreauthenticationpolicy_aaaglobal_binding resources configured on Citrix ADC. Filtering can be done on any of the properties of the resource.

pagination

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/aaapreauthenticationpolicy_aaaglobal_binding/name_value<String>?**pagesize=#no;pageno=#no**

Use this query-parameter to get the aaapreauthenticationpolicy_aaaglobal_binding resources in chunks.

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "aaapreauthenticationpolicy_aaaglobal_binding": [ {

"name":<String_value>,

"boundto":<String_value>,

"priority":<Double_value>,

"activepolicy":<Double_value>

}]}
```

get (all)

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/aaapreauthenticationpolicy_aaaglobal_binding

Query-parameters:

bulkbindings

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/aaapreauthenticationpolicy_aaaglobal_binding?**bulkbindings=yes**

NITRO allows you to fetch bindings in bulk.

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "aaapreauthenticationpolicy_aaaglobal_binding": [ {

"name":<String_value>,

"boundto":<String_value>,

"priority":<Double_value>,

"activepolicy":<Double_value>

}]}
```

count

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/aaapreauthenticationpolicy_aaaglobal_binding/name_value<String>?**count=yes**

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{"aaapreauthenticationpolicy_aaaglobal_binding": [ { "__count": "#no"} ] }
```

aaapreauthenticationpolicy_binding

Binding object which returns the resources bound to aaapreauthenticationpolicy.

Properties

(click to see Operations)

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
name	<String>	Read-write	Name of the preauthentication policy whose properties you want to view. Minimum length = 1
aaapreauthenticationpolicy_vpnvserver_binding	<aaapreauthenticationpolicy_vpnvserver_binding[]>	Read-only	vpnvserver that can be bound to aaapreauthentication policy.
aaapreauthenticationpolicy_aaaglobal_binding	<aaapreauthenticationpolicy_aaaglobal_binding[]>	Read-only	aaaglobal that can be bound to aaapreauthentication policy.

Operations

(click to see Properties)

GET| GET (ALL)

Some options that you can use for each operations:

- **Getting warnings in response:**NITRO allows you to get warnings in an operation by specifying the "warning" query parameter as "yes". For example, to get warnings while connecting to the Citrix ADC appliance, the URL is as follows:

  http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/login?warning=yes

  If any, the warnings are displayed in the response payload with the HTTP code "209 X-NITRO-WARNING".

- **Authenticated access for individual NITRO operations:**NITRO allows you to logon to the Citrix ADC appliance to perform individual operations. You can use this option instead of creating a NITRO session (using the login object) and then using that session to perform all operations,

  To do this, you must specify the username and password in the request header of the NITRO request as follows:

  X-NITRO-USER:<username>

  X-NITRO-PASS:<password>

  Note: In such cases, make sure that the request header DOES not include the following:

  Cookie:NITRO_AUTH_TOKEN=<tokenvalue>

`*Note:`*

Mandatory parameters are marked in redand placeholder content is marked in <green>.

get

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/aaapreauthenticationpolicy_binding/name_value<String>

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "aaapreauthenticationpolicy_binding": [ {

"name":<String_value>,

"aaapreauthenticationpolicy_vpnvserver_binding":<aaapreauthenticationpolic
y_vpnvserver_binding[]_value>,

"aaapreauthenticationpolicy_aaaglobal_binding":<aaapreauthenticationpolicy
_aaaglobal_binding[]_value>

}]}
```

get (all)

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/aaapreauthenticationpolicy_binding

Query-parameters:

bulkbindings

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/aaapreauthenticationpolicy_binding?**bulkbindings=yes**

NITRO allows you to fetch bindings in bulk.

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "aaapreauthenticationpolicy_binding": [ {

"name":<String_value>,

"aaapreauthenticationpolicy_vpnvserver_binding":<aaapreauthenticationpolic
y_vpnvserver_binding[]_value>,

"aaapreauthenticationpolicy_aaaglobal_binding":<aaapreauthenticationpolicy
_aaaglobal_binding[]_value>

}]}
```

aaapreauthenticationpolicy_vpnvserver_binding

Binding object showing the vpnvserver that can be bound to aaapreauthenticationpolicy.

Properties

(click to see Operations)

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
name	<String>	Read-write	Name of the preauthentication policy whose properties you want to view. Minimum length = 1

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
boundto	<String>	Read-write	The entity name to which policy is bound.
priority	<Double>	Read-only	.
activepolicy	<Double>	Read-only	.
__count	<Double>	Read-only	count parameter

Operations

(click to see Properties)

GET| GET (ALL)| COUNT

Some options that you can use for each operations:

- **Getting warnings in response:**NITRO allows you to get warnings in an operation by specifying the "warning" query parameter as "yes". For example, to get warnings while connecting to the Citrix ADC appliance, the URL is as follows:

  http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/login?warning=yes

  If any, the warnings are displayed in the response payload with the HTTP code "209 X-NITRO-WARNING".

- **Authenticated access for individual NITRO operations:**NITRO allows you to logon to the Citrix ADC appliance to perform individual operations. You can use this option instead of creating a NITRO session (using the login object) and then using that session to perform all operations,

  To do this, you must specify the username and password in the request header of the NITRO request as follows:

  X-NITRO-USER:<username>

  X-NITRO-PASS:<password>

  Note: In such cases, make sure that the request header DOES not include the following:

  Cookie:NITRO_AUTH_TOKEN=<tokenvalue>

`*Note:`*

Mandatory parameters are marked in redand placeholder content is marked in <green>.

get

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/aaapreauthenticationpolicy_vpnvserver_binding/name_value<String>

Query-parameters:

filter

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/aaapreauthenticationpolicy_vpnvserver_binding/name_value<String>?**filter=property-name1:property-value1,property-name2:property-value2**

Use this query-parameter to get the filtered set of aaapreauthenticationpolicy_vpnvserver_binding resources configured on Citrix ADC. Filtering can be done on any of the properties of the resource.

pagination

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/aaapreauthenticationpolicy_vpnvserver_binding/name_value<String>?**pagesize=#no;pageno=#no**

Use this query-parameter to get the aaapreauthenticationpolicy_vpnvserver_binding resources in chunks.

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "aaapreauthenticationpolicy_vpnvserver_binding": [ {

"name":<String_value>,

"boundto":<String_value>,

"priority":<Double_value>,

"activepolicy":<Double_value>

}]}
```

get (all)

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/aaapreauthenticationpolicy_vpnvserver_binding

Query-parameters:

bulkbindings

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/aaapreauthenticationpolicy_vpnvserver_binding?**bulkbindings=yes**

NITRO allows you to fetch bindings in bulk.

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "aaapreauthenticationpolicy_vpnvserver_binding": [ {

"name":<String_value>,

"boundto":<String_value>,

"priority":<Double_value>,

"activepolicy":<Double_value>

}]}
```

count

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/aaapreauthenticationpolicy_vpnvserver_binding/name_value<String>?**count=yes**

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{"aaapreauthenticationpolicy_vpnvserver_binding": [ { "__count":
"#no"} ] }
```

aaaradiusparams

Configuration for RADIUS parameter resource.

Properties

(click to see Operations)

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
serverip	<String>	Read-write	IP address of your RADIUS server. Minimum length = 1
serverport	<Integer>	Read-write	Port number on which the RADIUS server listens for connections. Default value: 1812 Minimum value = 1
authtimeout	<Double>	Read-write	Maximum number of seconds that the Citrix ADC waits for a response from the RADIUS server. Default value: 3 Minimum value = 1
radkey	<String>	Read-write	The key shared between the RADIUS server and clients. Required for allowing the Citrix ADC to communicate with the RADIUS server. Minimum length = 1

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
radnasip	<String>	Read-write	Send the Citrix ADC IP (NSIP) address to the RADIUS server as the Network Access Server IP (NASIP) part of the Radius protocol. Possible values = ENABLED, DISABLED
radnasid	<String>	Read-write	Send the Network Access Server ID (NASID) for your Citrix ADC to the RADIUS server as the nasid part of the Radius protocol.
radvendorid	<Double>	Read-write	Vendor ID for RADIUS group extraction. Minimum value = 1
radattributetype	<Double>	Read-write	Attribute type for RADIUS group extraction. Minimum value = 1
radgroupsprefix	<String>	Read-write	Prefix string that precedes group names within a RADIUS attribute for RADIUS group extraction.
radgroupseparator	<String>	Read-write	Group separator string that delimits group names within a RADIUS attribute

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
			for RADIUS group extraction.
passencoding	<String>	Read-write	Enable password encoding in RADIUS packets that the Citrix ADC sends to the RADIUS server. Default value: pap Possible values = pap, chap, mschapv1, mschapv2
ipvendorid	<Double>	Read-write	Vendor ID attribute in the RADIUS response. If the attribute is not vendor-encoded, it is set to 0.
ipattributetype	<Double>	Read-write	IP attribute type in the RADIUS response. Minimum value = 1
accounting	<String>	Read-write	Configure the RADIUS server state to accept or refuse accounting messages. Possible values = ON, OFF
pwdvendorid	<Double>	Read-write	Vendor ID of the password in the RADIUS response. Used to extract the user password.

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
			Minimum value = 1
pwdattributetype	<Double>	Read-write	Attribute type of the Vendor ID in the RADIUS response. Minimum value = 1
defaultauthentication group	<String>	Read-write	This is the default group that is chosen when the authentication succeeds in addition to extracted groups. Maximum length = 64
callingstationid	<String>	Read-write	Send Calling-Station-ID of the client to the RADIUS server. IP Address of the client is sent as its Calling-Station-ID. Default value: DISABLED Possible values = ENABLED, DISABLED
authservretry	<Double>	Read-write	Number of retry by the Citrix ADC before getting response from the RADIUS server. Default value: 3 Minimum value = 1 Maximum value = 10

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
authentication	<String>	Read-write	Configure the RADIUS server state to accept or refuse authentication messages. Default value: ON Possible values = ON, OFF
tunnelendpointclientip	<String>	Read-write	Send Tunnel Endpoint Client IP address to the RADIUS server. Default value: DISABLED Possible values = ENABLED, DISABLED
groupauthname	<String>	Read-only	To associate AAA users with an AAA group, use the command "bind AAA group ... -username ...". You can bind different policies to each AAA group. Use the command "bind AAA group ... -policy ...".
ipaddress	<String>	Read-only	IP Address.

Operations

(click to see Properties)

UPDATE| UNSET| GET (ALL)

Some options that you can use for each operations:

- **Getting warnings in response:**NITRO allows you to get warnings in an operation by specifying the "warning" query parameter as "yes". For example, to get warnings while connecting to the Citrix ADC appliance, the URL is as follows:

  http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/login?warning=yes

  If any, the warnings are displayed in the response payload with the HTTP code "209 X-NITRO-WARNING".

- **Authenticated access for individual NITRO operations:**NITRO allows you to logon to the Citrix ADC appliance to perform individual operations. You can use this option instead of creating a NITRO session (using the login object) and then using that session to perform all operations,

  To do this, you must specify the username and password in the request header of the NITRO request as follows:

  X-NITRO-USER:<username>

  X-NITRO-PASS:<password>

  Note: In such cases, make sure that the request header DOES not include the following:

  Cookie:NITRO_AUTH_TOKEN=<tokenvalue>

`*Note:`*

Mandatory parameters are marked in redand placeholder content is marked in <green>.

update

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/aaaradiusparams

HTTP Method:PUT

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Content-Type:application/json

Request Payload:

```
{"aaaradiusparams":{

"serverip":<String_value>,

"serverport":<Integer_value>,

"authtimeout":<Double_value>,

<b>"radkey":<String_value>,

</b>"radnasip":<String_value>,

"radnasid":<String_value>,

"radvendorid":<Double_value>,

"radattributetype":<Double_value>,

"radgroupsprefix":<String_value>,

"radgroupseparator":<String_value>,

"passencoding":<String_value>,

"ipvendorid":<Double_value>,

"ipattributetype":<Double_value>,

"accounting":<String_value>,

"pwdvendorid":<Double_value>,

"pwdattributetype":<Double_value>,

"defaultauthenticationgroup":<String_value>,

"callingstationid":<String_value>,

"authservretry":<Double_value>,
```

```
"authentication":<String_value>,

"tunnelendpointclientip":<String_value>

}}
```

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error

unset

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/aaaradiusparams?**action=unset**

HTTP Method:POST

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Content-Type:application/json

Request Payload:

```
{"aaaradiusparams":{

"serverip":true,

"serverport":true,

"authtimeout":true,

"radnasip":true,

"radnasid":true,

"radvendorid":true,

"radattributetype":true,
```

```
"radgroupsprefix":true,

"radgroupseparator":true,

"passencoding":true,

"ipvendorid":true,

"ipattributetype":true,

"accounting":true,

"pwdvendorid":true,

"pwdattributetype":true,

"defaultauthenticationgroup":true,

"callingstationid":true,

"authservretry":true,

"authentication":true,

"tunnelendpointclientip":true

}}
```

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error

get (all)

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/aaaradiusparams

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "aaaradiusparams": [ {

"serverip":<String_value>,

"serverport":<Integer_value>,

"radkey":<String_value>,

"groupauthname":<String_value>,

"authtimeout":<Double_value>,

"radnasip":<String_value>,

"radnasid":<String_value>,

"ipaddress":<String_value>,

"radvendorid":<Double_value>,

"radattributetype":<Double_value>,

"radgroupsprefix":<String_value>,

"radgroupseparator":<String_value>,

"passencoding":<String_value>,

"ipvendorid":<Double_value>,
```

```
"ipattributetype":<Double_value>,

"accounting":<String_value>,

"pwdvendorid":<Double_value>,

"pwdattributetype":<Double_value>,

"defaultauthenticationgroup":<String_value>,

"callingstationid":<String_value>,

"authservretry":<Double_value>,

"authentication":<String_value>,

"tunnelendpointclientip":<String_value>

}]}
```

aaasession

Configuration for active connection resource.

Properties

(click to see Operations)

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
username	<String>	Read-write	Name of the AAA user. Minimum length = 1
groupname	<String>	Read-write	Name of the AAA group. Minimum length = 1
iip	<String>	Read-write	IP address or the first address in the intranet IP range.

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
			Minimum length = 1
netmask	<String>	Read-write	Subnet mask for the intranet IP range. Minimum length = 1
sessionkey	<String>	Read-write	Show aaa session associated with given session key. Minimum length = 1 Maximum length = 127
nodeid	<Double>	Read-write	Unique number that identifies the cluster node. Minimum value = 0 Maximum value = 31
all	<Boolean>	Read-write	Terminate all active AAA-TM/VPN sessions.
publicip	<String>	Read-only	Client's public IP address.
publicport	<Integer>	Read-only	Client's public port. Range 1 - 65535 * in CLI is represented as 65535 in NITRO API
ipaddress	<String>	Read-only	Citrix ADC's IP address.
port	<Integer>	Read-only	Citrix ADC's port.

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
			Range 1 - 65535 * in CLI is represented as 65535 in NITRO API
privateip	<String>	Read-only	Client's private/ mapped IP address.
privateport	<Integer>	Read-only	Client's private/ mapped port. Range 1 - 65535 * in CLI is represented as 65535 in NITRO API
destip	<String>	Read-only	Destination IP address.
destport	<Integer>	Read-only	Destination port. Range 1 - 65535 * in CLI is represented as 65535 in NITRO API
intranetip	<String>	Read-only	Specifies the Intranet IP.
intranetip6	<String>	Read-only	Specifies the Intranet IP6.
peid	<Double>	Read-only	Core id of the session owner.
__count	<Double>	Read-only	count parameter

Operations

(click to see Properties)

KILL| GET (ALL)| COUNT

Some options that you can use for each operations:

- **Getting warnings in response:**NITRO allows you to get warnings in an operation by specifying the "warning" query parameter as "yes". For example, to get warnings while connecting to the Citrix ADC appliance, the URL is as follows:

  http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/login?warning=yes

  If any, the warnings are displayed in the response payload with the HTTP code "209 X-NITRO-WARNING".

- **Authenticated access for individual NITRO operations:**NITRO allows you to logon to the Citrix ADC appliance to perform individual operations. You can use this option instead of creating a NITRO session (using the login object) and then using that session to perform all operations,

  To do this, you must specify the username and password in the request header of the NITRO request as follows:

  X-NITRO-USER:<username>

  X-NITRO-PASS:<password>

  Note: In such cases, make sure that the request header DOES not include the following:

  Cookie:NITRO_AUTH_TOKEN=<tokenvalue>

`*Note:`*

Mandatory parameters are marked in redand placeholder content is marked in <green>.

kill

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/aaasession?**action=kill**

HTTP Method:POST

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Content-Type:application/json

Request Payload:

```
{"aaasession":{

"username":<String_value>,

"groupname":<String_value>,

"iip":<String_value>,

"netmask":<String_value>,

"sessionkey":<String_value>,

"all":<Boolean_value>

}}
```

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error

get (all)

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/aaasession

Query-parameters:

args

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/aaasession?**args=username:<String_value>,groupname:<String_value>,iip:<String_value>,netmask:<String_value>**

Use this query-parameter to get aaasession resources based on additional properties.

attrs

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/aaasession?**attrs=property-name1,property-name2**

Use this query parameter to specify the resource details that you want to retrieve.

filter

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/aaasession?**filter=property-name1:property-val1,property-name2:property-val2**

Use this query-parameter to get the filtered set of aaasession resources configured on Citrix ADC. Filtering can be done on any of the properties of the resource.

view

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/aaasession?**view=summary**

Note: By default, the retrieved results are displayed in detail view (?view=detail).

pagination

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/aaasession?**pagesize=#no;pageno=#no**

Use this query-parameter to get the aaasession resources in chunks.

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "aaasession": [ {

username:<String_value>,groupname:<String_value>,iip:<String_value>,netmas
k:<String_value>,sessionkey:<String_value>,nodeid:<Double_value>"publicip"
:<String_value>,
```

```
"publicport":<Integer_value>,

"ipaddress":<String_value>,

"port":<Integer_value>,

"privateip":<String_value>,

"privateport":<Integer_value>,

"destip":<String_value>,

"destport":<Integer_value>,

"intranetip":<String_value>,

"intranetip6":<String_value>,

"peid":<Double_value>

}]}
```

count

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/aaasession?**count=yes**

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "aaasession": [ { "__count": "#no"} ] }
```

aaatacacsparams

Configuration for tacacs parameters resource.

Properties

(click to see Operations)

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
serverip	<String>	Read-write	IP address of your TACACS+ server. Minimum length = 1
serverport	<Integer>	Read-write	Port number on which the TACACS+ server listens for connections. Default value: 49 Minimum value = 1
authtimeout	<Double>	Read-write	Maximum number of seconds that the Citrix ADC waits for a response from the TACACS+ server. Default value: 3 Minimum value = 1
tacacssecret	<String>	Read-write	Key shared between the TACACS+ server and clients. Required for allowing the Citrix ADC to communicate

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
			with the TACACS+ server. Minimum length = 1
authorization	<String>	Read-write	Use streaming authorization on the TACACS+ server. Possible values = ON, OFF
accounting	<String>	Read-write	Send accounting messages to the TACACS+ server. Possible values = ON, OFF
auditfailedcmds	<String>	Read-write	The option for sending accounting messages to the TACACS+ server. Possible values = ON, OFF
groupattrname	<String>	Read-write	TACACS+ group attribute name.Used for group extraction on the TACACS+ server.
defaultauthentication group	<String>	Read-write	This is the default group that is chosen when the authentication succeeds in addition to extracted groups. Maximum length = 64

Operations

(click to see Properties)

UPDATE| UNSET| GET (ALL)

Some options that you can use for each operations:

- **Getting warnings in response:**NITRO allows you to get warnings in an operation by specifying the "warning" query parameter as "yes". For example, to get warnings while connecting to the Citrix ADC appliance, the URL is as follows:

  http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/login?warning=yes

  If any, the warnings are displayed in the response payload with the HTTP code "209 X-NITRO-WARNING".

- **Authenticated access for individual NITRO operations:**NITRO allows you to logon to the Citrix ADC appliance to perform individual operations. You can use this option instead of creating a NITRO session (using the login object) and then using that session to perform all operations,

  To do this, you must specify the username and password in the request header of the NITRO request as follows:

  X-NITRO-USER:<username>

  X-NITRO-PASS:<password>

  Note: In such cases, make sure that the request header DOES not include the following:

  Cookie:NITRO_AUTH_TOKEN=<tokenvalue>

`*Note:`*

Mandatory parameters are marked in redand placeholder content is marked in <green>.

update

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/aaatacacsparams

HTTP Method:PUT

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Content-Type:application/json

Request Payload:

```
{"aaatacacsparams":{

"serverip":<String_value>,

"serverport":<Integer_value>,

"authtimeout":<Double_value>,

"tacacssecret":<String_value>,

"authorization":<String_value>,

"accounting":<String_value>,

"auditfailedcmds":<String_value>,

"groupattrname":<String_value>,

"defaultauthenticationgroup":<String_value>

}}
```

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error

unset

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/aaatacacsparams?**action=unset**

HTTP Method:POST

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Content-Type:application/json

Request Payload:

```
{"aaatacacsparams":{
"serverip":true,
"serverport":true,
"authtimeout":true,
"tacacssecret":true,
"authorization":true,
"accounting":true,
"auditfailedcmds":true,
"groupattrname":true,
"defaultauthenticationgroup":true
}}
```

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error

get (all)

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/aaatacacsparams

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "aaatacacsparams": [ {

"serverip":<String_value>,

"serverport":<Integer_value>,

"authtimeout":<Double_value>,

"tacacssecret":<String_value>,

"authorization":<String_value>,

"accounting":<String_value>,

"auditfailedcmds":<String_value>,

"groupattrname":<String_value>,

"defaultauthenticationgroup":<String_value>

}]}
```

aaauser

Configuration for AAA user resource.

Properties

(click to see Operations)

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
username	<String>	Read-write	Name for the user. Must begin with a letter, number, or the underscore character (_), and must contain only letters, numbers, and the hyphen (-), period (.) pound (#), space ( ), at (@), equals (=), colon (:), and underscore characters. Cannot be changed after the user is added. The following requirement applies only to the Citrix ADC CLI: If the name includes one or more spaces, enclose the name in double or single quotation marks (for example, "my aaa user" or "my aaa user"). Minimum length = 1
password	<String>	Read-write	Password with which the user logs on. Required for any user account that does not exist on an external

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
			authentication server. If you are not using an external authentication server, all user accounts must have a password. If you are using an external authentication server, you must provide a password for local user accounts that do not exist on the authentication server. Minimum length = 1
loggedin	<Boolean>	Read-write	Show whether the user is logged in or not.
__count	<Double>	Read-only	count parameter

Operations

(click to see Properties)

ADD| DELETE| UPDATE| UNLOCK| GET (ALL)| GET| COUNT

Some options that you can use for each operations:

- **Getting warnings in response:**NITRO allows you to get warnings in an operation by specifying the "warning" query parameter as "yes". For example, to get warnings while connecting to the Citrix ADC appliance, the URL is as follows:

  http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/login?warning=yes

If any, the warnings are displayed in the response payload with the HTTP code "209 X-NITRO-WARNING".

- **Authenticated access for individual NITRO operations:**NITRO allows you to logon to the Citrix ADC appliance to perform individual operations. You can use this option instead of creating a NITRO session (using the login object) and then using that session to perform all operations,

  To do this, you must specify the username and password in the request header of the NITRO request as follows:

  X-NITRO-USER:<username>

  X-NITRO-PASS:<password>

  Note: In such cases, make sure that the request header DOES not include the following:

  Cookie:NITRO_AUTH_TOKEN=<tokenvalue>

`*Note:`*

Mandatory parameters are marked in redand placeholder content is marked in <green>.

add

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/aaauser

HTTP Method:POST

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Content-Type:application/json

Request Payload:

```
{"aaauser":{

<b>"username":<String_value>,

</b>"password":<String_value>

}}
```

Response:

HTTP Status Code on Success: 201 Created HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error

delete

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/aaauser/username_value<String>

HTTP Method:DELETE

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue>

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error

update

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/aaauser

HTTP Method:PUT

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Content-Type:application/json

Request Payload:

```
{"aaauser":{

<b>"username":<String_value>,

</b><b>"password":<String_value>

</b>}}
```

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error

unlock

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/aaauser?**action=unlock**

HTTP Method:POST

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Content-Type:application/json

Request Payload:

```
{"aaauser":{

<b>"username":<String_value>

</b>}}
```

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error

get (all)

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/aaauser

Query-parameters:

args

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/aaauser?**args=username:<String_value>,loggedin:<Boolean_value>**

Use this query-parameter to get aaauser resources based on additional properties.

attrs

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/aaauser?**attrs=property-name1,property-name2**

Use this query parameter to specify the resource details that you want to retrieve.

filter

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/aaauser?**filter=property-name1:property-val1,property-name2:property-val2**

Use this query-parameter to get the filtered set of aaauser resources configured on Citrix ADC. Filtering can be done on any of the properties of the resource.

view

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/aaauser?**view=summary**

Note: By default, the retrieved results are displayed in detail view (?view=detail).

pagination

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/aaauser?**pagesize=#no;pageno=#no**

Use this query-parameter to get the aaauser resources in chunks.

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "aaauser": [ {

username:<String_value>,loggedin:<Boolean_value>"password":<String_value>

}]}
```

get

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/aaauser/username_value<String>

Query-parameters:

attrs

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/aaauser/username_value<String>?**attrs=property-name1,property-name2**

Use this query parameter to specify the resource details that you want to retrieve.

view

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/aaauser/username_value<String>?**view=summary**

Note: By default, the retrieved results are displayed in detail view (?view=detail).

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "aaauser": [ {

username:<String_value>,loggedin:<Boolean_value>"password":<String_value>

}]}
```

count

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/aaauser?**count=yes**

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "aaauser": [ { "__count": "#no"} ] }
```

aaauser_aaagroup_binding

Binding object showing the aaagroup that can be bound to aaauser.

Properties

(click to see Operations)

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
username	<String>	Read-write	User account to which to bind the policy. Minimum length = 1
groupname	<String>	Read-write	The group name.
gotopriorityexpression	<String>	Read-write	Expression or other value specifying the next policy to evaluate if the current policy evaluates to TRUE. Specify one of the following values: * NEXT - Evaluate the policy with the next higher priority number. * END - End policy evaluation. * USE_INVOCATION_RESULT - Applicable if this policy invokes another policy label. If the final goto in the invoked policy label has a value of END, the evaluation stops. If the final goto is anything other than END, the current policy label performs a NEXT. * An expression that evaluates to a number. If you specify an expression, the number to which it evaluates determines the next

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
			policy to evaluate, as follows: * If the expression evaluates to a higher numbered priority, the policy with that priority is evaluated next. * If the expression evaluates to the priority of the current policy, the policy with the next higher numbered priority is evaluated next. * If the expression evaluates to a number that is larger than the largest numbered priority, policy evaluation ends. An UNDEF event is triggered if: * The expression is invalid. * The expression evaluates to a priority number that is numerically lower than the current policy's priority. * The expression evaluates to a priority number that is between the current policy's priority number (say, 30) and the highest priority number (say, 100), but does not match any configured priority number (for

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
			example, the expression evaluates to the number 85). This example assumes that the priority number increments by 10 for every successive policy, and therefore a priority number of 85 does not exist in the policy label.
__count	<Double>	Read-write	count parameter

Operations

(click to see Properties)

GET| GET (ALL)| COUNT

Some options that you can use for each operations:

- **Getting warnings in response:**NITRO allows you to get warnings in an operation by specifying the "warning" query parameter as "yes". For example, to get warnings while connecting to the Citrix ADC appliance, the URL is as follows:

  http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/login?warning=yes

  If any, the warnings are displayed in the response payload with the HTTP code "209 X-NITRO-WARNING".

- **Authenticated access for individual NITRO operations:**NITRO allows you to logon to the Citrix ADC appliance to perform individual operations. You can use this option instead of creating a NITRO session (using the login object) and then using that session to perform all operations,

  To do this, you must specify the username and password in the request header of the NITRO request as follows:

  X-NITRO-USER:<username>

X-NITRO-PASS:<password>

Note: In such cases, make sure that the request header DOES not include the following:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue>

`*Note:`*

Mandatory parameters are marked in redand placeholder content is marked in <green>.

get

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/aaauser_aaagroup_binding/username_value<String>

Query-parameters:

filter

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/aaauser_aaagroup_binding/username_value<String>?**filter=property-name1:property-value1,property-name2:property-value2**

Use this query-parameter to get the filtered set of aaauser_aaagroup_binding resources configured on Citrix ADC. Filtering can be done on any of the properties of the resource.

pagination

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/aaauser_aaagroup_binding/username_value<String>?**pagesize=#no;pageno=#no**

Use this query-parameter to get the aaauser_aaagroup_binding resources in chunks.

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "aaauser_aaagroup_binding": [ {

"username":<String_value>,

"groupname":<String_value>,

"gotopriorityexpression":<String_value>

}]}
```

get (all)

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/aaauser_aaagroup_binding

Query-parameters:

bulkbindings

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/aaauser_aaagroup_binding?**bulkbindings=yes**

NITRO allows you to fetch bindings in bulk.

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "aaauser_aaagroup_binding": [ {

"username":<String_value>,

"groupname":<String_value>,

"gotopriorityexpression":<String_value>

}]}
```

count

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/aaauser_aaagroup_binding/username_value<String>?**count=yes**

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{"aaauser_aaagroup_binding": [ { "__count": "#no"} ] }
```

aaauser_auditnslogpolicy_binding

Binding object showing the auditnslogpolicy that can be bound to aaauser.

Properties

(click to see Operations)

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
priority	<Double>	Read-write	Integer specifying the priority of the policy. A lower number indicates a higher priority. Policies are evaluated in the order of their priority numbers. Maximum value for default syntax policies is 2147483647 and for classic policies max priority is 64000. . Minimum value = 0 Maximum value = 2147483647
username	<String>	Read-write	User account to which to bind the policy. Minimum length = 1
gotopriorityexpression	<String>	Read-write	Expression or other value specifying the next policy to evaluate if the current policy evaluates to TRUE. Specify one of the following values: * NEXT - Evaluate the policy with the next higher priority number. * END - End policy evaluation. *

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
			USE_INVOCATION_RESULT - Applicable if this policy invokes another policy label. If the final goto in the invoked policy label has a value of END, the evaluation stops. If the final goto is anything other than END, the current policy label performs a NEXT. * An expression that evaluates to a number. If you specify an expression, the number to which it evaluates determines the next policy to evaluate, as follows: * If the expression evaluates to a higher numbered priority, the policy with that priority is evaluated next. * If the expression evaluates to the priority of the current policy, the policy with the next higher numbered priority is evaluated next. * If the expression evaluates to a number that is larger than the largest numbered priority, policy

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
			evaluation ends. An UNDEF event is triggered if: * The expression is invalid. * The expression evaluates to a priority number that is numerically lower than the current policy's priority. * The expression evaluates to a priority number that is between the current policy's priority number (say, 30) and the highest priority number (say, 100), but does not match any configured priority number (for example, the expression evaluates to the number 85). This example assumes that the priority number increments by 10 for every successive policy, and therefore a priority number of 85 does not exist in the policy label.
policy	<String>	Read-write	The policy Name.
type	<String>	Read-write	Bindpoint to which the policy is bound. Default value: REQUEST

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
			Possible values = REQUEST, UDP_REQUEST, DNS_REQUEST, ICMP_REQUEST
acttype	<Double>	Read-only	.
__count	<Double>	Read-only	count parameter

Operations

(click to see Properties)

ADD:| DELETE:| GET| GET (ALL)| COUNT

Some options that you can use for each operations:

- **Getting warnings in response:**NITRO allows you to get warnings in an operation by specifying the "warning" query parameter as "yes". For example, to get warnings while connecting to the Citrix ADC appliance, the URL is as follows:

  http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/login?warning=yes

  If any, the warnings are displayed in the response payload with the HTTP code "209 X-NITRO-WARNING".

- **Authenticated access for individual NITRO operations:**NITRO allows you to logon to the Citrix ADC appliance to perform individual operations. You can use this option instead of creating a NITRO session (using the login object) and then using that session to perform all operations,

  To do this, you must specify the username and password in the request header of the NITRO request as follows:

  X-NITRO-USER:<username>

  X-NITRO-PASS:<password>

  Note: In such cases, make sure that the request header DOES not include the following:

  Cookie:NITRO_AUTH_TOKEN=<tokenvalue>

`*Note:` *

Mandatory parameters are marked in redand placeholder content is marked in <green>.

add

URL:http://<Citrix-ADC-IP-address(NSIP)/nitro/v1/config/aaauser_auditnslogpolicy_binding

HTTP Method:PUT

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Content-Type:application/json

Request Payload:

```
{

"aaauser_auditnslogpolicy_binding":{

<b>"username":<String_value>,

</b>"policy":<String_value>,

"priority":<Double_value>,

"type":<String_value>,

"gotopriorityexpression":<String_value>

}}
```

Response:

HTTP Status Code on Success: 201 Created HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error

delete

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/aaauser_auditnslogpolicy_binding/username_value<String>

Query-parameters:

args

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/aaauser_auditnslogpolicy_binding/username_value<String>?**args=policy:<String_value>,type:<String_value>**

HTTP Method:DELETE

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue>

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error

get

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/aaauser_auditnslogpolicy_binding/username_value<String>

Query-parameters:

filter

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/aaauser_auditnslogpolicy_binding/username_value<String>?**filter=property-name1:property-value1,property-name2:property-value2**

Use this query-parameter to get the filtered set of aaauser_auditnslogpolicy_binding resources configured on Citrix ADC. Filtering can be done on any of the properties of the resource.

pagination

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/aaauser_auditnslogpolicy_binding/username_value<String>?**pagesize=#no;pageno=#no**

Use this query-parameter to get the aaauser_auditnslogpolicy_binding resources in chunks.

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "aaauser_auditnslogpolicy_binding": [ {

"priority":<Double_value>,

"username":<String_value>,

"gotopriorityexpression":<String_value>,

"policy":<String_value>,

"type":<String_value>,

"acttype":<Double_value>

}]}
```

get (all)

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/aaauser_auditnslogpolicy_binding

Query-parameters:

bulkbindings

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/aaauser_auditnslogpolicy_binding?**bulkbindings=yes**

NITRO allows you to fetch bindings in bulk.

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "aaauser_auditnslogpolicy_binding": [ {

"priority":<Double_value>,

"username":<String_value>,

"gotopriorityexpression":<String_value>,

"policy":<String_value>,

"type":<String_value>,

"acttype":<Double_value>

}]}
```

count

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/aaauser_auditnslogpolicy_binding/username_value<String>?**count=yes**

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{"aaauser_auditnslogpolicy_binding": [ { "__count": "#no"} ] }
```

aaauser_auditsyslogpolicy_binding

Binding object showing the auditsyslogpolicy that can be bound to aaauser.

Properties

(click to see Operations)

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
priority	<Double>	Read-write	Integer specifying the priority of the policy. A lower number indicates a higher priority. Policies are evaluated in the order of their priority numbers. Maximum value for default syntax policies is 2147483647 and for classic policies max priority is 64000. . Minimum value = 0

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
			Maximum value = 2147483647
username	<String>	Read-write	User account to which to bind the policy. Minimum length = 1
gotopriorityexpression	<String>	Read-write	Expression or other value specifying the next policy to evaluate if the current policy evaluates to TRUE. Specify one of the following values: * NEXT - Evaluate the policy with the next higher priority number. * END - End policy evaluation. * USE_INVOCATION_RESULT - Applicable if this policy invokes another policy label. If the final goto in the invoked policy label has a value of END, the evaluation stops. If the final goto is anything other than END, the current policy label performs a NEXT. * An expression that evaluates to a number. If you specify an expression, the number to which it evaluates

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
			determines the next policy to evaluate, as follows: * If the expression evaluates to a higher numbered priority, the policy with that priority is evaluated next. * If the expression evaluates to the priority of the current policy, the policy with the next higher numbered priority is evaluated next. * If the expression evaluates to a number that is larger than the largest numbered priority, policy evaluation ends. An UNDEF event is triggered if: * The expression is invalid. * The expression evaluates to a priority number that is numerically lower than the current policy's priority. * The expression evaluates to a priority number that is between the current policy's priority number (say, 30) and the highest priority number (say, 100), but does not match any configured priority

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
			number (for example, the expression evaluates to the number 85). This example assumes that the priority number increments by 10 for every successive policy, and therefore a priority number of 85 does not exist in the policy label.
policy	<String>	Read-write	The policy Name.
type	<String>	Read-write	Bindpoint to which the policy is bound. Default value: REQUEST Possible values = REQUEST, UDP_REQUEST, DNS_REQUEST, ICMP_REQUEST
acttype	<Double>	Read-only	.
__count	<Double>	Read-only	count parameter

Operations

(click to see Properties)

ADD:| DELETE:| GET| GET (ALL)| COUNT

Some options that you can use for each operations:

- **Getting warnings in response:**NITRO allows you to get warnings in an operation by specifying the "warning" query parameter as "yes". For example, to get warnings while connecting to the Citrix ADC appliance, the URL is as follows:

  http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/login?warning=yes

  If any, the warnings are displayed in the response payload with the HTTP code "209 X-NITRO-WARNING".

- **Authenticated access for individual NITRO operations:**NITRO allows you to logon to the Citrix ADC appliance to perform individual operations. You can use this option instead of creating a NITRO session (using the login object) and then using that session to perform all operations,

  To do this, you must specify the username and password in the request header of the NITRO request as follows:

  X-NITRO-USER:<username>

  X-NITRO-PASS:<password>

  Note: In such cases, make sure that the request header DOES not include the following:

  Cookie:NITRO_AUTH_TOKEN=<tokenvalue>

`*Note:`*

Mandatory parameters are marked in redand placeholder content is marked in <green>.

add

URL:http://<Citrix-ADC-IP-address(NSIP)/nitro/v1/config/aaauser_auditsyslogpolicy_binding

HTTP Method:PUT

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Content-Type:application/json

Request Payload:

```
{
```

```
"aaauser_auditsyslogpolicy_binding":{

<b>"username":<String_value>,

</b>"policy":<String_value>,

"priority":<Double_value>,

"type":<String_value>,

"gotopriorityexpression":<String_value>

}}
```

Response:

HTTP Status Code on Success: 201 Created HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error

delete

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/aaauser_auditsyslogpolicy_binding/username_value<String>

Query-parameters:

args

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/aaauser_auditsyslogpolicy_binding/username_value<String>?**args=policy:<String_value>,type:<String_value>**

HTTP Method:DELETE

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue>

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error

get

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/aaauser_auditsyslogpolicy_binding/username_value<String>

Query-parameters:

filter

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/aaauser_auditsyslogpolicy_binding/username_value<String>?**filter=property-name1:property-value1,property-name2:property-value2**

Use this query-parameter to get the filtered set of aaauser_auditsyslogpolicy_binding resources configured on Citrix ADC. Filtering can be done on any of the properties of the resource.

pagination

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/aaauser_auditsyslogpolicy_binding/username_value<String>?**pagesize=#no;pageno=#no**

Use this query-parameter to get the aaauser_auditsyslogpolicy_binding resources in chunks.

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "aaauser_auditsyslogpolicy_binding": [ {

"priority":<Double_value>,

"username":<String_value>,

"gotopriorityexpression":<String_value>,

"policy":<String_value>,

"type":<String_value>,

"acttype":<Double_value>

}]}
```

get (all)

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/aaauser_auditsyslogpolicy_binding

Query-parameters:

bulkbindings

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/aaauser_auditsyslogpolicy_binding?**bulkbindings=yes**

NITRO allows you to fetch bindings in bulk.

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "aaauser_auditsyslogpolicy_binding": [ {

"priority":<Double_value>,

"username":<String_value>,

"gotopriorityexpression":<String_value>,

"policy":<String_value>,

"type":<String_value>,

"acttype":<Double_value>

}]}
```

count

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/aaauser_auditsyslogpolicy_binding/username_value<String>?**count=yes**

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{"aaauser_auditsyslogpolicy_binding": [ { "__count": "#no"} ] }
```

aaauser_authorizationpolicy_binding

Binding object showing the authorizationpolicy that can be bound to aaauser.

Properties

(click to see Operations)

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
priority	<Double>	Read-write	Integer specifying the priority of the policy. A lower number indicates a higher priority. Policies are evaluated in the order of their priority numbers. Maximum value for default syntax policies is 2147483647 and for classic policies max priority is 64000. . Minimum value = 0 Maximum value = 2147483647
username	<String>	Read-write	User account to which to bind the policy. Minimum length = 1
gotopriorityexpression	<String>	Read-write	Expression specifying the priority of the next policy

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
			which will get evaluated if the current policy rule evaluates to TRUE.
policy	<String>	Read-write	The policy Name.
type	<String>	Read-write	Bindpoint to which the policy is bound. Default value: REQUEST Possible values = REQUEST, UDP_REQUEST, DNS_REQUEST, ICMP_REQUEST
acttype	<Double>	Read-only	.
__count	<Double>	Read-only	count parameter

Operations

(click to see Properties)

ADD:| DELETE:| GET| GET (ALL)| COUNT

Some options that you can use for each operations:

- **Getting warnings in response:**NITRO allows you to get warnings in an operation by specifying the "warning" query parameter as "yes". For example, to get warnings while connecting to the Citrix ADC appliance, the URL is as follows:

  http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/login?warning=yes

  If any, the warnings are displayed in the response payload with the HTTP code "209 X-NITRO-WARNING".
- **Authenticated access for individual NITRO operations:**NITRO allows you to logon to the Citrix ADC appliance to perform individual operations. You can use this option instead of

creating a NITRO session (using the login object) and then using that session to perform all operations,

To do this, you must specify the username and password in the request header of the NITRO request as follows:

X-NITRO-USER:<username>

X-NITRO-PASS:<password>

Note: In such cases, make sure that the request header DOES not include the following:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue>

`*Note:`*

Mandatory parameters are marked in redand placeholder content is marked in <green>.

add

URL:http://<Citrix-ADC-IP-address(NSIP)/nitro/v1/config/aaauser_authorizationpolicy_binding

HTTP Method:PUT

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Content-Type:application/json

Request Payload:

```
{

"aaauser_authorizationpolicy_binding":{

<b>"username":<String_value>,

</b>"policy":<String_value>,

"priority":<Double_value>,

"type":<String_value>,

"gotopriorityexpression":<String_value>
```

```
}}
```

Response:

HTTP Status Code on Success: 201 Created HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error

delete

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/aaauser_authorizationpolicy_binding/username_value<String>

Query-parameters:

args

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/aaauser_authorizationpolicy_binding/username_value<String>?**args=policy:<String_value>,type:<String_value>**

HTTP Method:DELETE

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue>

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error

get

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/aaauser_authorizationpolicy_binding/username_value<String>

Query-parameters:

filter

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/aaauser_authorizationpolicy_binding/username_value<String>?**filter=property-name1:property-value1,property-name2:property-value2**

Use this query-parameter to get the filtered set of aaauser_authorizationpolicy_binding resources configured on Citrix ADC. Filtering can be done on any of the properties of the resource.

pagination

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/aaauser_authorizationpolicy_binding/username_value<String>?**pagesize=#no;pageno=#no**

Use this query-parameter to get the aaauser_authorizationpolicy_binding resources in chunks.

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "aaauser_authorizationpolicy_binding": [ {

"priority":<Double_value>,

"username":<String_value>,

"gotopriorityexpression":<String_value>,

"policy":<String_value>,

"type":<String_value>,
```

```
"acttype":<Double_value>

}]}
```

get (all)

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/aaauser_authorizationpolicy_binding

Query-parameters:

bulkbindings

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/aaauser_authorizationpolicy_binding?**bulkbindings=yes**

NITRO allows you to fetch bindings in bulk.

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "aaauser_authorizationpolicy_binding": [ {

"priority":<Double_value>,

"username":<String_value>,

"gotopriorityexpression":<String_value>,
```

```
"policy":<String_value>,

"type":<String_value>,

"acttype":<Double_value>

}]}
```

count

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/aaauser_authorizationpolicy_binding/username_value<String>?**count=yes**

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{"aaauser_authorizationpolicy_binding": [ { "__count": "#no"} ] }
```

aaauser_binding

Binding object which returns the resources bound to aaauser.

Properties

(click to see Operations)

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
username	<String>	Read-write	Name of the user who has the account. Minimum length = 1
aaauser_auditnslogpolicy_binding	<aaauser_auditnslogpolicy_binding[]>	Read-only	auditnslogpolicy that can be bound to aaauser.
aaauser_intranetip6_binding	<aaauser_intranetip6_binding[]>	Read-only	intranetip6 that can be bound to aaauser.
aaauser_vpnsessionpolicy_binding	<aaauser_vpnsessionpolicy_binding[]>	Read-only	vpnsessionpolicy that can be bound to aaauser.
aaauser_authorizationpolicy_binding	<aaauser_authorizationpolicy_binding[]>	Read-only	authorizationpolicy that can be bound to aaauser.
aaauser_vpnurlpolicy_binding	<aaauser_vpnurlpolicy_binding[]>	Read-only	vpnurlpolicy that can be bound to aaauser.
aaauser_intranetip_binding	<aaauser_intranetip_binding[]>	Read-only	intranetip that can be bound to aaauser.
aaauser_auditsyslogpolicy_binding	<aaauser_auditsyslogpolicy_binding[]>	Read-only	auditsyslogpolicy that can be bound to aaauser.
aaauser_aaagroup_binding	<aaauser_aaagroup_binding[]>	Read-only	aaagroup that can be bound to aaauser.
aaauser_vpnurl_binding	<aaauser_vpnurl_binding[]>	Read-only	vpnurl that can be bound to aaauser.
aaauser_vpntrafficpolicy_binding	<aaauser_vpntrafficpolicy_binding[]>	Read-only	vpntrafficpolicy that can be bound to aaauser.

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
aaauser_vpnintranet application_binding	<aaauser_vpnintrane tapplication_binding[] >	Read-only	vpnintranetapplicatio n that can be bound to aaauser.
aaauser_tmsessionp olicy_binding	<aaauser_tmsession policy_binding[]>	Read-only	tmsessionpolicy that can be bound to aaauser.

Operations

(click to see Properties)

GET| GET (ALL)

Some options that you can use for each operations:

- **Getting warnings in response:**NITRO allows you to get warnings in an operation by specifying the "warning" query parameter as "yes". For example, to get warnings while connecting to the Citrix ADC appliance, the URL is as follows:

  http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/login?warning=yes

  If any, the warnings are displayed in the response payload with the HTTP code "209 X-NITRO-WARNING".

- **Authenticated access for individual NITRO operations:**NITRO allows you to logon to the Citrix ADC appliance to perform individual operations. You can use this option instead of creating a NITRO session (using the login object) and then using that session to perform all operations,

  To do this, you must specify the username and password in the request header of the NITRO request as follows:

  X-NITRO-USER:<username>

  X-NITRO-PASS:<password>

  Note: In such cases, make sure that the request header DOES not include the following:

  Cookie:NITRO_AUTH_TOKEN=<tokenvalue>

`*Note:`*

Mandatory parameters are marked in redand placeholder content is marked in <green>.

get

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/aaauser_binding/ username_value<String>

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "aaauser_binding": [ {

"username":<String_value>,

"aaauser_authorizationpolicy_binding":<aaauser_authorizationpolicy_binding
[]_value>,

"aaauser_vpnurl_binding":<aaauser_vpnurl_binding[]_value>,

"aaauser_vpntrafficpolicy_binding":<aaauser_vpntrafficpolicy_binding[]_val
ue>,

"aaauser_aaagroup_binding":<aaauser_aaagroup_binding[]_value>,

"aaauser_vpnurlpolicy_binding":<aaauser_vpnurlpolicy_binding[]_value>,

"aaauser_vpnsessionpolicy_binding":<aaauser_vpnsessionpolicy_binding[]_val
ue>,

"aaauser_intranetip_binding":<aaauser_intranetip_binding[]_value>,
```

```
"aaauser_intranetip6_binding":<aaauser_intranetip6_binding[]_value>,

"aaauser_auditsyslogpolicy_binding":<aaauser_auditsyslogpolicy_binding[]_v
alue>,

"aaauser_tmsessionpolicy_binding":<aaauser_tmsessionpolicy_binding[]_value
>,

"aaauser_vpnintranetapplication_binding":<aaauser_vpnintranetapplication_b
inding[]_value>,

"aaauser_auditnslogpolicy_binding":<aaauser_auditnslogpolicy_binding[]_val
ue>

}]}
```

get (all)

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/aaauser_binding

Query-parameters:

bulkbindings

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/aaauser_binding?**bulkbindings=yes**

NITRO allows you to fetch bindings in bulk.

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "aaauser_binding": [ {

"username":<String_value>,

"aaauser_authorizationpolicy_binding":<aaauser_authorizationpolicy_binding
[]_value>,

"aaauser_vpnurl_binding":<aaauser_vpnurl_binding[]_value>,

"aaauser_vpntrafficpolicy_binding":<aaauser_vpntrafficpolicy_binding[]_val
ue>,

"aaauser_aaagroup_binding":<aaauser_aaagroup_binding[]_value>,

"aaauser_vpnurlpolicy_binding":<aaauser_vpnurlpolicy_binding[]_value>,

"aaauser_vpnsessionpolicy_binding":<aaauser_vpnsessionpolicy_binding[]_val
ue>,

"aaauser_intranetip_binding":<aaauser_intranetip_binding[]_value>,

"aaauser_intranetip6_binding":<aaauser_intranetip6_binding[]_value>,

"aaauser_auditsyslogpolicy_binding":<aaauser_auditsyslogpolicy_binding[]_v
alue>,

"aaauser_tmsessionpolicy_binding":<aaauser_tmsessionpolicy_binding[]_value
>,

"aaauser_vpnintranetapplication_binding":<aaauser_vpnintranetapplication_b
inding[]_value>,

"aaauser_auditnslogpolicy_binding":<aaauser_auditnslogpolicy_binding[]_val
ue>

}]}
```

aaauser_intranetip6_binding

Binding object showing the intranetip6 that can be bound to aaauser.

Properties

(click to see Operations)

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
intranetip6	<String>	Read-write	The Intranet IP6 bound to the user.
username	<String>	Read-write	User account to which to bind the policy. Minimum length = 1
gotopriorityexpression	<String>	Read-write	Expression or other value specifying the next policy to evaluate if the current policy evaluates to TRUE. Specify one of the following values: * NEXT - Evaluate the policy with the next higher priority number. * END - End policy evaluation. * USE_INVOCATION_RESULT - Applicable if this policy invokes another policy label. If the final goto in the invoked policy label has a value of END, the evaluation stops. If the final goto is anything other than END, the current policy label performs a NEXT. * An expression that evaluates to a number. If you

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
			specify an expression, the number to which it evaluates determines the next policy to evaluate, as follows: * If the expression evaluates to a higher numbered priority, the policy with that priority is evaluated next. * If the expression evaluates to the priority of the current policy, the policy with the next higher numbered priority is evaluated next. * If the expression evaluates to a number that is larger than the largest numbered priority, policy evaluation ends. An UNDEF event is triggered if: * The expression is invalid. * The expression evaluates to a priority number that is numerically lower than the current policy's priority. * The expression evaluates to a priority number that is between the current policy's priority number (say, 30) and the highest

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
			priority number (say, 100), but does not match any configured priority number (for example, the expression evaluates to the number 85). This example assumes that the priority number increments by 10 for every successive policy, and therefore a priority number of 85 does not exist in the policy label.
numaddr	<Double>	Read-write	Numbers of ipv6 address bound starting with intranetip6.
__count	<Double>	Read-write	count parameter

Operations

(click to see Properties)

ADD:| DELETE:| GET| GET (ALL)| COUNT

Some options that you can use for each operations:

- **Getting warnings in response:**NITRO allows you to get warnings in an operation by specifying the "warning" query parameter as "yes". For example, to get warnings while connecting to the Citrix ADC appliance, the URL is as follows:

  http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/login?warning=yes

  If any, the warnings are displayed in the response payload with the HTTP code "209 X-NITRO-WARNING".

- **Authenticated access for individual NITRO operations:**NITRO allows you to logon to the Citrix ADC appliance to perform individual operations. You can use this option instead of creating a NITRO session (using the login object) and then using that session to perform all operations,

  To do this, you must specify the username and password in the request header of the NITRO request as follows:

  X-NITRO-USER:<username>

  X-NITRO-PASS:<password>

  Note: In such cases, make sure that the request header DOES not include the following:

  Cookie:NITRO_AUTH_TOKEN=<tokenvalue>

`*Note:`*

Mandatory parameters are marked in redand placeholder content is marked in <green>.

add

URL:http://<Citrix-ADC-IP-address(NSIP)/nitro/v1/config/aaauser_intranetip6_binding

HTTP Method:PUT

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Content-Type:application/json

Request Payload:

```
{

"aaauser_intranetip6_binding":{

<b>"username":<String_value>,

</b>"gotopriorityexpression":<String_value>,

"intranetip6":<String_value>,

"numaddr":<Double_value>
```

```
}}
```

Response:

HTTP Status Code on Success: 201 Created HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error

delete

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/aaauser_intranetip6_binding/username_value<String>

Query-parameters:

args

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/aaauser_intranetip6_binding/username_value<String>?**args=intranetip6:<String_value>,numaddr:<Double_value>**

HTTP Method:DELETE

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue>

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error

get

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/aaauser_intranetip6_binding/username_value<String>

Query-parameters:

filter

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/aaauser_intranetip6_binding/ username_value<String>?**filter=property-name1:property-value1,property-name2:property-value2**

Use this query-parameter to get the filtered set of aaauser_intranetip6_binding resources configured on Citrix ADC. Filtering can be done on any of the properties of the resource.

pagination

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/aaauser_intranetip6_binding/ username_value<String>?**pagesize=#no;pageno=#no**

Use this query-parameter to get the aaauser_intranetip6_binding resources in chunks.

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "aaauser_intranetip6_binding": [ {

"intranetip6":<String_value>,

"username":<String_value>,

"gotopriorityexpression":<String_value>,

"numaddr":<Double_value>

}]}
```

get (all)

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/aaauser_intranetip6_binding

Query-parameters:

bulkbindings

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/aaauser_intranetip6_binding?**bulkbindings=yes**

NITRO allows you to fetch bindings in bulk.

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "aaauser_intranetip6_binding": [ {

"intranetip6":<String_value>,

"username":<String_value>,

"gotopriorityexpression":<String_value>,

"numaddr":<Double_value>

}]}
```

count

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/aaauser_intranetip6_binding/username_value<String>?**count=yes**

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{"aaauser_intranetip6_binding": [ { "__count": "#no"} ] }
```

aaauser_intranetip_binding

Binding object showing the intranetip that can be bound to aaauser.

Properties

(click to see Operations)

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
intranetip	<String>	Read-write	The Intranet IP bound to the user.
username	<String>	Read-write	User account to which to bind the policy.

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
			Minimum length = 1
gotopriorityexpression	<String>	Read-write	Expression or other value specifying the next policy to evaluate if the current policy evaluates to TRUE. Specify one of the following values: * NEXT - Evaluate the policy with the next higher priority number. * END - End policy evaluation. * USE_INVOCATION_RESULT - Applicable if this policy invokes another policy label. If the final goto in the invoked policy label has a value of END, the evaluation stops. If the final goto is anything other than END, the current policy label performs a NEXT. * An expression that evaluates to a number. If you specify an expression, the number to which it evaluates determines the next policy to evaluate, as follows: * If the expression evaluates to a higher numbered priority,

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
			the policy with that priority is evaluated next. * If the expression evaluates to the priority of the current policy, the policy with the next higher numbered priority is evaluated next. * If the expression evaluates to a number that is larger than the largest numbered priority, policy evaluation ends. An UNDEF event is triggered if: * The expression is invalid. * The expression evaluates to a priority number that is numerically lower than the current policy's priority. * The expression evaluates to a priority number that is between the current policy's priority number (say, 30) and the highest priority number (say, 100), but does not match any configured priority number (for example, the expression evaluates to the number 85). This example assumes that the

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
			priority number increments by 10 for every successive policy, and therefore a priority number of 85 does not exist in the policy label.
netmask	<String>	Read-write	The netmask for the Intranet IP.
__count	<Double>	Read-write	count parameter

Operations

(click to see Properties)

ADD:| DELETE:| GET| GET (ALL)| COUNT

Some options that you can use for each operations:

- **Getting warnings in response:**NITRO allows you to get warnings in an operation by specifying the "warning" query parameter as "yes". For example, to get warnings while connecting to the Citrix ADC appliance, the URL is as follows:

  http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/login?warning=yes

  If any, the warnings are displayed in the response payload with the HTTP code "209 X-NITRO-WARNING".

- **Authenticated access for individual NITRO operations:**NITRO allows you to logon to the Citrix ADC appliance to perform individual operations. You can use this option instead of creating a NITRO session (using the login object) and then using that session to perform all operations,

  To do this, you must specify the username and password in the request header of the NITRO request as follows:

  X-NITRO-USER:<username>

  X-NITRO-PASS:<password>

  Note: In such cases, make sure that the request header DOES not include the following:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue>

`*Note:` *

Mandatory parameters are marked in redand placeholder content is marked in <green>.

add

URL:http://<Citrix-ADC-IP-address(NSIP)/nitro/v1/config/aaauser_intranetip_binding

HTTP Method:PUT

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Content-Type:application/json

Request Payload:

```
{

"aaauser_intranetip_binding":{

<b>"username":<String_value>,

</b>"intranetip":<String_value>,

"netmask":<String_value>,

"gotopriorityexpression":<String_value>

}}
```

Response:

HTTP Status Code on Success: 201 Created HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error

delete

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/aaauser_intranetip_binding/username_value<String>

Query-parameters:

args

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/aaauser_intranetip_binding/username_value<String>?**args=intranetip:<String_value>,netmask:<String_value>**

HTTP Method:DELETE

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue>

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error

get

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/aaauser_intranetip_binding/username_value<String>

Query-parameters:

filter

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/aaauser_intranetip_binding/username_value<String>?**filter=property-name1:property-value1,property-name2:property-value2**

Use this query-parameter to get the filtered set of aaauser_intranetip_binding resources configured on Citrix ADC. Filtering can be done on any of the properties of the resource.

pagination

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/aaauser_intranetip_binding/username_value<String>?**pagesize=#no;pageno=#no**

Use this query-parameter to get the aaauser_intranetip_binding resources in chunks.

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "aaauser_intranetip_binding": [ {

"intranetip":<String_value>,

"username":<String_value>,

"gotopriorityexpression":<String_value>,

"netmask":<String_value>

}]}
```

get (all)

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/aaauser_intranetip_binding

Query-parameters:

bulkbindings

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/aaauser_intranetip_binding?**bulkbindings=yes**

NITRO allows you to fetch bindings in bulk.

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "aaauser_intranetip_binding": [ {

"intranetip":<String_value>,

"username":<String_value>,

"gotopriorityexpression":<String_value>,

"netmask":<String_value>

}]}
```

count

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/aaauser_intranetip_binding/username_value<String>?**count=yes**

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{"aaauser_intranetip_binding": [ { "__count": "#no"} ] }
```

aaauser_tmsessionpolicy_binding

Binding object showing the tmsessionpolicy that can be bound to aaauser.

Properties

(click to see Operations)

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
priority	<Double>	Read-write	Integer specifying the priority of the policy. A lower number indicates a higher priority. Policies are evaluated in the order of their priority numbers. Maximum value for default syntax policies is 2147483647 and for classic policies max priority is 64000. . Minimum value = 0 Maximum value = 2147483647
username	<String>	Read-write	User account to which to bind the policy. Minimum length = 1

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
gotopriorityexpression	<String>	Read-write	Expression specifying the priority of the next policy which will get evaluated if the current policy rule evaluates to TRUE.
policy	<String>	Read-write	The policy Name.
type	<String>	Read-write	Bindpoint to which the policy is bound. Default value: REQUEST Possible values = REQUEST, UDP_REQUEST, DNS_REQUEST, ICMP_REQUEST
acttype	<Double>	Read-only	.
__count	<Double>	Read-only	count parameter

Operations

(click to see Properties)

ADD:| DELETE:| GET| GET (ALL)| COUNT

Some options that you can use for each operations:

- **Getting warnings in response:**NITRO allows you to get warnings in an operation by specifying the "warning" query parameter as "yes". For example, to get warnings while connecting to the Citrix ADC appliance, the URL is as follows:

  http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/login?warning=yes

  If any, the warnings are displayed in the response payload with the HTTP code "209 X-NITRO-WARNING".

- **Authenticated access for individual NITRO operations:**NITRO allows you to logon to the Citrix ADC appliance to perform individual operations. You can use this option instead of creating a NITRO session (using the login object) and then using that session to perform all operations,

  To do this, you must specify the username and password in the request header of the NITRO request as follows:

  X-NITRO-USER:<username>

  X-NITRO-PASS:<password>

  Note: In such cases, make sure that the request header DOES not include the following:

  Cookie:NITRO_AUTH_TOKEN=<tokenvalue>

`*Note:`*

Mandatory parameters are marked in redand placeholder content is marked in <green>.

add

URL:http://<Citrix-ADC-IP-address(NSIP)/nitro/v1/config/aaauser_tmsessionpolicy_binding

HTTP Method:PUT

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Content-Type:application/json

Request Payload:

```
{

"aaauser_tmsessionpolicy_binding":{

<b>"username":<String_value>,

</b>"policy":<String_value>,

"priority":<Double_value>,

"type":<String_value>,
```

```
"gotopriorityexpression":<String_value>

}}
```

Response:

HTTP Status Code on Success: 201 Created HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error

delete

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/aaauser_tmsessionpolicy_binding/username_value<String>

Query-parameters:

args

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/aaauser_tmsessionpolicy_binding/username_value<String>?**args=policy:<String_value>,type:<String_value>**

HTTP Method:DELETE

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue>

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error

get

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/aaauser_tmsessionpolicy_binding/username_value<String>

Query-parameters:

filter

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/aaauser_tmsessionpolicy_binding/username_value<String>?**filter=property-name1:property-value1,property-name2:property-value2**

Use this query-parameter to get the filtered set of aaauser_tmsessionpolicy_binding resources configured on Citrix ADC. Filtering can be done on any of the properties of the resource.

pagination

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/aaauser_tmsessionpolicy_binding/username_value<String>?**pagesize=#no;pageno=#no**

Use this query-parameter to get the aaauser_tmsessionpolicy_binding resources in chunks.

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "aaauser_tmsessionpolicy_binding": [ {

"priority":<Double_value>,

"username":<String_value>,

"gotopriorityexpression":<String_value>,

"policy":<String_value>,

"type":<String_value>,
```

```
"acttype":<Double_value>

}]}
```

get (all)

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/aaauser_tmsessionpolicy_binding

Query-parameters:

bulkbindings

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/aaauser_tmsessionpolicy_binding?**bulkbindings=yes**

NITRO allows you to fetch bindings in bulk.

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "aaauser_tmsessionpolicy_binding": [ {

"priority":<Double_value>,

"username":<String_value>,

"gotopriorityexpression":<String_value>,
```

```
"policy":<String_value>,

"type":<String_value>,

"acttype":<Double_value>

}]}
```

count

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/aaauser_tmsessionpolicy_binding/username_value<String>?**count=yes**

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{"aaauser_tmsessionpolicy_binding": [ { "__count": "#no"} ] }
```

aaauser_vpnintranetapplication_binding

Binding object showing the vpnintranetapplication that can be bound to aaauser.

Properties

(click to see Operations)

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
username	<String>	Read-write	User account to which to bind the policy. Minimum length = 1
gotopriorityexpression	<String>	Read-write	Expression or other value specifying the next policy to evaluate if the current policy evaluates to TRUE. Specify one of the following values: * NEXT - Evaluate the policy with the next higher priority number. * END - End policy evaluation. * USE_INVOCATION_RESULT - Applicable if this policy invokes another policy label. If the final goto in the invoked policy label has a value of END, the evaluation stops. If the final goto is anything other than END, the current policy label performs a NEXT. * An expression that evaluates to a number. If you specify an expression, the number to which it evaluates determines the next policy to evaluate, as follows: * If the

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
			expression evaluates to a higher numbered priority, the policy with that priority is evaluated next. * If the expression evaluates to the priority of the current policy, the policy with the next higher numbered priority is evaluated next. * If the expression evaluates to a number that is larger than the largest numbered priority, policy evaluation ends. An UNDEF event is triggered if: * The expression is invalid. * The expression evaluates to a priority number that is numerically lower than the current policy's priority. * The expression evaluates to a priority number that is between the current policy's priority number (say, 30) and the highest priority number (say, 100), but does not match any configured priority number (for example, the expression evaluates

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
			to the number 85). This example assumes that the priority number increments by 10 for every successive policy, and therefore a priority number of 85 does not exist in the policy label.
intranetapplication	<String>	Read-write	Name of the intranet VPN application to which the policy applies.
acttype	<Double>	Read-only	.
__count	<Double>	Read-only	count parameter

Operations

(click to see Properties)

ADD:| DELETE:| GET| GET (ALL)| COUNT

Some options that you can use for each operations:

- **Getting warnings in response:**NITRO allows you to get warnings in an operation by specifying the "warning" query parameter as "yes". For example, to get warnings while connecting to the Citrix ADC appliance, the URL is as follows:

  http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/login?warning=yes

  If any, the warnings are displayed in the response payload with the HTTP code "209 X-NITRO-WARNING".

- **Authenticated access for individual NITRO operations:**NITRO allows you to logon to the Citrix ADC appliance to perform individual operations. You can use this option instead of creating a NITRO session (using the login object) and then using that session to perform all operations,

To do this, you must specify the username and password in the request header of the NITRO request as follows:

X-NITRO-USER:<username>

X-NITRO-PASS:<password>

Note: In such cases, make sure that the request header DOES not include the following:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue>

`*Note:`*

Mandatory parameters are marked in redand placeholder content is marked in <green>.

add

URL:http://<Citrix-ADC-IP-address(NSIP)/nitro/v1/config/aaauser_vpnintranetapplication_binding

HTTP Method:PUT

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Content-Type:application/json

Request Payload:

```
{

"aaauser_vpnintranetapplication_binding":{

<b>"username":<String_value>,

</b>"intranetapplication":<String_value>,

"gotopriorityexpression":<String_value>

}}
```

Response:

HTTP Status Code on Success: 201 Created HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error

delete

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/aaauser_vpnintranetapplication_binding/username_value<String>

Query-parameters:

args

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/aaauser_vpnintranetapplication_binding/username_value<String>?**args=intranetapplication:<String_value>**

HTTP Method:DELETE

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue>

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error

get

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/aaauser_vpnintranetapplication_binding/username_value<String>

Query-parameters:

filter

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/aaauser_vpnintranetapplication_binding/username_value<String>?**filter=property-name1:property-value1,property-name2:property-value2**

Use this query-parameter to get the filtered set of aaauser_vpnintranetapplication_binding resources configured on Citrix ADC. Filtering can be done on any of the properties of the resource.

pagination

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/aaauser_vpnintranetapplication_binding/username_value<String>?**pagesize=#no;pageno=#no**

Use this query-parameter to get the aaauser_vpnintranetapplication_binding resources in chunks.

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "aaauser_vpnintranetapplication_binding": [ {

"username":<String_value>,

"gotopriorityexpression":<String_value>,

"intranetapplication":<String_value>,

"acttype":<Double_value>

}]}
```

get (all)

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/aaauser_vpnintranetapplication_binding

Query-parameters:

bulkbindings

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/aaauser_vpnintranetapplication_binding?**bulkbindings=yes**

NITRO allows you to fetch bindings in bulk.

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "aaauser_vpnintranetapplication_binding": [ {

"username":<String_value>,

"gotopriorityexpression":<String_value>,

"intranetapplication":<String_value>,

"acttype":<Double_value>

}]}
```

count

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/aaauser_vpnintranetapplication_binding/username_value<String>?**count=yes**

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{"aaauser_vpnintranetapplication_binding": [ { "__count": "#no"} ] }
```

aaauser_vpnsessionpolicy_binding

Binding object showing the vpnsessionpolicy that can be bound to aaauser.

Properties

(click to see Operations)

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
priority	<Double>	Read-write	Integer specifying the priority of the policy. A lower number indicates a higher priority. Policies are evaluated in the order of their priority numbers. Maximum value for default syntax policies is 2147483647 and for classic policies max priority is 64000. . Minimum value = 0

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
			Maximum value = 2147483647
username	<String>	Read-write	User account to which to bind the policy. Minimum length = 1
gotopriorityexpression	<String>	Read-write	Expression specifying the priority of the next policy which will get evaluated if the current policy rule evaluates to TRUE.
policy	<String>	Read-write	The policy Name.
type	<String>	Read-write	Bindpoint to which the policy is bound. Default value: REQUEST Possible values = REQUEST, UDP_REQUEST, DNS_REQUEST, ICMP_REQUEST
acttype	<Double>	Read-only	.
__count	<Double>	Read-only	count parameter

Operations

(click to see Properties)

ADD:| DELETE:| GET| GET (ALL)| COUNT

Some options that you can use for each operations:

- **Getting warnings in response:**NITRO allows you to get warnings in an operation by specifying the "warning" query parameter as "yes". For example, to get warnings while connecting to the Citrix ADC appliance, the URL is as follows:

  http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/login?warning=yes

  If any, the warnings are displayed in the response payload with the HTTP code "209 X-NITRO-WARNING".

- **Authenticated access for individual NITRO operations:**NITRO allows you to logon to the Citrix ADC appliance to perform individual operations. You can use this option instead of creating a NITRO session (using the login object) and then using that session to perform all operations,

  To do this, you must specify the username and password in the request header of the NITRO request as follows:

  X-NITRO-USER:<username>

  X-NITRO-PASS:<password>

  Note: In such cases, make sure that the request header DOES not include the following:

  Cookie:NITRO_AUTH_TOKEN=<tokenvalue>

`*Note:`*

Mandatory parameters are marked in redand placeholder content is marked in <green>.

add

URL:http://<Citrix-ADC-IP-address(NSIP)/nitro/v1/config/aaauser_vpnsessionpolicy_binding

HTTP Method:PUT

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Content-Type:application/json

Request Payload:

```
{
```

```
"aaauser_vpnsessionpolicy_binding":{

<b>"username":<String_value>,

</b>"policy":<String_value>,

"priority":<Double_value>,

"type":<String_value>,

"gotopriorityexpression":<String_value>

}}
```

Response:

HTTP Status Code on Success: 201 Created HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error

delete

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/aaauser_vpnsessionpolicy_binding/username_value<String>

Query-parameters:

args

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/aaauser_vpnsessionpolicy_binding/username_value<String>?**args=policy:<String_value>,type:<String_value>**

HTTP Method:DELETE

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue>

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error

get

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/aaauser_vpnsessionpolicy_binding/username_value<String>

Query-parameters:

filter

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/aaauser_vpnsessionpolicy_binding/username_value<String>?**filter=property-name1:property-value1,property-name2:property-value2**

Use this query-parameter to get the filtered set of aaauser_vpnsessionpolicy_binding resources configured on Citrix ADC. Filtering can be done on any of the properties of the resource.

pagination

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/aaauser_vpnsessionpolicy_binding/username_value<String>?**pagesize=#no;pageno=#no**

Use this query-parameter to get the aaauser_vpnsessionpolicy_binding resources in chunks.

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "aaauser_vpnsessionpolicy_binding": [ {

"priority":<Double_value>,

"username":<String_value>,

"gotopriorityexpression":<String_value>,

"policy":<String_value>,

"type":<String_value>,

"acttype":<Double_value>

}]}
```

get (all)

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/aaauser_vpnsessionpolicy_binding

Query-parameters:

bulkbindings

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/aaauser_vpnsessionpolicy_binding?**bulkbindings=yes**

NITRO allows you to fetch bindings in bulk.

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "aaauser_vpnsessionpolicy_binding": [ {

"priority":<Double_value>,

"username":<String_value>,

"gotopriorityexpression":<String_value>,

"policy":<String_value>,

"type":<String_value>,

"acttype":<Double_value>

}]}
```

count

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/aaauser_vpnsessionpolicy_binding/username_value<String>?**count=yes**

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{"aaauser_vpnsessionpolicy_binding": [ { "__count": "#no"} ] }
```

aaauser_vpntrafficpolicy_binding

Binding object showing the vpntrafficpolicy that can be bound to aaauser.

Properties

(click to see Operations)

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
priority	<Double>	Read-write	Integer specifying the priority of the policy. A lower number indicates a higher priority. Policies are evaluated in the order of their priority numbers. Maximum value for default syntax policies is 2147483647 and for classic policies max priority is 64000. . Minimum value = 0 Maximum value = 2147483647
username	<String>	Read-write	User account to which to bind the policy. Minimum length = 1
gotopriorityexpression	<String>	Read-write	Expression or other value specifying the next policy to

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
			evaluate if the current policy evaluates to TRUE. Specify one of the following values: * NEXT - Evaluate the policy with the next higher priority number. * END - End policy evaluation. * USE_INVOCATION_RESULT - Applicable if this policy invokes another policy label. If the final goto in the invoked policy label has a value of END, the evaluation stops. If the final goto is anything other than END, the current policy label performs a NEXT. * An expression that evaluates to a number. If you specify an expression, the number to which it evaluates determines the next policy to evaluate, as follows: * If the expression evaluates to a higher numbered priority, the policy with that priority is evaluated next. * If the expression evaluates to the priority of the

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
			current policy, the policy with the next higher numbered priority is evaluated next. * If the expression evaluates to a number that is larger than the largest numbered priority, policy evaluation ends. An UNDEF event is triggered if: * The expression is invalid. * The expression evaluates to a priority number that is numerically lower than the current policy's priority. * The expression evaluates to a priority number that is between the current policy's priority number (say, 30) and the highest priority number (say, 100), but does not match any configured priority number (for example, the expression evaluates to the number 85). This example assumes that the priority number increments by 10 for every successive policy, and therefore a priority number of

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
			85 does not exist in the policy label.
policy	<String>	Read-write	The policy Name.
type	<String>	Read-write	Bindpoint to which the policy is bound. Default value: REQUEST Possible values = REQUEST, UDP_REQUEST, DNS_REQUEST, ICMP_REQUEST
acttype	<Double>	Read-only	.
__count	<Double>	Read-only	count parameter

Operations

(click to see Properties)

ADD:| DELETE:| GET| GET (ALL)| COUNT

Some options that you can use for each operations:

- **Getting warnings in response:**NITRO allows you to get warnings in an operation by specifying the "warning" query parameter as "yes". For example, to get warnings while connecting to the Citrix ADC appliance, the URL is as follows:

  http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/login?warning=yes

  If any, the warnings are displayed in the response payload with the HTTP code "209 X-NITRO-WARNING".

- **Authenticated access for individual NITRO operations:**NITRO allows you to logon to the Citrix ADC appliance to perform individual operations. You can use this option instead of creating a NITRO session (using the login object) and then using that session to perform all operations,

To do this, you must specify the username and password in the request header of the NITRO request as follows:

X-NITRO-USER:<username>

X-NITRO-PASS:<password>

Note: In such cases, make sure that the request header DOES not include the following:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue>

`*Note:`*

Mandatory parameters are marked in redand placeholder content is marked in <green>.

add

URL:http://<Citrix-ADC-IP-address(NSIP)/nitro/v1/config/aaauser_vpntrafficpolicy_binding

HTTP Method:PUT

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Content-Type:application/json

Request Payload:

```
{

"aaauser_vpntrafficpolicy_binding":{

<b>"username":<String_value>,

</b>"policy":<String_value>,

"priority":<Double_value>,

"type":<String_value>,

"gotopriorityexpression":<String_value>

}}
```

Response:

HTTP Status Code on Success: 201 Created HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error

delete

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/aaauser_vpntrafficpolicy_binding/username_value<String>

Query-parameters:

args

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/aaauser_vpntrafficpolicy_binding/username_value<String>?**args=policy:<String_value>,type:<String_value>**

HTTP Method:DELETE

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue>

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error

get

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/aaauser_vpntrafficpolicy_binding/username_value<String>

Query-parameters:

filter

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/aaauser_vpntrafficpolicy_binding/username_value<String>?**filter=property-name1:property-value1,property-name2:property-value2**

Use this query-parameter to get the filtered set of aaauser_vpntrafficpolicy_binding resources configured on Citrix ADC. Filtering can be done on any of the properties of the resource.

pagination

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/aaauser_vpntrafficpolicy_binding/username_value<String>?**pagesize=#no;pageno=#no**

Use this query-parameter to get the aaauser_vpntrafficpolicy_binding resources in chunks.

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "aaauser_vpntrafficpolicy_binding": [ {

"priority":<Double_value>,

"username":<String_value>,

"gotopriorityexpression":<String_value>,

"policy":<String_value>,

"type":<String_value>,

"acttype":<Double_value>

}]}
```

get (all)

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/aaauser_vpntrafficpolicy_binding

Query-parameters:

bulkbindings

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/aaauser_vpntrafficpolicy_binding?**bulkbindings=yes**

NITRO allows you to fetch bindings in bulk.

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "aaauser_vpntrafficpolicy_binding": [ {

"priority":<Double_value>,

"username":<String_value>,

"gotopriorityexpression":<String_value>,

"policy":<String_value>,

"type":<String_value>,
```

```
"acttype":<Double_value>

}]}
```

count

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/aaauser_vpntrafficpolicy_binding/username_value<String>?**count=yes**

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{"aaauser_vpntrafficpolicy_binding": [ { "__count": "#no"} ] }
```

aaauser_vpnurl_binding

Binding object showing the vpnurl that can be bound to aaauser.

Properties

(click to see Operations)

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
urlname	<String>	Read-write	The intranet url.

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
username	<String>	Read-write	User account to which to bind the policy. Minimum length = 1
gotopriorityexpression	<String>	Read-write	Expression or other value specifying the next policy to evaluate if the current policy evaluates to TRUE. Specify one of the following values: * NEXT - Evaluate the policy with the next higher priority number. * END - End policy evaluation. * USE_INVOCATION_RESULT - Applicable if this policy invokes another policy label. If the final goto in the invoked policy label has a value of END, the evaluation stops. If the final goto is anything other than END, the current policy label performs a NEXT. * An expression that evaluates to a number. If you specify an expression, the number to which it evaluates determines the next policy to evaluate, as follows: * If the

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
			expression evaluates to a higher numbered priority, the policy with that priority is evaluated next. * If the expression evaluates to the priority of the current policy, the policy with the next higher numbered priority is evaluated next. * If the expression evaluates to a number that is larger than the largest numbered priority, policy evaluation ends. An UNDEF event is triggered if: * The expression is invalid. * The expression evaluates to a priority number that is numerically lower than the current policy's priority. * The expression evaluates to a priority number that is between the current policy's priority number (say, 30) and the highest priority number (say, 100), but does not match any configured priority number (for example, the expression evaluates

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
			to the number 85). This example assumes that the priority number increments by 10 for every successive policy, and therefore a priority number of 85 does not exist in the policy label.
acttype	<Double>	Read-only	.
__count	<Double>	Read-only	count parameter

Operations

(click to see Properties)

ADD:| DELETE:| GET| GET (ALL)| COUNT

Some options that you can use for each operations:

- **Getting warnings in response:**NITRO allows you to get warnings in an operation by specifying the "warning" query parameter as "yes". For example, to get warnings while connecting to the Citrix ADC appliance, the URL is as follows:

  http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/login?warning=yes

  If any, the warnings are displayed in the response payload with the HTTP code "209 X-NITRO-WARNING".

- **Authenticated access for individual NITRO operations:**NITRO allows you to logon to the Citrix ADC appliance to perform individual operations. You can use this option instead of creating a NITRO session (using the login object) and then using that session to perform all operations,

  To do this, you must specify the username and password in the request header of the NITRO request as follows:

  X-NITRO-USER:<username>

X-NITRO-PASS:<password>

Note: In such cases, make sure that the request header DOES not include the following:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue>

`*Note:`*

Mandatory parameters are marked in redand placeholder content is marked in <green>.

add

URL:http://<Citrix-ADC-IP-address(NSIP)/nitro/v1/config/aaauser_vpnurl_binding

HTTP Method:PUT

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Content-Type:application/json

Request Payload:

```
{

"aaauser_vpnurl_binding":{

<b>"username":<String_value>,

</b>"urlname":<String_value>,

"gotopriorityexpression":<String_value>

}}
```

Response:

HTTP Status Code on Success: 201 Created HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error

delete

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/aaauser_vpnurl_binding/username_value<String>

Query-parameters:

args

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/aaauser_vpnurl_binding/username_value<String>?**args=urlname:<String_value>**

HTTP Method:DELETE

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue>

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error

get

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/aaauser_vpnurl_binding/username_value<String>

Query-parameters:

filter

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/aaauser_vpnurl_binding/username_value<String>?**filter=property-name1:property-value1,property-name2:property-value2**

Use this query-parameter to get the filtered set of aaauser_vpnurl_binding resources configured on Citrix ADC. Filtering can be done on any of the properties of the resource.

pagination

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/aaauser_vpnurl_binding/username_value<String>?**pagesize=#no;pageno=#no**

Use this query-parameter to get the aaauser_vpnurl_binding resources in chunks.

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "aaauser_vpnurl_binding": [ {

"urlname":<String_value>,

"username":<String_value>,

"gotopriorityexpression":<String_value>,

"acttype":<Double_value>

}]}
```

get (all)

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/aaauser_vpnurl_binding

Query-parameters:

bulkbindings

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/aaauser_vpnurl_binding?**bulkbindings=yes**

NITRO allows you to fetch bindings in bulk.

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "aaauser_vpnurl_binding": [ {

"urlname":<String_value>,

"username":<String_value>,

"gotopriorityexpression":<String_value>,

"acttype":<Double_value>

}]}
```

count

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/aaauser_vpnurl_binding/username_value<String>?**count=yes**

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{"aaauser_vpnurl_binding": [ { "__count": "#no"} ] }
```

Analytics

Analytics

Analytics configuration.

NAME	DESCRIPTION
analyticsprofile	Configuration for Analytics profile

analyticsprofile

Configuration for Analytics profile resource.

Properties

(click to see Operations)

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
name	<String>	Read-write	Name for the analytics profile. Must begin with an ASCII alphabetic or underscore (_) character, and must contain only ASCII alphanumeric, underscore, hash

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
			(#), period (.), space, colon (:), at (@), equals (=), and hyphen (-) characters. The following requirement applies only to the Citrix ADC CLI: If the name includes one or more spaces, enclose the name in double or single quotation marks (for example, "my appflow profile" or 'my appflow profile'). Minimum length = 1 Maximum length = 127
collectors	<String>	Read-write	The collector can be an IP, an appflow collector name, a service or a vserver. If IP is specified, the transport is considered as logstream and default port of 5557 is taken. If collector name is specified, the collector properties are taken from the configured

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
			collector. If service is specified, the configured service is assumed as the collector. If vserver is specified, the services bound to it are considered as collectors and the records are load balanced. Minimum length = 1
type	<String>	Read-write	This option indicates what information needs to be collected and exported. Possible values = global, webinsight, tcpinsight, securityinsight, videoinsight, hdxinsight, gatewayinsight, httpstream, lsninsight
httpclientsidemeasurements	<String>	Read-write	On enabling this option, the Citrix ADC will insert a javascript into the HTTP response to collect the client side page-timings and will send the same to the configured collectors. Default value: DISABLED

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
			Possible values = ENABLED, DISABLED
httppagetracking	<String>	Read-write	On enabling this option, the Citrix ADC will link the embedded objects of a page together. Default value: DISABLED Possible values = ENABLED, DISABLED
httpurl	<String>	Read-write	On enabling this option, the Citrix ADC will log the URL in appflow records. Default value: DISABLED Possible values = ENABLED, DISABLED
httphost	<String>	Read-write	On enabling this option, the Citrix ADC will log the Host header in appflow records. Default value: DISABLED Possible values = ENABLED, DISABLED

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
httpmethod	<String>	Read-write	On enabling this option, the Citrix ADC will log the method header in appflow records. Default value: DISABLED Possible values = ENABLED, DISABLED
httpreferer	<String>	Read-write	On enabling this option, the Citrix ADC will log the referer header in appflow records. Default value: DISABLED Possible values = ENABLED, DISABLED
httpuseragent	<String>	Read-write	On enabling this option, the Citrix ADC will log User-Agent header. Default value: DISABLED Possible values = ENABLED, DISABLED
httpcookie	<String>	Read-write	On enabling this option, the Citrix ADC will log cookie header.

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
			Default value: DISABLED Possible values = ENABLED, DISABLED
httplocation	<String>	Read-write	On enabling this option, the Citrix ADC will log location header. Default value: DISABLED Possible values = ENABLED, DISABLED
urlcategory	<String>	Read-write	On enabling this option, the Citrix ADC will send the URL category record. Default value: DISABLED Possible values = ENABLED, DISABLED
httpcontenttype	<String>	Read-write	On enabling this option, the Citrix ADC will log content-length header. Default value: DISABLED

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
			Possible values = ENABLED, DISABLED
httpauthentication	<String>	Read-write	On enabling this option, the Citrix ADC will log Authentication header. Default value: DISABLED Possible values = ENABLED, DISABLED
httpvia	<String>	Read-write	On enabling this option, the Citrix ADC will Via header. Default value: DISABLED Possible values = ENABLED, DISABLED
httpxforwardedforheader	<String>	Read-write	On enabling this option, the Citrix ADC will log X-Forwarded-For header. Default value: DISABLED Possible values = ENABLED, DISABLED

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
httpsetcookie	<String>	Read-write	On enabling this option, the Citrix ADC will log set-cookie header. Default value: DISABLED Possible values = ENABLED, DISABLED
httpsetcookie2	<String>	Read-write	On enabling this option, the Citrix ADC will log set-cookie2 header. Default value: DISABLED Possible values = ENABLED, DISABLED
httpdomainname	<String>	Read-write	On enabling this option, the Citrix ADC will log domain name. Default value: DISABLED Possible values = ENABLED, DISABLED
httpurlquery	<String>	Read-write	On enabling this option, the Citrix ADC will log URL Query.

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
			Default value: DISABLED Possible values = ENABLED, DISABLED
tcpburstreporting	<String>	Read-write	On enabling this option, the Citrix ADC will log TCP burst parameters. Default value: ENABLED Possible values = ENABLED, DISABLED
cqareporting	<String>	Read-write	On enabling this option, the Citrix ADC will log TCP CQA parameters. Default value: DISABLED Possible values = ENABLED, DISABLED
integratedcache	<String>	Read-write	On enabling this option, the Citrix ADC will log the Integrated Caching appflow records. Default value: DISABLED

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
			Possible values = ENABLED, DISABLED
refcnt	<Double>	Read-only	The number of references to the profile.
__count	<Double>	Read-only	count parameter

Operations

(click to see Properties)

ADD| UPDATE| UNSET| DELETE| GET (ALL)| GET| COUNT

Some options that you can use for each operations:

- **Getting warnings in response:**NITRO allows you to get warnings in an operation by specifying the "warning" query parameter as "yes". For example, to get warnings while connecting to the Citrix ADC appliance, the URL is as follows:

  http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/login?warning=yes

  If any, the warnings are displayed in the response payload with the HTTP code "209 X-NITRO-WARNING".

- **Authenticated access for individual NITRO operations:**NITRO allows you to logon to the Citrix ADC appliance to perform individual operations. You can use this option instead of creating a NITRO session (using the login object) and then using that session to perform all operations,

  To do this, you must specify the username and password in the request header of the NITRO request as follows:

  X-NITRO-USER:<username>

  X-NITRO-PASS:<password>

  Note: In such cases, make sure that the request header DOES not include the following:

  Cookie:NITRO_AUTH_TOKEN=<tokenvalue>

`*Note:` *

Mandatory parameters are marked in redand placeholder content is marked in <green>.

add

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/analyticsprofile

HTTP Method:POST

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Content-Type:application/json

Request Payload:

```
{"analyticsprofile":{

<b>"name":<String_value>,

</b>"collectors":<String_value>,

<b>"type":<String_value>,

</b>"httpclientsidemeasurements":<String_value>,

"httppagetracking":<String_value>,

"httpurl":<String_value>,

"httphost":<String_value>,

"httpmethod":<String_value>,

"httpreferer":<String_value>,

"httpuseragent":<String_value>,

"httpcookie":<String_value>,

"httplocation":<String_value>,

"urlcategory":<String_value>,
```

```
"httpcontenttype":<String_value>,

"httpauthentication":<String_value>,

"httpvia":<String_value>,

"httpxforwardedforheader":<String_value>,

"httpsetcookie":<String_value>,

"httpsetcookie2":<String_value>,

"httpdomainname":<String_value>,

"httpurlquery":<String_value>,

"tcpburstreporting":<String_value>,

"cqareporting":<String_value>,

"integratedcache":<String_value>

}}
```

Response:

HTTP Status Code on Success: 201 Created HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error

update

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/analyticsprofile

HTTP Method:PUT

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Content-Type:application/json

Request Payload:

```
{"analyticsprofile":{

<b>"name":<String_value>,

</b>"collectors":<String_value>,

"httpclientsidemeasurements":<String_value>,

"httppagetracking":<String_value>,

"httpurl":<String_value>,

"httphost":<String_value>,

"httpmethod":<String_value>,

"httpreferer":<String_value>,

"httpuseragent":<String_value>,

"httpcookie":<String_value>,

"httplocation":<String_value>,

"urlcategory":<String_value>,

"httpcontenttype":<String_value>,

"httpauthentication":<String_value>,

"httpvia":<String_value>,

"httpxforwardedforheader":<String_value>,

"httpsetcookie":<String_value>,

"httpsetcookie2":<String_value>,

"httpdomainname":<String_value>,

"httpurlquery":<String_value>,
```

```
"tcpburstreporting":<String_value>,

"cqareporting":<String_value>,

"integratedcache":<String_value>

}}
```

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error

unset

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/analyticsprofile?**action=unset**

HTTP Method:POST

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Content-Type:application/json

Request Payload:

```
{"analyticsprofile":{

<b>"name":<String_value>,

</b>"httpclientsidemeasurements":true,

"httppagetracking":true,

"httpurl":true,

"httphost":true,

"httpmethod":true,
```

```
"httpreferer":true,

"httpuseragent":true,

"httpcookie":true,

"httplocation":true,

"urlcategory":true,

"httpcontenttype":true,

"httpauthentication":true,

"httpvia":true,

"httpxforwardedforheader":true,

"httpsetcookie":true,

"httpsetcookie2":true,

"httpdomainname":true,

"httpurlquery":true,

"tcpburstreporting":true,

"cqareporting":true,

"integratedcache":true

}}
```


Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error

delete

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/analyticsprofile/name_value<String>

HTTP Method:DELETE

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue>

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error

get (all)

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/analyticsprofile

Query-parameters:

attrs

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/analyticsprofile?**attrs=property-name1,property-name2**

Use this query parameter to specify the resource details that you want to retrieve.

filter

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/analyticsprofile?**filter=property-name1:property-val1,property-name2:property-val2**

Use this query-parameter to get the filtered set of analyticsprofile resources configured on Citrix ADC. Filtering can be done on any of the properties of the resource.

view

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/analyticsprofile?**view=summary**

Note: By default, the retrieved results are displayed in detail view (?view=detail).

pagination

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/analyticsprofile?**pagesize=#no;pageno=#no**

Use this query-parameter to get the analyticsprofile resources in chunks.

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "analyticsprofile": [ {

"name":<String_value>,

"collectors":<String_value>,

"refcnt":<Double_value>,

"type":<String_value>,

"httpclientsidemeasurements":<String_value>,

"httppagetracking":<String_value>,

"httpurl":<String_value>,

"httphost":<String_value>,

"httpmethod":<String_value>,

"httpreferer":<String_value>,

"httpuseragent":<String_value>,
```

```
"httpcookie":<String_value>,

"httplocation":<String_value>,

"urlcategory":<String_value>,

"httpcontenttype":<String_value>,

"httpauthentication":<String_value>,

"httpvia":<String_value>,

"httpxforwardedforheader":<String_value>,

"httpsetcookie":<String_value>,

"httpsetcookie2":<String_value>,

"httpdomainname":<String_value>,

"httpurlquery":<String_value>,

"tcpburstreporting":<String_value>,

"cqareporting":<String_value>,

"integratedcache":<String_value>

}]}
```

get

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/analyticsprofile/name_value<String>

Query-parameters:

attrs

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/analyticsprofile/name_value<String>?**attrs=property-name1,property-name2**

Use this query parameter to specify the resource details that you want to retrieve.

view

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/analyticsprofile/name_value<String>?**view=summary**

Note: By default, the retrieved results are displayed in detail view (?view=detail).

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "analyticsprofile": [ {

"name":<String_value>,

"collectors":<String_value>,

"refcnt":<Double_value>,

"type":<String_value>,

"httpclientsidemeasurements":<String_value>,

"httppagetracking":<String_value>,

"httpurl":<String_value>,

"httphost":<String_value>,

"httpmethod":<String_value>,
```

```
"httpreferer":<String_value>,

"httpuseragent":<String_value>,

"httpcookie":<String_value>,

"httplocation":<String_value>,

"urlcategory":<String_value>,

"httpcontenttype":<String_value>,

"httpauthentication":<String_value>,

"httpvia":<String_value>,

"httpxforwardedforheader":<String_value>,

"httpsetcookie":<String_value>,

"httpsetcookie2":<String_value>,

"httpdomainname":<String_value>,

"httpurlquery":<String_value>,

"tcpburstreporting":<String_value>,

"cqareporting":<String_value>,

"integratedcache":<String_value>

}]}
```

count

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/analyticsprofile?**count=yes**

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "analyticsprofile": [ { "__count": "#no"} ] }
```

App

App

Application

NAME	DESCRIPTION
application	Configuration for application

application

Configuration for application resource.

Properties

(click to see Operations)

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
apptemplatefilename	<String>	Read-write	Name of the AppExpert application template file.

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
appname	<String>	Read-write	Name to assign to the application on the Citrix ADC. If you do not provide a name, the appliance assigns the application the name of the template file.
deploymentfilename	<String>	Read-write	Name of the deployment file.

Operations

(click to see Properties)

IMPORT| EXPORT| DELETE

Some options that you can use for each operations:

- **Getting warnings in response:**NITRO allows you to get warnings in an operation by specifying the "warning" query parameter as "yes". For example, to get warnings while connecting to the Citrix ADC appliance, the URL is as follows:

  http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/login?warning=yes

  If any, the warnings are displayed in the response payload with the HTTP code "209 X-NITRO-WARNING".

- **Authenticated access for individual NITRO operations:**NITRO allows you to logon to the Citrix ADC appliance to perform individual operations. You can use this option instead of creating a NITRO session (using the login object) and then using that session to perform all operations,

  To do this, you must specify the username and password in the request header of the NITRO request as follows:

  X-NITRO-USER:<username>

  X-NITRO-PASS:<password>

  Note: In such cases, make sure that the request header DOES not include the following:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue>

`*Note:` *

Mandatory parameters are marked in redand placeholder content is marked in <green>.

Import

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/application?**action=Import**

HTTP Method:POST

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Content-Type:application/json

Request Payload:

```
{"application":{

<b>"apptemplatefilename":<String_value>,

</b>"appname":<String_value>,

"deploymentfilename":<String_value>

}}
```

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error

export

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/application?**action=export**

HTTP Method:POST

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Content-Type:application/json

Request Payload:

```
{"application":{

<b>"appname":<String_value>,

</b>"apptemplatefilename":<String_value>,

"deploymentfilename":<String_value>

}}
```

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error

delete

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/application

HTTP Method:DELETE

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue>

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error

Appflow

Appflow

AppFlow configuration.

NAME	DESCRIPTION
appflowaction	Configuration for AppFlow action
appflowaction_analyticsprofile_binding	Binding object showing the analyticsprofile that can be bound to appflowaction
appflowaction_binding	Binding object showing the resources that can be bound to appflowaction
appflowcollector	Configuration for AppFlow collector
appflowglobal_appflowpolicy_binding	Binding object showing the appflowpolicy that can be bound to appflowglobal
appflowglobal_binding	Binding object showing the resources that can be bound to appflowglobal
appflowparam	Configuration for AppFlow parameter
appflowpolicy	Configuration for AppFlow policy
appflowpolicy_appflowglobal_binding	Binding object showing the appflowglobal that can be bound to appflowpolicy
appflowpolicy_appflowpolicylabel_binding	Binding object showing the appflowpolicylabel that can be bound to appflowpolicy
appflowpolicy_binding	Binding object showing the resources that can be bound to appflowpolicy
appflowpolicy_csvserver_binding	Binding object showing the csvserver that can be bound to appflowpolicy
appflowpolicy_lbvserver_binding	Binding object showing the lbvserver that can be bound to appflowpolicy
appflowpolicy_vpnvserver_binding	Binding object showing the vpnvserver that can be bound to appflowpolicy
appflowpolicylabel	Configuration for AppFlow policy label
appflowpolicylabel_appflowpolicy_binding	Binding object showing the appflowpolicy that can be bound to appflowpolicylabel

NAME	DESCRIPTION
appflowpolicylabel_binding	Binding object showing the resources that can be bound to appflowpolicylabel

appflowaction

Configuration for AppFlow action resource.

Properties

(click to see Operations)

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
name	<String>	Read-write	Name for the action. Must begin with an ASCII alphabetic or underscore (_) character, and must contain only ASCII alphanumeric, underscore, hash (#), period (.), space, colon (:), at (@), equals (=), and hyphen (-) characters. The following requirement applies only to the Citrix ADC CLI: If the name includes one or more spaces, enclose the name in double or single quotation marks (for example, "my appflow action"

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
			or 'my appflow action').
collectors	<String[]>	Read-write	Name(s) of collector(s) to be associated with the AppFlow action. Minimum length = 1
clientsidemeasurements	<String>	Read-write	On enabling this option, the Citrix ADC will collect the time required to load and render the mainpage on the client. Default value: DISABLED Possible values = ENABLED, DISABLED
pagetracking	<String>	Read-write	On enabling this option, the Citrix ADC will start tracking the page for waterfall chart by inserting a NS_ESNS cookie in the response. Default value: DISABLED Possible values = ENABLED, DISABLED
webinsight	<String>	Read-write	On enabling this option, the Citrix

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
			ADC will send the webinsight records to the configured collectors. Default value: ENABLED Possible values = ENABLED, DISABLED
securityinsight	<String>	Read-write	On enabling this option, the Citrix ADC will send the security insight records to the configured collectors. Default value: DISABLED Possible values = ENABLED, DISABLED
videoanalytics	<String>	Read-write	On enabling this option, the Citrix ADC will send the videoinsight records to the configured collectors. Default value: DISABLED Possible values = ENABLED, DISABLED
distributionalgorithm	<String>	Read-write	On enabling this option, the Citrix ADC will distribute

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
			records among the collectors. Else, all records will be sent to all the collectors. Default value: DISABLED Possible values = ENABLED, DISABLED
metricslog	<Boolean>	Read-write	If only the stats records are to be exported, turn on this option.
transactionlog	<String>	Read-write	Log ANOMALOUS or ALL transactions. Default value: ALL Possible values = ALL, ANOMALOUS
comment	<String>	Read-write	Any comments about this action. In the CLI, if including spaces between words, enclose the comment in quotation marks. (The quotation marks are not required in the configuration utility.). Maximum length = 256
newname	<String>	Read-write	New name for the AppFlow action. Must begin with an ASCII alphabetic or

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
			underscore (_) character, and must contain only ASCII alphanumeric, underscore, hash (#), period (.), space, colon (:), at (@), equals (=), and hyphen (-) characters. The following requirement applies only to the Citrix ADC CLI: If the name includes one or more spaces, enclose the name in double or single quotation marks (for example, "my appflow action" or 'my appflow action'). Minimum length = 1
hits	<Double>	Read-only	The number of times the action has been taken.
referencecount	<Double>	Read-only	The number of references to the action.
description	<String>	Read-only	Description of the action.
__count	<Double>	Read-only	count parameter

Operations

(click to see Properties)

ADD| DELETE| UPDATE| UNSET| RENAME| GET (ALL)| GET| COUNT

Some options that you can use for each operations:

- **Getting warnings in response:**NITRO allows you to get warnings in an operation by specifying the "warning" query parameter as "yes". For example, to get warnings while connecting to the Citrix ADC appliance, the URL is as follows:

  http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/login?warning=yes

  If any, the warnings are displayed in the response payload with the HTTP code "209 X-NITRO-WARNING".

- **Authenticated access for individual NITRO operations:**NITRO allows you to logon to the Citrix ADC appliance to perform individual operations. You can use this option instead of creating a NITRO session (using the login object) and then using that session to perform all operations,

  To do this, you must specify the username and password in the request header of the NITRO request as follows:

  X-NITRO-USER:<username>

  X-NITRO-PASS:<password>

  Note: In such cases, make sure that the request header DOES not include the following:

  Cookie:NITRO_AUTH_TOKEN=<tokenvalue>

`*Note:`*

Mandatory parameters are marked in redand placeholder content is marked in <green>.

add

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/appflowaction

HTTP Method:POST

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Content-Type:application/json

Request Payload:

```
{"appflowaction":{

<b>"name":<String_value>,

</b><b>"collectors":<String[]_value>,

</b>"clientsidemeasurements":<String_value>,

"pagetracking":<String_value>,

"webinsight":<String_value>,

"securityinsight":<String_value>,

"videoanalytics":<String_value>,

"distributionalgorithm":<String_value>,

"metricslog":<Boolean_value>,

"transactionlog":<String_value>,

"comment":<String_value>

}}
```

Response:

HTTP Status Code on Success: 201 Created HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error

delete

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/appflowaction/name_value<String>

HTTP Method:DELETE

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue>

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error

update

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/appflowaction

HTTP Method:PUT

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Content-Type:application/json

Request Payload:

```
{"appflowaction":{

<b>"name":<String_value>,

</b>"collectors":<String[]_value>,

"clientsidemeasurements":<String_value>,

"comment":<String_value>,

"pagetracking":<String_value>,

"webinsight":<String_value>,

"securityinsight":<String_value>,

"videoanalytics":<String_value>,

"distributionalgorithm":<String_value>

}}
```

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error

unset

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/appflowaction?**action=unset**

HTTP Method:POST

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Content-Type:application/json

Request Payload:

```
{"appflowaction":{

<b>"name":<String_value>,

</b>"clientsidemeasurements":true,

"comment":true,

"pagetracking":true,

"webinsight":true,

"securityinsight":true,

"videoanalytics":true,

"distributionalgorithm":true

}}
```

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error

rename

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/appflowaction?**action=rename**

HTTP Method:POST

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Content-Type:application/json

Request Payload:

```
{"appflowaction":{

<b>"name":<String_value>,

</b><b>"newname":<String_value>

</b>}}
```

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error

get (all)

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/appflowaction

Query-parameters:

attrs

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/appflowaction?**attrs=property-name1,property-name2**

Use this query parameter to specify the resource details that you want to retrieve.

filter

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/appflowaction?**filter=property-name1:property-val1,property-name2:property-val2**

Use this query-parameter to get the filtered set of appflowaction resources configured on Citrix ADC. Filtering can be done on any of the properties of the resource.

view

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/appflowaction?**view=summary**

Note: By default, the retrieved results are displayed in detail view (?view=detail).

pagination

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/appflowaction?**pagesize=#no;pageno=#no**

Use this query-parameter to get the appflowaction resources in chunks.

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "appflowaction": [ {

"name":<String_value>,

"hits":<Double_value>,

"collectors":<String[]_value>,
```

```
"clientsidemeasurements":<String_value>,

"pagetracking":<String_value>,

"webinsight":<String_value>,

"securityinsight":<String_value>,

"videoanalytics":<String_value>,

"distributionalgorithm":<String_value>,

"metricslog":<Boolean_value>,

"transactionlog":<String_value>,

"referencecount":<Double_value>,

"description":<String_value>,

"comment":<String_value>

}]}
```

get

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/appflowaction/name_value<String>

Query-parameters:

attrs

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/appflowaction/name_value<String>?**attrs=property-name1,property-name2**

Use this query parameter to specify the resource details that you want to retrieve.

view

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/appflowaction/name_value<String>?**view=summary**

Note: By default, the retrieved results are displayed in detail view (?view=detail).

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "appflowaction": [ {

"name":<String_value>,

"hits":<Double_value>,

"collectors":<String[]_value>,

"clientsidemeasurements":<String_value>,

"pagetracking":<String_value>,

"webinsight":<String_value>,

"securityinsight":<String_value>,

"videoanalytics":<String_value>,

"distributionalgorithm":<String_value>,

"metricslog":<Boolean_value>,

"transactionlog":<String_value>,

"referencecount":<Double_value>,
```

```
"description":<String_value>,

"comment":<String_value>

}]}
```

count

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/appflowaction?**count=yes**

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "appflowaction": [ { "__count": "#no"} ] }
```

appflowcollector

Configuration for AppFlow collector resource.

Properties

(click to see Operations)

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
name	<String>	Read-write	Name for the collector. Must begin with an ASCII alphabetic or underscore (_) character, and must contain only ASCII alphanumeric, underscore, hash (#), period (.), space, colon (:), at (@), equals (=), and hyphen (-) characters. Only four collectors can be configured. The following requirement applies only to the Citrix ADC CLI: If the name includes one or more spaces, enclose the name in double or single quotation marks (for example, "my appflow collector" or 'my appflow collector'). Minimum length = 1 Maximum length = 127

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
ipaddress	<String>	Read-write	IPv4 address of the collector.
port	<Integer>	Read-write	Port on which the collector listens.
netprofile	<String>	Read-write	Netprofile to associate with the collector. The IP address defined in the profile is used as the source IP address for AppFlow traffic for this collector. If you do not set this parameter, the Citrix ADC IP (NSIP) address is used as the source IP address. Maximum length = 128
transport	<String>	Read-write	Type of collector: either logstream or ipfix or rest. Default value: ipfix, Possible values = ipfix, logstream, rest
newname	<String>	Read-write	New name for the collector. Must begin with an ASCII alphabetic or underscore (_) character, and must

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
			contain only ASCII alphanumeric, underscore, hash (#), period (.), space, colon (:), at(@), equals (=), and hyphen (-) characters. The following requirement applies only to the Citrix ADC CLI: If the name includes one or more spaces, enclose the name in double or single quotation marks (for example, "my appflow coll" or 'my appflow coll'). Minimum length = 1
state	<String>	Read-only	Appflow collector state. Possible values = UP, DOWN, UNKNOWN, BUSY, OUT OF SERVICE, GOING OUT OF SERVICE, DOWN WHEN GOING OUT OF SERVICE, NS_EMPTY_STR,

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
			Unknown, DISABLED
__count	<Double>	Read-only	count parameter

Operations

(click to see Properties)

ADD| UPDATE| UNSET| DELETE| RENAME| GET (ALL)| GET| COUNT

Some options that you can use for each operations:

- **Getting warnings in response:**NITRO allows you to get warnings in an operation by specifying the "warning" query parameter as "yes". For example, to get warnings while connecting to the Citrix ADC appliance, the URL is as follows:

  http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/login?warning=yes

  If any, the warnings are displayed in the response payload with the HTTP code "209 X-NITRO-WARNING".

- **Authenticated access for individual NITRO operations:**NITRO allows you to logon to the Citrix ADC appliance to perform individual operations. You can use this option instead of creating a NITRO session (using the login object) and then using that session to perform all operations,

  To do this, you must specify the username and password in the request header of the NITRO request as follows:

  X-NITRO-USER:<username>

  X-NITRO-PASS:<password>

  Note: In such cases, make sure that the request header DOES not include the following:

  Cookie:NITRO_AUTH_TOKEN=<tokenvalue>

`*Note:`*

Mandatory parameters are marked in redand placeholder content is marked in <green>.

add

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/appflowcollector

HTTP Method:POST

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Content-Type:application/json

Request Payload:

```
{"appflowcollector":{

<b>"name":<String_value>,

</b><b>"ipaddress":<String_value>,

</b>"port":<Integer_value>,

"netprofile":<String_value>,

"transport":<String_value>

}}
```

Response:

HTTP Status Code on Success: 201 Created HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error

update

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/appflowcollector

HTTP Method:PUT

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Content-Type:application/json

Request Payload:

```
{"appflowcollector":{

<b>"name":<String_value>,

</b>"ipaddress":<String_value>,

"port":<Integer_value>,

"netprofile":<String_value>

}}
```

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error

unset

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/appflowcollector?**action=unset**

HTTP Method:POST

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Content-Type:application/json

Request Payload:

```
{"appflowcollector":{

<b>"name":<String_value>,

</b>"ipaddress":true,

"port":true,
```

```
"netprofile":true

}}
```

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error

delete

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/appflowcollector/name_value<String>

HTTP Method:DELETE

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue>

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error

rename

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/appflowcollector?**action=rename**

HTTP Method:POST

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Content-Type:application/json

Request Payload:

```
{"appflowcollector":{

<b>"name":<String_value>,
```

```
</b><b>"newname":<String_value>

</b>}}
```

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error

get (all)

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/appflowcollector

Query-parameters:

attrs

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/appflowcollector?**attrs=property-name1,property-name2**

Use this query parameter to specify the resource details that you want to retrieve.

filter

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/appflowcollector?**filter=property-name1:property-val1,property-name2:property-val2**

Use this query-parameter to get the filtered set of appflowcollector resources configured on Citrix ADC. Filtering can be done on any of the properties of the resource.

view

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/appflowcollector?**view=summary**

Note: By default, the retrieved results are displayed in detail view (?view=detail).

pagination

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/appflowcollector?**pagesize=#no;pageno=#no**

Use this query-parameter to get the appflowcollector resources in chunks.

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "appflowcollector": [ {

"name":<String_value>,

"ipaddress":<String_value>,

"port":<Integer_value>,

"netprofile":<String_value>,

"transport":<String_value>,

"state":<String_value>

}]}
```

get

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/appflowcollector/name_value<String>

Query-parameters:

attrs

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/appflowcollector/name_value<String>?**attrs=property-name1,property-name2**

Use this query parameter to specify the resource details that you want to retrieve.

view

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/appflowcollector/name_value<String>?**view=summary**

Note: By default, the retrieved results are displayed in detail view (?view=detail).

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "appflowcollector": [ {

"name":<String_value>,

"ipaddress":<String_value>,

"port":<Integer_value>,

"netprofile":<String_value>,

"transport":<String_value>,

"state":<String_value>
```

```
}]}
```

count

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/appflowcollector?**count=yes**

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "appflowcollector": [ { "__count": "#no"} ] }
```

appflowglobal_appflowpolicy_binding

Binding object showing the appflowpolicy that can be bound to appflowglobal.

Properties

(click to see Operations)

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
priority	<Double>	Read-write	Specifies the priority of the policy.
globalbindtype	<String>	Read-write	.

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
			Default value: SYSTEM_GLOBAL Possible values = SYSTEM_GLOBAL, VPN_GLOBAL, RNAT_GLOBAL
policyname	<String>	Read-write	Name of the AppFlow policy.
labelname	<String>	Read-write	Name of the label to invoke if the current policy evaluates to TRUE.
gotopriorityexpression	<String>	Read-write	Expression specifying the priority of the next policy which will get evaluated if the current policy rule evaluates to TRUE.
invoke	<Boolean>	Read-write	Invoke policies bound to a virtual server or a user-defined policy label. After the invoked policies are evaluated, the flow returns to the policy with the next priority.
type	<String>	Read-write	Global bind point for which to show detailed information about the policies bound to the bind point.

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
			Possible values = REQ_OVERRIDE, REQ_DEFAULT, OVERRIDE, DEFAULT, OTHERTCP_REQ_OVERRIDE, OTHERTCP_REQ_DEFAULT, MSSQL_REQ_OVERRIDE, MSSQL_REQ_DEFAULT, MYSQL_REQ_OVERRIDE, MYSQL_REQ_DEFAULT, ICA_REQ_OVERRIDE, ICA_REQ_DEFAULT, ORACLE_REQ_OVERRIDE, ORACLE_REQ_DEFAULT
labeltype	<String>	Read-write	Type of policy label to invoke. Specify vserver for a policy label associated with a virtual server, or policylabel for a user-defined policy label. Possible values = vserver, policylabel

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
flowtype	<Double>	Read-only	Flow type of the bound AppFlow policy.
numpol	<Double>	Read-only	The number of policies bound to the bindpoint.
__count	<Double>	Read-only	count parameter

Operations

(click to see Properties)

ADD:| DELETE:| GET| GET (ALL)| COUNT

Some options that you can use for each operations:

- **Getting warnings in response:**NITRO allows you to get warnings in an operation by specifying the "warning" query parameter as "yes". For example, to get warnings while connecting to the Citrix ADC appliance, the URL is as follows:

  http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/login?warning=yes

  If any, the warnings are displayed in the response payload with the HTTP code "209 X-NITRO-WARNING".

- **Authenticated access for individual NITRO operations:**NITRO allows you to logon to the Citrix ADC appliance to perform individual operations. You can use this option instead of creating a NITRO session (using the login object) and then using that session to perform all operations,

  To do this, you must specify the username and password in the request header of the NITRO request as follows:

  X-NITRO-USER:<username>

  X-NITRO-PASS:<password>

  Note: In such cases, make sure that the request header DOES not include the following:

  Cookie:NITRO_AUTH_TOKEN=<tokenvalue>

`*Note:` *

Mandatory parameters are marked in redand placeholder content is marked in <green>.

add

URL:http://<Citrix-ADC-IP-address(NSIP)/nitro/v1/config/appflowglobal_appflowpolicy_binding

HTTP Method:PUT

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Content-Type:application/json

Request Payload:

```
{

"appflowglobal_appflowpolicy_binding":{

<b>"policyname":<String_value>,

</b><b>"priority":<Double_value>,

</b>"gotopriorityexpression":<String_value>,

"type":<String_value>,

"invoke":<Boolean_value>,

"labeltype":<String_value>,

"labelname":<String_value>

}}
```

Response:

HTTP Status Code on Success: 201 Created HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error

delete

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/appflowglobal_appflowpolicy_binding

Query-parameters:

args

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/appflowglobal_appflowpolicy_binding?**args=policyname:<String_value>,type:<String_value>,priority:<Double_value>**

HTTP Method:DELETE

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue>

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error

get

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/appflowglobal_appflowpolicy_binding

Query-parameters:

filter

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/appflowglobal_appflowpolicy_binding?**filter=property-name1:property-value1,property-name2:property-value2**

Use this query-parameter to get the filtered set of appflowglobal_appflowpolicy_binding resources configured on Citrix ADC. Filtering can be done on any of the properties of the resource.

pagination

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/appflowglobal_appflowpolicy_binding?**pagesize=#no;pageno=#no**

Use this query-parameter to get the appflowglobal_appflowpolicy_binding resources in chunks.

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "appflowglobal_appflowpolicy_binding": [ {

"priority":<Double_value>,

"globalbindtype":<String_value>,

"policyname":<String_value>,

"labelname":<String_value>,

"gotopriorityexpression":<String_value>,

"invoke":<Boolean_value>,

"type":<String_value>,

"labeltype":<String_value>,

"flowtype":<Double_value>,

"numpol":<Double_value>

}]}
```

get (all)

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/appflowglobal_appflowpolicy_binding

Query-parameters:

bulkbindings

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/appflowglobal_appflowpolicy_binding?**bulkbindings=yes**

NITRO allows you to fetch bindings in bulk.

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "appflowglobal_appflowpolicy_binding": [ {

"priority":<Double_value>,

"globalbindtype":<String_value>,

"policyname":<String_value>,

"labelname":<String_value>,

"gotopriorityexpression":<String_value>,

"invoke":<Boolean_value>,

"type":<String_value>,

"labeltype":<String_value>,

"flowtype":<Double_value>,
```

```
"numpol":<Double_value>

}]}
```

count

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/appflowglobal_appflowpolicy_binding?**count=yes**

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "appflowglobal_appflowpolicy_binding": [ { "__count": "#no"} ] }
```

appflowglobal_binding

Binding object which returns the resources bound to appflowglobal.

Properties

(click to see Operations)

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
appflowglobal_appflowpolicy_binding	<appflowglobal_appflowpolicy_binding[]>	Read-only	appflowpolicy that can be bound to appflowglobal.

Operations

(click to see Properties)

GET| GET (ALL)

Some options that you can use for each operations:

- **Getting warnings in response:**NITRO allows you to get warnings in an operation by specifying the "warning" query parameter as "yes". For example, to get warnings while connecting to the Citrix ADC appliance, the URL is as follows:

  http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/login?warning=yes

  If any, the warnings are displayed in the response payload with the HTTP code "209 X-NITRO-WARNING".

- **Authenticated access for individual NITRO operations:**NITRO allows you to logon to the Citrix ADC appliance to perform individual operations. You can use this option instead of creating a NITRO session (using the login object) and then using that session to perform all operations,

  To do this, you must specify the username and password in the request header of the NITRO request as follows:

  X-NITRO-USER:<username>

  X-NITRO-PASS:<password>

  Note: In such cases, make sure that the request header DOES not include the following:

  Cookie:NITRO_AUTH_TOKEN=<tokenvalue>

`*Note:`*

Mandatory parameters are marked in redand placeholder content is marked in <green>.

get

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/appflowglobal_binding

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "appflowglobal_binding": [ {

"appflowglobal_appflowpolicy_binding":<appflowglobal_appflowpolicy_binding
[]_value>

}]}
```

get (all)

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/appflowglobal_binding

Query-parameters:

bulkbindings

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/appflowglobal_binding?**bulkbindings=yes**

NITRO allows you to fetch bindings in bulk.

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "appflowglobal_binding": [ {

"appflowglobal_appflowpolicy_binding":<appflowglobal_appflowpolicy_binding
[]_value>

}]}
```

appflowparam

Configuration for AppFlow parameter resource.

Properties

(click to see Operations)

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
templaterefresh	<Double>	Read-write	Refresh interval, in seconds, at which to export the template data. Because data transmission is in UDP, the templates must be resent at regular intervals. Default value: 600

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
			Minimum value = 60 Maximum value = 3600
appnamerefresh	<Double>	Read-write	Interval, in seconds, at which to send Appnames to the configured collectors. Appname refers to the name of an entity (virtual server, service, or service group) in the Citrix ADC. Default value: 600 Minimum value = 60 Maximum value = 3600
flowrecordinterval	<Double>	Read-write	Interval, in seconds, at which to send flow records to the configured collectors. Default value: 60 Minimum value = 60 Maximum value = 3600
securityinsightrecordinterval	<Double>	Read-write	Interval, in seconds, at which to send security insight flow records to the configured collectors. Default value: 600

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
			Minimum value = 60 Maximum value = 3600
udppmtu	<Double>	Read-write	MTU, in bytes, for IPFIX UDP packets. Default value: 1472 Minimum value = 128 Maximum value = 1472
httpurl	<String>	Read-write	Include the http URL that the Citrix ADC received from the client. Default value: DISABLED Possible values = ENABLED, DISABLED
aaausername	<String>	Read-write	Enable AppFlow AAA Username logging. Default value: DISABLED Possible values = ENABLED, DISABLED
httpcookie	<String>	Read-write	Include the cookie that was in the HTTP request the appliance received from the client.

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
			Default value: DISABLED Possible values = ENABLED, DISABLED
httpreferer	<String>	Read-write	Include the web page that was last visited by the client. Default value: DISABLED Possible values = ENABLED, DISABLED
httpmethod	<String>	Read-write	Include the method that was specified in the HTTP request that the appliance received from the client. Default value: DISABLED Possible values = ENABLED, DISABLED
httphost	<String>	Read-write	Include the host identified in the HTTP request that the appliance received from the client. Default value: DISABLED

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
			Possible values = ENABLED, DISABLED
httpuseragent	<String>	Read-write	Include the client application through which the HTTP request was received by the Citrix ADC. Default value: DISABLED Possible values = ENABLED, DISABLED
clienttrafficonly	<String>	Read-write	Generate AppFlow records for only the traffic from the client. Default value: NO Possible values = YES, NO
httpcontenttype	<String>	Read-write	Include the HTTP Content-Type header sent from the server to the client to determine the type of the content sent. Default value: DISABLED Possible values = ENABLED, DISABLED
httpauthorization	<String>	Read-write	Include the HTTP Authorization header information.

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
			Default value: DISABLED Possible values = ENABLED, DISABLED
httpvia	<String>	Read-write	Include the httpVia header which contains the IP address of proxy server through which the client accessed the server. Default value: DISABLED Possible values = ENABLED, DISABLED
httpxforwardedfor	<String>	Read-write	Include the httpXForwardedFor header, which contains the original IP Address of the client using a proxy server to access the server. Default value: DISABLED Possible values = ENABLED, DISABLED
httplocation	<String>	Read-write	Include the HTTP location headers returned from the HTTP responses.

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
			Default value: DISABLED Possible values = ENABLED, DISABLED
httpsetcookie	<String>	Read-write	Include the Set-cookie header sent from the server to the client in response to a HTTP request. Default value: DISABLED Possible values = ENABLED, DISABLED
httpsetcookie2	<String>	Read-write	Include the Set-cookie header sent from the server to the client in response to a HTTP request. Default value: DISABLED Possible values = ENABLED, DISABLED
connectionchaining	<String>	Read-write	Enable connection chaining so that the client server flows of a connection are linked. Also the connection chain ID is propagated across Citrix ADCs, so that in a multi-hop

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
			environment the flows belonging to the same logical connection are linked. This id is also logged as part of appflow record. Default value: DISABLED Possible values = ENABLED, DISABLED
httpdomain	<String>	Read-write	Include the http domain request to be exported. Default value: DISABLED Possible values = ENABLED, DISABLED
skipcacheredirection httptransaction	<String>	Read-write	Skip Cache http transaction. This HTTP transaction is specific to Cache Redirection module. In Case of Cache Miss there will be another HTTP transaction initiated by the cache server. Default value: DISABLED Possible values = ENABLED, DISABLED

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
identifiername	<String>	Read-write	Include the stream identifier name to be exported. Default value: DISABLED Possible values = ENABLED, DISABLED
identifiersessionname	<String>	Read-write	Include the stream identifier session name to be exported. Default value: DISABLED Possible values = ENABLED, DISABLED
observationdomainid	<Double>	Read-write	An observation domain groups a set of Citrix ADCs based on deployment: cluster, HA etc. A unique Observation Domain ID is required to be assigned to each such group. Default value: 0 Minimum value = 1000
observationdomainname	<String>	Read-write	Name of the Observation Domain defined by the observation domain ID.

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
			Maximum length = 127
subscriberawareness	<String>	Read-write	Enable this option for logging end user MSISDN in L4/L7 appflow records. Default value: DISABLED Possible values = ENABLED, DISABLED
subscriberidobfuscation	<String>	Read-write	Enable this option for obfuscating MSISDN in L4/L7 appflow records. Default value: DISABLED Possible values = ENABLED, DISABLED
subscriberidobfuscationalgo	<String>	Read-write	Algorithm(MD5 or SHA256) to be used for obfuscating MSISDN. Default value: MD5 Possible values = MD5, SHA256
gxsessionreporting	<String>	Read-write	Enable this option for Gx session reporting. Default value: DISABLED

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
			Possible values = ENABLED, DISABLED
securityinsighttraffic	<String>	Read-write	Enable/disable the feature individually on appflow action. Default value: DISABLED Possible values = ENABLED, DISABLED
cacheinsight	<String>	Read-write	Flag to determine whether cache records need to be exported or not. If this flag is true and IC is enabled, cache records are exported instead of L7 HTTP records. Default value: DISABLED Possible values = ENABLED, DISABLED
videoinsight	<String>	Read-write	Enable/disable the feature individually on appflow action. Default value: DISABLED Possible values = ENABLED, DISABLED

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
httpquerywithurl	<String>	Read-write	Include the HTTP query segment along with the URL that the Citrix ADC received from the client. Default value: DISABLED Possible values = ENABLED, DISABLED
urlcategory	<String>	Read-write	Include the URL category record. Default value: DISABLED Possible values = ENABLED, DISABLED
lsnlogging	<String>	Read-write	On enabling this option, the Citrix ADC will send the Large Scale Nat(LSN) records to the configured collectors. Default value: DISABLED Possible values = ENABLED, DISABLED
cqareporting	<String>	Read-write	TCP CQA reporting enable/disable knob.

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
			Default value: DISABLED Possible values = ENABLED, DISABLED
emailaddress	<String>	Read-write	Enable AppFlow user email-id logging. Default value: DISABLED Possible values = ENABLED, DISABLED
usagerecordinterval	<Double>	Read-write	On enabling this option, the NGS will send bandwidth usage record to configured collectors. Default value: 0 Minimum value = 0 Maximum value = 7200
metrics	<String>	Read-write	Enable Citrix ADC Stats to be sent to the Telemetry Agent. Default value: DISABLED Possible values = ENABLED, DISABLED

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
events	<String>	Read-write	Enable Events to be sent to the Telemetry Agent. Default value: DISABLED Possible values = ENABLED, DISABLED
auditlogs	<String>	Read-write	Enable Auditlogs to be sent to the Telemetry Agent. Default value: DISABLED Possible values = ENABLED, DISABLED
observationpointid	<Double>	Read-write	An observation point ID is identifier for the Citrix ADC from which appflow records are being exported. By default, the Citrix ADC IP is the observation point ID. Minimum value = 1
distributedtracing	<String>	Read-write	Enable generation of the distributed tracing templates in the Appflow records. Default value: DISABLED

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
			Possible values = ENABLED, DISABLED
disttracingsamplingrate	<Double>	Read-write	Sampling rate for Distributed Tracing. Default value: 0 Minimum value = 0 Maximum value = 100
tcpattackcounterinterval	<Double>	Read-write	Interval, in seconds, at which to send tcp attack counters to the configured collectors. If 0 is configured, the record is not sent. Default value: 0 Minimum value = 0 Maximum value = 3600
tcpburstreporting	<String>	Read-only	TCP burst reporting enable/disable knob. Default value: ENABLED Possible values = ENABLED, DISABLED
tcpburstreportingthreshold	<Double>	Read-only	TCP burst reporting threshold. Default value: 1500

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
			Minimum value = 10 Maximum value = 5000

Operations

(click to see Properties)

UPDATE| UNSET| GET (ALL)

Some options that you can use for each operations:

- **Getting warnings in response:**NITRO allows you to get warnings in an operation by specifying the "warning" query parameter as "yes". For example, to get warnings while connecting to the Citrix ADC appliance, the URL is as follows:

  http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/login?warning=yes

  If any, the warnings are displayed in the response payload with the HTTP code "209 X-NITRO-WARNING".

- **Authenticated access for individual NITRO operations:**NITRO allows you to logon to the Citrix ADC appliance to perform individual operations. You can use this option instead of creating a NITRO session (using the login object) and then using that session to perform all operations,

  To do this, you must specify the username and password in the request header of the NITRO request as follows:

  X-NITRO-USER:<username>

  X-NITRO-PASS:<password>

  Note: In such cases, make sure that the request header DOES not include the following:

  Cookie:NITRO_AUTH_TOKEN=<tokenvalue>

`*Note:`*

Mandatory parameters are marked in redand placeholder content is marked in <green>.

update

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/appflowparam

HTTP Method:PUT

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Content-Type:application/json

Request Payload:

```
{"appflowparam":{

"templaterefresh":<Double_value>,

"appnamerefresh":<Double_value>,

"flowrecordinterval":<Double_value>,

"securityinsightrecordinterval":<Double_value>,

"udppmtu":<Double_value>,

"httpurl":<String_value>,

"aaausername":<String_value>,

"httpcookie":<String_value>,

"httpreferer":<String_value>,

"httpmethod":<String_value>,

"httphost":<String_value>,

"httpuseragent":<String_value>,

"clienttrafficonly":<String_value>,

"httpcontenttype":<String_value>,
```

```
"httpauthorization":<String_value>,

"httpvia":<String_value>,

"httpxforwardedfor":<String_value>,

"httplocation":<String_value>,

"httpsetcookie":<String_value>,

"httpsetcookie2":<String_value>,

"connectionchaining":<String_value>,

"httpdomain":<String_value>,

"skipcacheredirectionhttptransaction":<String_value>,

"identifiername":<String_value>,

"identifiersessionname":<String_value>,

"observationdomainid":<Double_value>,

"observationdomainname":<String_value>,

"subscriberawareness":<String_value>,

"subscriberidobfuscation":<String_value>,

"subscriberidobfuscationalgo":<String_value>,

"gxsessionreporting":<String_value>,

"securityinsighttraffic":<String_value>,

"cacheinsight":<String_value>,

"videoinsight":<String_value>,

"httpquerywithurl":<String_value>,

"urlcategory":<String_value>,
```

```
"lsnlogging":<String_value>,

"cqareporting":<String_value>,

"emailaddress":<String_value>,

"usagerecordinterval":<Double_value>,

"metrics":<String_value>,

"events":<String_value>,

"auditlogs":<String_value>,

"observationpointid":<Double_value>,

"distributedtracing":<String_value>,

"disttracingsamplingrate":<Double_value>,

"tcpattackcounterinterval":<Double_value>

}}
```

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error

unset

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/appflowparam?**action=unset**

HTTP Method:POST

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Content-Type:application/json

Request Payload:

```
{"appflowparam":{
"templaterefresh":true,
"appnamerefresh":true,
"flowrecordinterval":true,
"securityinsightrecordinterval":true,
"udppmtu":true,
"httpurl":true,
"aaausername":true,
"httpcookie":true,
"httpreferer":true,
"httpmethod":true,
"httphost":true,
"httpuseragent":true,
"clienttrafficonly":true,
"httpcontenttype":true,
"httpauthorization":true,
"httpvia":true,
"httpxforwardedfor":true,
"httplocation":true,
"httpsetcookie":true,
"httpsetcookie2":true,
```

```
"connectionchaining":true,

"httpdomain":true,

"skipcacheredirectionhttptransaction":true,

"identifiername":true,

"identifiersessionname":true,

"observationdomainid":true,

"observationdomainname":true,

"subscriberawareness":true,

"subscriberidobfuscation":true,

"subscriberidobfuscationalgo":true,

"gxsessionreporting":true,

"securityinsighttraffic":true,

"cacheinsight":true,

"videoinsight":true,

"httpquerywithurl":true,

"urlcategory":true,

"lsnlogging":true,

"cqareporting":true,

"emailaddress":true,

"usagerecordinterval":true,

"metrics":true,

"events":true,
```

```
"auditlogs":true,

"observationpointid":true,

"distributedtracing":true,

"disttracingsamplingrate":true,

"tcpattackcounterinterval":true

}}
```

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error

get (all)

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/appflowparam

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "appflowparam": [ {
```

```
"templaterefresh":<Double_value>,

"appnamerefresh":<Double_value>,

"flowrecordinterval":<Double_value>,

"securityinsightrecordinterval":<Double_value>,

"udppmtu":<Double_value>,

"httpurl":<String_value>,

"aaausername":<String_value>,

"httpcookie":<String_value>,

"httpreferer":<String_value>,

"httpmethod":<String_value>,

"httphost":<String_value>,

"httpuseragent":<String_value>,

"clienttrafficonly":<String_value>,

"httpcontenttype":<String_value>,

"httpauthorization":<String_value>,

"httpvia":<String_value>,

"httpxforwardedfor":<String_value>,

"httplocation":<String_value>,

"httpsetcookie":<String_value>,

"httpsetcookie2":<String_value>,

"connectionchaining":<String_value>,

"httpdomain":<String_value>,
```

```
"skipcacheredirectionhttptransaction":<String_value>,

"identifiername":<String_value>,

"identifiersessionname":<String_value>,

"observationdomainid":<Double_value>,

"observationpointid":<Double_value>,

"observationdomainname":<String_value>,

"securityinsighttraffic":<String_value>,

"subscriberawareness":<String_value>,

"subscriberidobfuscation":<String_value>,

"subscriberidobfuscationalgo":<String_value>,

"gxsessionreporting":<String_value>,

"cacheinsight":<String_value>,

"videoinsight":<String_value>,

"httpquerywithurl":<String_value>,

"tcpburstreporting":<String_value>,

"tcpburstreportingthreshold":<Double_value>,

"urlcategory":<String_value>,

"lsnlogging":<String_value>,

"cqareporting":<String_value>,

"emailaddress":<String_value>,

"usagerecordinterval":<Double_value>,

"metrics":<String_value>,
```

```
"events":<String_value>,

"auditlogs":<String_value>,

"distributedtracing":<String_value>,

"disttracingsamplingrate":<Double_value>,

"tcpattackcounterinterval":<Double_value>

}]}
```

appflowpolicy

Configuration for AppFlow policy resource.

Properties

(click to see Operations)

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
name	<String>	Read-write	Name for the policy. Must begin with an ASCII alphabetic or underscore (_) character, and must contain only ASCII alphanumeric, underscore, hash (#), period (.), space, colon (:), at (@), equals (=), and hyphen (-) characters. The following requirement applies only to the Citrix ADC CLI:

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
			If the name includes one or more spaces, enclose the name in double or single quotation marks (for example, "my appflow policy" or 'my appflow policy').
rule	<String>	Read-write	Expression or other value against which the traffic is evaluated. Must be a Boolean expression. The following requirements apply only to the Citrix ADC CLI: * If the expression includes one or more spaces, enclose the entire expression in double quotation marks. * If the expression itself includes double quotation marks, escape the quotations by using the \ character. * Alternatively, you can use single quotation marks to

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
			enclose the rule, in which case you do not have to escape the double quotation marks.
action	<String>	Read-write	Name of the action to be associated with this policy.
undefaction	<String>	Read-write	Name of the appflow action to be associated with this policy when an undef event occurs.
comment	<String>	Read-write	Any comments about this policy.
newname	<String>	Read-write	New name for the policy. Must begin with an ASCII alphabetic or underscore (_)character, and must contain only ASCII alphanumeric, underscore, hash (#), period (.), space, colon (:), at (@), equals (=), and hyphen (-) characters. The following requirement applies only to the Citrix ADC CLI:

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
			If the name includes one or more spaces, enclose the name in double or single quotation marks (for example, "my appflow policy" or 'my appflow policy'). Minimum length = 1
hits	<Double>	Read-only	Number of hits.
undefhits	<Double>	Read-only	Number of policy UNDEF hits.
description	<String>	Read-only	Description of the policy.
__count	<Double>	Read-only	count parameter

Operations

(click to see Properties)

ADD| DELETE| UPDATE| UNSET| RENAME| GET (ALL)| GET| COUNT

Some options that you can use for each operations:

- **Getting warnings in response:**NITRO allows you to get warnings in an operation by specifying the "warning" query parameter as "yes". For example, to get warnings while connecting to the Citrix ADC appliance, the URL is as follows:

  http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/login?warning=yes

  If any, the warnings are displayed in the response payload with the HTTP code "209 X-NITRO-WARNING".

- **Authenticated access for individual NITRO operations:**NITRO allows you to logon to the Citrix ADC appliance to perform individual operations. You can use this option instead of

creating a NITRO session (using the login object) and then using that session to perform all operations,

To do this, you must specify the username and password in the request header of the NITRO request as follows:

X-NITRO-USER:<username>

X-NITRO-PASS:<password>

Note: In such cases, make sure that the request header DOES not include the following:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue>

`*Note:`*

Mandatory parameters are marked in redand placeholder content is marked in <green>.

add

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/appflowpolicy

HTTP Method:POST

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Content-Type:application/json

Request Payload:

```
{"appflowpolicy":{

<b>"name":<String_value>,

</b><b>"rule":<String_value>,

</b><b>"action":<String_value>,

</b>"undefaction":<String_value>,

"comment":<String_value>

}}
```

Response:

HTTP Status Code on Success: 201 Created HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error

delete

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/appflowpolicy/name_value<String>

HTTP Method:DELETE

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue>

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error

update

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/appflowpolicy

HTTP Method:PUT

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Content-Type:application/json

Request Payload:

```
{"appflowpolicy":{

<b>"name":<String_value>,

</b>"rule":<String_value>,

"action":<String_value>,
```

```
"undefaction":<String_value>,

"comment":<String_value>

}}
```

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error

unset

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/appflowpolicy?**action=unset**

HTTP Method:POST

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Content-Type:application/json

Request Payload:

```
{"appflowpolicy":{

<b>"name":<String_value>,

</b>"undefaction":true,

"comment":true

}}
```

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error

rename

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/appflowpolicy?**action=rename**

HTTP Method:POST

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Content-Type:application/json

Request Payload:

```
{"appflowpolicy":{

<b>"name":<String_value>,

</b><b>"newname":<String_value>

</b>}}
```

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error

get (all)

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/appflowpolicy

Query-parameters:

attrs

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/appflowpolicy?**attrs=property-name1,property-name2**

Use this query parameter to specify the resource details that you want to retrieve.

filter

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/appflowpolicy?**filter=property-name1:property-val1,property-name2:property-val2**

Use this query-parameter to get the filtered set of appflowpolicy resources configured on Citrix ADC. Filtering can be done on any of the properties of the resource.

view

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/appflowpolicy?**view=summary**

Note: By default, the retrieved results are displayed in detail view (?view=detail).

pagination

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/appflowpolicy?**pagesize=#no;pageno=#no**

Use this query-parameter to get the appflowpolicy resources in chunks.

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "appflowpolicy": [ {

"name":<String_value>,

"rule":<String_value>,

"action":<String_value>,

"undefaction":<String_value>,
```

```
"hits":<Double_value>,

"undefhits":<Double_value>,

"description":<String_value>,

"comment":<String_value>

}]}
```

get

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/appflowpolicy/name_value<String>

Query-parameters:

attrs

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/appflowpolicy/name_value<String>?**attrs=property-name1,property-name2**

Use this query parameter to specify the resource details that you want to retrieve.

view

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/appflowpolicy/name_value<String>?**view=summary**

Note: By default, the retrieved results are displayed in detail view (?view=detail).

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "appflowpolicy": [ {

"name":<String_value>,

"rule":<String_value>,

"action":<String_value>,

"undefaction":<String_value>,

"hits":<Double_value>,

"undefhits":<Double_value>,

"description":<String_value>,

"comment":<String_value>

}]}
```

count

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/appflowpolicy?**count=yes**

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "appflowpolicy": [ { "__count": "#no"} ] }
```

appflowpolicylabel

Configuration for AppFlow policy label resource.

Properties

(click to see Operations)

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
labelname	<String>	Read-write	Name of the AppFlow policy label. Must begin with an ASCII alphabetic or underscore (_) character, and must contain only ASCII alphanumeric, underscore, hash (#), period (.), space, colon (:), at (@), equals (=), and hyphen (-) characters. The following requirement applies only to the Citrix ADC CLI: If the name includes one or more spaces, enclose the name in double or single quotation marks (for example,

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
			"my appflow policylabel" or 'my appflow policylabel'). Minimum length = 1
policylabeltype	<String>	Read-write	Type of traffic evaluated by the policies bound to the policy label. Default value: HTTP Possible values = HTTP, OTHERTCP
newname	<String>	Read-write	New name for the policy label. Must begin with an ASCII alphabetic or underscore (_) character, and must contain only ASCII alphanumeric, underscore, hash (#), period (.), space, colon (:), at (@), equals (=), and hyphen (-) characters. The following requirement applies only to the Citrix ADC CLI: If the name includes one or more spaces, enclose the name in double or single quotation

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
			marks (for example, "my appflow policylabel" or 'my appflow policylabel'). Minimum length = 1
numpol	<Double>	Read-only	Number of polices bound to the policy label.
hits	<Double>	Read-only	Number of times the policy label was invoked.
policyname	<String>	Read-only	Name of the AppFlow policy.
priority	<Double>	Read-only	Specifies the priority of the policy.
gotopriorityexpression	<String>	Read-only	Expression specifying the priority of the next policy which will get evaluated if the current policy rule evaluates to TRUE.
labeltype	<String>	Read-only	Type of policy label to be invoked. Possible values = vserver, policylabel
invoke_labelname	<String>	Read-only	Name of the label to invoke if the current policy evaluates to TRUE.

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
flowtype	<Double>	Read-only	Flowtype of the bound AppFlow policy.
description	<String>	Read-only	Description of the policylabel.
__count	<Double>	Read-only	count parameter

Operations

(click to see Properties)

ADD| DELETE| RENAME| GET (ALL)| GET| COUNT

Some options that you can use for each operations:

- **Getting warnings in response:**NITRO allows you to get warnings in an operation by specifying the "warning" query parameter as "yes". For example, to get warnings while connecting to the Citrix ADC appliance, the URL is as follows:

  http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/login?warning=yes

  If any, the warnings are displayed in the response payload with the HTTP code "209 X-NITRO-WARNING".

- **Authenticated access for individual NITRO operations:**NITRO allows you to logon to the Citrix ADC appliance to perform individual operations. You can use this option instead of creating a NITRO session (using the login object) and then using that session to perform all operations,

  To do this, you must specify the username and password in the request header of the NITRO request as follows:

  X-NITRO-USER:<username>

  X-NITRO-PASS:<password>

  Note: In such cases, make sure that the request header DOES not include the following:

  Cookie:NITRO_AUTH_TOKEN=<tokenvalue>

`*Note:`*

Mandatory parameters are marked in redand placeholder content is marked in <green>.

add

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/appflowpolicylabel

HTTP Method:POST

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Content-Type:application/json

Request Payload:

```
{"appflowpolicylabel":{

<b>"labelname":<String_value>,

</b>"policylabeltype":<String_value>

}}
```

Response:

HTTP Status Code on Success: 201 Created HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error

delete

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/appflowpolicylabel/labelname_value<String>

HTTP Method:DELETE

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue>

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error

rename

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/appflowpolicylabel?**action=rename**

HTTP Method:POST

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Content-Type:application/json

Request Payload:

```
{"appflowpolicylabel":{

<b>"labelname":<String_value>,

</b><b>"newname":<String_value>

</b>}}
```

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error

get (all)

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/appflowpolicylabel

Query-parameters:

attrs

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/appflowpolicylabel?**attrs=property-name1,property-name2**

Use this query parameter to specify the resource details that you want to retrieve.

filter

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/appflowpolicylabel?**filter=property-name1:property-val1,property-name2:property-val2**

Use this query-parameter to get the filtered set of appflowpolicylabel resources configured on Citrix ADC. Filtering can be done on any of the properties of the resource.

view

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/appflowpolicylabel?**view=summary**

Note: By default, the retrieved results are displayed in detail view (?view=detail).

pagination

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/appflowpolicylabel?**pagesize=#no;pageno=#no**

Use this query-parameter to get the appflowpolicylabel resources in chunks.

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "appflowpolicylabel": [ {

"labelname":<String_value>,

"policylabeltype":<String_value>,
```

```
"numpol":<Double_value>,

"hits":<Double_value>,

"policyname":<String_value>,

"priority":<Double_value>,

"gotopriorityexpression":<String_value>,

"labeltype":<String_value>,

"invoke_labelname":<String_value>,

"flowtype":<Double_value>,

"description":<String_value>

}]}
```

get

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/appflowpolicylabel/labelname_value<String>

Query-parameters:

attrs

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/appflowpolicylabel/labelname_value<String>?**attrs=property-name1,property-name2**

Use this query parameter to specify the resource details that you want to retrieve.

view

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/appflowpolicylabel/labelname_value<String>?**view=summary**

Note: By default, the retrieved results are displayed in detail view (?view=detail).

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "appflowpolicylabel": [ {

"labelname":<String_value>,

"policylabeltype":<String_value>,

"numpol":<Double_value>,

"hits":<Double_value>,

"policyname":<String_value>,

"priority":<Double_value>,

"gotopriorityexpression":<String_value>,

"labeltype":<String_value>,

"invoke_labelname":<String_value>,

"flowtype":<Double_value>,

"description":<String_value>

}]}
```

count

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/appflowpolicylabel?**count=yes**

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "appflowpolicylabel": [ { "__count": "#no"} ] }
```

appflowpolicylabel_appflowpolicy_binding

Binding object showing the appflowpolicy that can be bound to appflowpolicylabel.

Properties

(click to see Operations)

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
priority	<Double>	Read-write	Specifies the priority of the policy.
policyname	<String>	Read-write	Name of the AppFlow policy.
labelname	<String>	Read-write	Name of the policy label to which to bind the policy.

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
			Minimum length = 1
invoke_labelname	<String>	Read-write	Name of the label to invoke if the current policy evaluates to TRUE.
gotopriorityexpression	<String>	Read-write	Expression specifying the priority of the next policy which will get evaluated if the current policy rule evaluates to TRUE.
invoke	<Boolean>	Read-write	Invoke policies bound to a virtual server or a user-defined policy label. After the invoked policies are evaluated, the flow returns to the policy with the next priority.
labeltype	<String>	Read-write	Type of policy label to be invoked. Possible values = vserver, policylabel
__count	<Double>	Read-write	count parameter

Operations

(click to see Properties)

ADD:| DELETE:| GET| GET (ALL)| COUNT

Some options that you can use for each operations:

- **Getting warnings in response:**NITRO allows you to get warnings in an operation by specifying the "warning" query parameter as "yes". For example, to get warnings while connecting to the Citrix ADC appliance, the URL is as follows:

  http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/login?warning=yes

  If any, the warnings are displayed in the response payload with the HTTP code "209 X-NITRO-WARNING".

- **Authenticated access for individual NITRO operations:**NITRO allows you to logon to the Citrix ADC appliance to perform individual operations. You can use this option instead of creating a NITRO session (using the login object) and then using that session to perform all operations,

  To do this, you must specify the username and password in the request header of the NITRO request as follows:

  X-NITRO-USER:<username>

  X-NITRO-PASS:<password>

  Note: In such cases, make sure that the request header DOES not include the following:

  Cookie:NITRO_AUTH_TOKEN=<tokenvalue>

`*Note:`*

Mandatory parameters are marked in redand placeholder content is marked in <green>.

add

URL:http://<Citrix-ADC-IP-address(NSIP)/nitro/v1/config/appflowpolicylabel_appflowpolicy_binding

HTTP Method:PUT

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Content-Type:application/json

Request Payload:

```
{
```

```
"appflowpolicylabel_appflowpolicy_binding":{

<b>"labelname":<String_value>,

</b><b>"policyname":<String_value>,

</b><b>"priority":<Double_value>,

</b>"gotopriorityexpression":<String_value>,

"invoke":<Boolean_value>,

"labeltype":<String_value>,

"invoke_labelname":<String_value>

}}
```

Response:

HTTP Status Code on Success: 201 Created HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error

delete

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/appflowpolicylabel_appflowpolicy_binding/labelname_value<String>

Query-parameters:

args

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/appflowpolicylabel_appflowpolicy_binding/labelname_value<String>?**args=policyname:<String_value>,priority:<Double_value>**

HTTP Method:DELETE

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue>

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error

get

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/appflowpolicylabel_appflowpolicy_binding/labelname_value<String>

Query-parameters:

filter

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/appflowpolicylabel_appflowpolicy_binding/labelname_value<String>?**filter=property-name1:property-value1,property-name2:property-value2**

Use this query-parameter to get the filtered set of appflowpolicylabel_appflowpolicy_binding resources configured on Citrix ADC. Filtering can be done on any of the properties of the resource.

pagination

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/appflowpolicylabel_appflowpolicy_binding/labelname_value<String>?**pagesize=#no;pageno=#no**

Use this query-parameter to get the appflowpolicylabel_appflowpolicy_binding resources in chunks.

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "appflowpolicylabel_appflowpolicy_binding": [ {

"priority":<Double_value>,

"policyname":<String_value>,

"labelname":<String_value>,

"invoke_labelname":<String_value>,

"gotopriorityexpression":<String_value>,

"invoke":<Boolean_value>,

"labeltype":<String_value>

}]}
```

get (all)

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/appflowpolicylabel_appflowpolicy_binding

Query-parameters:

bulkbindings

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/appflowpolicylabel_appflowpolicy_binding?**bulkbindings=yes**

NITRO allows you to fetch bindings in bulk.

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "appflowpolicylabel_appflowpolicy_binding": [ {

"priority":<Double_value>,

"policyname":<String_value>,

"labelname":<String_value>,

"invoke_labelname":<String_value>,

"gotopriorityexpression":<String_value>,

"invoke":<Boolean_value>,

"labeltype":<String_value>

}]}
```

count

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/appflowpolicylabel_appflowpolicy_binding/labelname_value<String>?**count=yes**

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{"appflowpolicylabel_appflowpolicy_binding": [ { "__count": "#no"} ] }
```

appflowpolicylabel_binding

Binding object which returns the resources bound to appflowpolicylabel.

Properties

(click to see Operations)

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
labelname	<String>	Read-write	Name of the policy label about which to display detailed information. Minimum length = 1
appflowpolicylabel_appflowpolicy_binding	<appflowpolicylabel_appflowpolicy_binding[]>	Read-only	appflowpolicy that can be bound to appflowpolicylabel.

Operations

(click to see Properties)

GET| GET (ALL)

Some options that you can use for each operations:

- **Getting warnings in response:**NITRO allows you to get warnings in an operation by specifying the "warning" query parameter as "yes". For example, to get warnings while connecting to the Citrix ADC appliance, the URL is as follows:

  http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/login?warning=yes

  If any, the warnings are displayed in the response payload with the HTTP code "209 X-NITRO-WARNING".

- **Authenticated access for individual NITRO operations:**NITRO allows you to logon to the Citrix ADC appliance to perform individual operations. You can use this option instead of creating a NITRO session (using the login object) and then using that session to perform all operations,

  To do this, you must specify the username and password in the request header of the NITRO request as follows:

  X-NITRO-USER:<username>

  X-NITRO-PASS:<password>

  Note: In such cases, make sure that the request header DOES not include the following:

  Cookie:NITRO_AUTH_TOKEN=<tokenvalue>

`*Note:`*

Mandatory parameters are marked in redand placeholder content is marked in <green>.

get

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/appflowpolicylabel_binding/labelname_value<String>

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "appflowpolicylabel_binding": [ {

"labelname":<String_value>,

"appflowpolicylabel_appflowpolicy_binding":<appflowpolicylabel_appflowpoli
cy_binding[]_value>

}]}
```

get (all)

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/appflowpolicylabel_binding

Query-parameters:

bulkbindings

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/appflowpolicylabel_binding?
bulkbindings=yes

NITRO allows you to fetch bindings in bulk.

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "appflowpolicylabel_binding": [ {

"labelname":<String_value>,

"appflowpolicylabel_appflowpolicy_binding":<appflowpolicylabel_appflowpoli
cy_binding[]_value>

}]}
```

appflowpolicy_appflowglobal_binding

Binding object showing the appflowglobal that can be bound to appflowpolicy.

Properties

(click to see Operations)

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
boundto	<String>	Read-write	Location where policy is bound.
name	<String>	Read-write	Name of the policy about which to display detailed information.
priority	<Double>	Read-only	Specifies the priority of the policy.
labelname	<String>	Read-only	Name of the label to invoke if the current policy rule evaluates to TRUE.
gotopriorityexpression	<String>	Read-only	Expression specifying the priority of the next policy which will get evaluated if the

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
			current policy rule evaluates to TRUE.
labeltype	<String>	Read-only	Type of policy label invocation. Possible values = vserver, policylabel
activepolicy	<Integer>	Read-only	Indicates whether policy is bound or not.
__count	<Double>	Read-only	count parameter

Operations

(click to see Properties)

GET| GET (ALL)| COUNT

Some options that you can use for each operations:

- **Getting warnings in response:**NITRO allows you to get warnings in an operation by specifying the "warning" query parameter as "yes". For example, to get warnings while connecting to the Citrix ADC appliance, the URL is as follows:

  http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/login?warning=yes

  If any, the warnings are displayed in the response payload with the HTTP code "209 X-NITRO-WARNING".

- **Authenticated access for individual NITRO operations:**NITRO allows you to logon to the Citrix ADC appliance to perform individual operations. You can use this option instead of creating a NITRO session (using the login object) and then using that session to perform all operations,

  To do this, you must specify the username and password in the request header of the NITRO request as follows:

  X-NITRO-USER:<username>

  X-NITRO-PASS:<password>

Note: In such cases, make sure that the request header DOES not include the following:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue>

`*Note:`*

Mandatory parameters are marked in redand placeholder content is marked in <green>.

get

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/appflowpolicy_appflowglobal_binding/name_value<String>

Query-parameters:

filter

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/appflowpolicy_appflowglobal_binding/name_value<String>?**filter=property-name1:property-value1,property-name2:property-value2**

Use this query-parameter to get the filtered set of appflowpolicy_appflowglobal_binding resources configured on Citrix ADC. Filtering can be done on any of the properties of the resource.

pagination

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/appflowpolicy_appflowglobal_binding/name_value<String>?**pagesize=#no;pageno=#no**

Use this query-parameter to get the appflowpolicy_appflowglobal_binding resources in chunks.

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "appflowpolicy_appflowglobal_binding": [ {

"boundto":<String_value>,

"name":<String_value>,

"priority":<Double_value>,

"labelname":<String_value>,

"gotopriorityexpression":<String_value>,

"labeltype":<String_value>,

"activepolicy":<Integer_value>

}]}
```

get (all)

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/appflowpolicy_appflowglobal_binding

Query-parameters:

bulkbindings

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/appflowpolicy_appflowglobal_binding?**bulkbindings=yes**

NITRO allows you to fetch bindings in bulk.

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "appflowpolicy_appflowglobal_binding": [ {

"boundto":<String_value>,

"name":<String_value>,

"priority":<Double_value>,

"labelname":<String_value>,

"gotopriorityexpression":<String_value>,

"labeltype":<String_value>,

"activepolicy":<Integer_value>

}]}
```

count

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/appflowpolicy_appflowglobal_binding/name_value<String>?**count=yes**

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{"appflowpolicy_appflowglobal_binding": [ { "__count": "#no"} ] }
```

appflowpolicy_appflowpolicylabel_binding

Binding object showing the appflowpolicylabel that can be bound to appflowpolicy.

Properties

(click to see Operations)

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
boundto	<String>	Read-write	Location where policy is bound.
name	<String>	Read-write	Name of the policy about which to display detailed information.
priority	<Double>	Read-only	Specifies the priority of the policy.
labelname	<String>	Read-only	Name of the label to invoke if the current policy rule evaluates to TRUE.
gotopriorityexpression	<String>	Read-only	Expression specifying the priority of the next policy which will get

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
			evaluated if the current policy rule evaluates to TRUE.
labeltype	<String>	Read-only	Type of policy label invocation. Possible values = vserver, policylabel
activepolicy	<Integer>	Read-only	Indicates whether policy is bound or not.
__count	<Double>	Read-only	count parameter

Operations

(click to see Properties)

GET| GET (ALL)| COUNT

Some options that you can use for each operations:

- **Getting warnings in response:**NITRO allows you to get warnings in an operation by specifying the "warning" query parameter as "yes". For example, to get warnings while connecting to the Citrix ADC appliance, the URL is as follows:

  http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/login?warning=yes

  If any, the warnings are displayed in the response payload with the HTTP code "209 X-NITRO-WARNING".

- **Authenticated access for individual NITRO operations:**NITRO allows you to logon to the Citrix ADC appliance to perform individual operations. You can use this option instead of creating a NITRO session (using the login object) and then using that session to perform all operations,

  To do this, you must specify the username and password in the request header of the NITRO request as follows:

  X-NITRO-USER:<username>

X-NITRO-PASS:<password>

Note: In such cases, make sure that the request header DOES not include the following:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue>

`*Note:`*

Mandatory parameters are marked in redand placeholder content is marked in <green>.

get

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/appflowpolicy_appflowpolicylabel_binding/name_value<String>

Query-parameters:

filter

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/appflowpolicy_appflowpolicylabel_binding/name_value<String>?**filter=property-name1:property-value1,property-name2:property-value2**

Use this query-parameter to get the filtered set of appflowpolicy_appflowpolicylabel_binding resources configured on Citrix ADC. Filtering can be done on any of the properties of the resource.

pagination

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/appflowpolicy_appflowpolicylabel_binding/name_value<String>?**pagesize=#no;pageno=#no**

Use this query-parameter to get the appflowpolicy_appflowpolicylabel_binding resources in chunks.

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "appflowpolicy_appflowpolicylabel_binding": [ {

"boundto":<String_value>,

"name":<String_value>,

"priority":<Double_value>,

"labelname":<String_value>,

"gotopriorityexpression":<String_value>,

"labeltype":<String_value>,

"activepolicy":<Integer_value>

}]}
```

get (all)

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/appflowpolicy_appflowpolicylabel_binding

Query-parameters:

bulkbindings

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/appflowpolicy_appflowpolicylabel_binding?**bulkbindings=yes**

NITRO allows you to fetch bindings in bulk.

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "appflowpolicy_appflowpolicylabel_binding": [ {

"boundto":<String_value>,

"name":<String_value>,

"priority":<Double_value>,

"labelname":<String_value>,

"gotopriorityexpression":<String_value>,

"labeltype":<String_value>,

"activepolicy":<Integer_value>

}]}
```

count

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/appflowpolicy_appflowpolicylabel_binding/name_value<String>?**count=yes**

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{"appflowpolicy_appflowpolicylabel_binding": [ { "__count": "#no"} ] }
```

appflowpolicy_binding

Binding object which returns the resources bound to appflowpolicy.

Properties

(click to see Operations)

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
name	<String>	Read-write	Name of the policy about which to display detailed information.
appflowpolicy_vpnvserver_binding	<appflowpolicy_vpnvserver_binding[]>	Read-only	vpnvserver that can be bound to appflowpolicy.
appflowpolicy_csvserver_binding	<appflowpolicy_csvserver_binding[]>	Read-only	csvserver that can be bound to appflowpolicy.
appflowpolicy_lbvserver_binding	<appflowpolicy_lbvserver_binding[]>	Read-only	lbvserver that can be bound to appflowpolicy.
appflowpolicy_appflowpolicylabel_binding	<appflowpolicy_appflowpolicylabel_binding[]>	Read-only	appflowpolicylabel that can be bound to appflowpolicy.

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
appflowpolicy_appflowglobal_binding	<appflowpolicy_appflowglobal_binding[]>	Read-only	appflowglobal that can be bound to appflowpolicy.

Operations

(click to see Properties)

GET| GET (ALL)

Some options that you can use for each operations:

- **Getting warnings in response:**NITRO allows you to get warnings in an operation by specifying the "warning" query parameter as "yes". For example, to get warnings while connecting to the Citrix ADC appliance, the URL is as follows:

  http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/login?warning=yes

  If any, the warnings are displayed in the response payload with the HTTP code "209 X-NITRO-WARNING".

- **Authenticated access for individual NITRO operations:**NITRO allows you to logon to the Citrix ADC appliance to perform individual operations. You can use this option instead of creating a NITRO session (using the login object) and then using that session to perform all operations,

  To do this, you must specify the username and password in the request header of the NITRO request as follows:

  X-NITRO-USER:<username>

  X-NITRO-PASS:<password>

  Note: In such cases, make sure that the request header DOES not include the following:

  Cookie:NITRO_AUTH_TOKEN=<tokenvalue>

`*Note:`*

Mandatory parameters are marked in redand placeholder content is marked in <green>.

get

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/appflowpolicy_binding/name_value<String>

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "appflowpolicy_binding": [ {

"name":<String_value>,

"appflowpolicy_appflowglobal_binding":<appflowpolicy_appflowglobal_binding
[]_value>,

"appflowpolicy_appflowpolicylabel_binding":<appflowpolicy_appflowpolicylab
el_binding[]_value>,

"appflowpolicy_csvserver_binding":<appflowpolicy_csvserver_binding[]_value
>,

"appflowpolicy_lbvserver_binding":<appflowpolicy_lbvserver_binding[]_value
>,

"appflowpolicy_vpnvserver_binding":<appflowpolicy_vpnvserver_binding[]_val
ue>

}]}
```

get (all)

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/appflowpolicy_binding

Query-parameters:

bulkbindings

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/appflowpolicy_binding?**bulkbindings=yes**

NITRO allows you to fetch bindings in bulk.

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "appflowpolicy_binding": [ {

"name":<String_value>,

"appflowpolicy_appflowglobal_binding":<appflowpolicy_appflowglobal_binding
[]_value>,

"appflowpolicy_appflowpolicylabel_binding":<appflowpolicy_appflowpolicylab
el_binding[]_value>,

"appflowpolicy_csvserver_binding":<appflowpolicy_csvserver_binding[]_value
>,
```

```
"appflowpolicy_lbvserver_binding":<appflowpolicy_lbvserver_binding[]_value
>,

"appflowpolicy_vpnvserver_binding":<appflowpolicy_vpnvserver_binding[]_val
ue>

}]}
```

appflowpolicy_csvserver_binding

Binding object showing the csvserver that can be bound to appflowpolicy.

Properties

(click to see Operations)

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
boundto	<String>	Read-write	Location where policy is bound.
name	<String>	Read-write	Name of the policy about which to display detailed information.
priority	<Double>	Read-only	Specifies the priority of the policy.
labelname	<String>	Read-only	Name of the label to invoke if the current policy rule evaluates to TRUE.
gotopriorityexpression	<String>	Read-only	Expression specifying the priority of the next policy which will get evaluated if the current policy rule evaluates to TRUE.

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
labeltype	<String>	Read-only	Type of policy label invocation. Possible values = vserver, policylabel
activepolicy	<Integer>	Read-only	Indicates whether policy is bound or not.
__count	<Double>	Read-only	count parameter

Operations

(click to see Properties)

GET| GET (ALL)| COUNT

Some options that you can use for each operations:

- **Getting warnings in response:**NITRO allows you to get warnings in an operation by specifying the "warning" query parameter as "yes". For example, to get warnings while connecting to the Citrix ADC appliance, the URL is as follows:

  http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/login?warning=yes

  If any, the warnings are displayed in the response payload with the HTTP code "209 X-NITRO-WARNING".

- **Authenticated access for individual NITRO operations:**NITRO allows you to logon to the Citrix ADC appliance to perform individual operations. You can use this option instead of creating a NITRO session (using the login object) and then using that session to perform all operations,

  To do this, you must specify the username and password in the request header of the NITRO request as follows:

  X-NITRO-USER:<username>

  X-NITRO-PASS:<password>

  Note: In such cases, make sure that the request header DOES not include the following:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue>

`*Note:`*

Mandatory parameters are marked in redand placeholder content is marked in <green>.

get

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/appflowpolicy_csvserver_binding/name_value<String>

Query-parameters:

filter

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/appflowpolicy_csvserver_binding/name_value<String>?**filter=property-name1:property-value1,property-name2:property-value2**

Use this query-parameter to get the filtered set of appflowpolicy_csvserver_binding resources configured on Citrix ADC. Filtering can be done on any of the properties of the resource.

pagination

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/appflowpolicy_csvserver_binding/name_value<String>?**pagesize=#no;pageno=#no**

Use this query-parameter to get the appflowpolicy_csvserver_binding resources in chunks.

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "appflowpolicy_csvserver_binding": [ {

"boundto":<String_value>,

"name":<String_value>,

"priority":<Double_value>,

"labelname":<String_value>,

"gotopriorityexpression":<String_value>,

"labeltype":<String_value>,

"activepolicy":<Integer_value>

}]}
```

get (all)

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/appflowpolicy_csvserver_binding

Query-parameters:

bulkbindings

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/appflowpolicy_csvserver_binding?**bulkbindings=yes**

NITRO allows you to fetch bindings in bulk.

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "appflowpolicy_csvserver_binding": [ {

"boundto":<String_value>,

"name":<String_value>,

"priority":<Double_value>,

"labelname":<String_value>,

"gotopriorityexpression":<String_value>,

"labeltype":<String_value>,

"activepolicy":<Integer_value>

}]}
```

count

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/appflowpolicy_csvserver_binding/name_value<String>?**count=yes**

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{"appflowpolicy_csvserver_binding": [ { "__count": "#no"} ] }
```

appflowpolicy_lbvserver_binding

Binding object showing the lbvserver that can be bound to appflowpolicy.

Properties

(click to see Operations)

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
boundto	<String>	Read-write	Location where policy is bound.
name	<String>	Read-write	Name of the policy about which to display detailed information.
priority	<Double>	Read-only	Specifies the priority of the policy.
labelname	<String>	Read-only	Name of the label to invoke if the current policy rule evaluates to TRUE.
gotopriorityexpression	<String>	Read-only	Expression specifying the priority of the next policy which will get

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
			evaluated if the current policy rule evaluates to TRUE.
labeltype	<String>	Read-only	Type of policy label invocation. Possible values = vserver, policylabel
activepolicy	<Integer>	Read-only	Indicates whether policy is bound or not.
__count	<Double>	Read-only	count parameter

Operations

(click to see Properties)

GET| GET (ALL)| COUNT

Some options that you can use for each operations:

- **Getting warnings in response:**NITRO allows you to get warnings in an operation by specifying the "warning" query parameter as "yes". For example, to get warnings while connecting to the Citrix ADC appliance, the URL is as follows:

  http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/login?warning=yes

  If any, the warnings are displayed in the response payload with the HTTP code "209 X-NITRO-WARNING".

- **Authenticated access for individual NITRO operations:**NITRO allows you to logon to the Citrix ADC appliance to perform individual operations. You can use this option instead of creating a NITRO session (using the login object) and then using that session to perform all operations,

  To do this, you must specify the username and password in the request header of the NITRO request as follows:

  X-NITRO-USER:<username>

X-NITRO-PASS:<password>

Note: In such cases, make sure that the request header DOES not include the following:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue>

`*Note:`*

Mandatory parameters are marked in redand placeholder content is marked in <green>.

get

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/appflowpolicy_lbvserver_binding/name_value<String>

Query-parameters:

filter

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/appflowpolicy_lbvserver_binding/name_value<String>?**filter=property-name1:property-value1,property-name2:property-value2**

Use this query-parameter to get the filtered set of appflowpolicy_lbvserver_binding resources configured on Citrix ADC. Filtering can be done on any of the properties of the resource.

pagination

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/appflowpolicy_lbvserver_binding/name_value<String>?**pagesize=#no;pageno=#no**

Use this query-parameter to get the appflowpolicy_lbvserver_binding resources in chunks.

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "appflowpolicy_lbvserver_binding": [ {

"boundto":<String_value>,

"name":<String_value>,

"priority":<Double_value>,

"labelname":<String_value>,

"gotopriorityexpression":<String_value>,

"labeltype":<String_value>,

"activepolicy":<Integer_value>

}]}
```

get (all)

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/appflowpolicy_lbvserver_binding

Query-parameters:

bulkbindings

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/appflowpolicy_lbvserver_binding?**bulkbindings=yes**

NITRO allows you to fetch bindings in bulk.

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "appflowpolicy_lbvserver_binding": [ {

"boundto":<String_value>,

"name":<String_value>,

"priority":<Double_value>,

"labelname":<String_value>,

"gotopriorityexpression":<String_value>,

"labeltype":<String_value>,

"activepolicy":<Integer_value>

}]}
```

count

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/appflowpolicy_lbvserver_binding/name_value<String>?**count=yes**

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{"appflowpolicy_lbvserver_binding": [ { "__count": "#no"} ] }
```

appflowpolicy_vpnvserver_binding

Binding object showing the vpnvserver that can be bound to appflowpolicy.

Properties

(click to see Operations)

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
boundto	<String>	Read-write	Location where policy is bound.
name	<String>	Read-write	Name of the policy about which to display detailed information.
priority	<Double>	Read-only	Specifies the priority of the policy.
labelname	<String>	Read-only	Name of the label to invoke if the current policy rule evaluates to TRUE.
gotopriorityexpression	<String>	Read-only	Expression specifying the priority of the next policy which will get

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
			evaluated if the current policy rule evaluates to TRUE.
labeltype	<String>	Read-only	Type of policy label invocation. Possible values = vserver, policylabel
activepolicy	<Integer>	Read-only	Indicates whether policy is bound or not.
__count	<Double>	Read-only	count parameter

Operations

(click to see Properties)

GET| GET (ALL)| COUNT

Some options that you can use for each operations:

- **Getting warnings in response:**NITRO allows you to get warnings in an operation by specifying the "warning" query parameter as "yes". For example, to get warnings while connecting to the Citrix ADC appliance, the URL is as follows:

  http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/login?warning=yes

  If any, the warnings are displayed in the response payload with the HTTP code "209 X-NITRO-WARNING".

- **Authenticated access for individual NITRO operations:**NITRO allows you to logon to the Citrix ADC appliance to perform individual operations. You can use this option instead of creating a NITRO session (using the login object) and then using that session to perform all operations,

  To do this, you must specify the username and password in the request header of the NITRO request as follows:

  X-NITRO-USER:<username>

X-NITRO-PASS:<password>

Note: In such cases, make sure that the request header DOES not include the following:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue>

`*Note:`*

Mandatory parameters are marked in redand placeholder content is marked in <green>.

get

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/appflowpolicy_vpnvserver_binding/name_value<String>

Query-parameters:

filter

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/appflowpolicy_vpnvserver_binding/name_value<String>?**filter=property-name1:property-value1,property-name2:property-value2**

Use this query-parameter to get the filtered set of appflowpolicy_vpnvserver_binding resources configured on Citrix ADC. Filtering can be done on any of the properties of the resource.

pagination

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/appflowpolicy_vpnvserver_binding/name_value<String>?**pagesize=#no;pageno=#no**

Use this query-parameter to get the appflowpolicy_vpnvserver_binding resources in chunks.

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "appflowpolicy_vpnvserver_binding": [ {

"boundto":<String_value>,

"name":<String_value>,

"priority":<Double_value>,

"labelname":<String_value>,

"gotopriorityexpression":<String_value>,

"labeltype":<String_value>,

"activepolicy":<Integer_value>

}]}
```

get (all)

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/appflowpolicy_vpnvserver_binding

Query-parameters:

bulkbindings

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/appflowpolicy_vpnvserver_binding?**bulkbindings=yes**

NITRO allows you to fetch bindings in bulk.

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "appflowpolicy_vpnvserver_binding": [ {

"boundto":<String_value>,

"name":<String_value>,

"priority":<Double_value>,

"labelname":<String_value>,

"gotopriorityexpression":<String_value>,

"labeltype":<String_value>,

"activepolicy":<Integer_value>

}]}
```

count

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/appflowpolicy_vpnvserver_binding/name_value<String>?**count=yes**

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{"appflowpolicy_vpnvserver_binding": [ { "__count": "#no"} ] }
```

appflowaction_analyticsprofile_binding

Binding object showing the analyticsprofile that can be bound to appflowaction.

Properties

(click to see Operations)

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
name	<String>	Read-write	Name for the action. Must begin with an ASCII alphabetic or underscore (_) character, and must contain only ASCII alphanumeric, underscore, hash (#), period (.), space, colon (:), at (@), equals (=), and hyphen (-) characters. The following requirement applies only to the Citrix ADC CLI: If the name includes one or more spaces, enclose the name in

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
			double or single quotation marks (for example, "my appflow action" or 'my appflow action').
analyticsprofile	<String>	Read-write	Analytics profile to be bound to the appflow action.
__count	<Double>	Read-write	count parameter

Operations

(click to see Properties)

ADD:| DELETE:| GET| GET (ALL)| COUNT

Some options that you can use for each operations:

- **Getting warnings in response:**NITRO allows you to get warnings in an operation by specifying the "warning" query parameter as "yes". For example, to get warnings while connecting to the Citrix ADC appliance, the URL is as follows:

  http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/login?warning=yes

  If any, the warnings are displayed in the response payload with the HTTP code "209 X-NITRO-WARNING".

- **Authenticated access for individual NITRO operations:**NITRO allows you to logon to the Citrix ADC appliance to perform individual operations. You can use this option instead of creating a NITRO session (using the login object) and then using that session to perform all operations,

  To do this, you must specify the username and password in the request header of the NITRO request as follows:

  X-NITRO-USER:<username>

  X-NITRO-PASS:<password>

  Note: In such cases, make sure that the request header DOES not include the following:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue>

`*Note:` *

Mandatory parameters are marked in redand placeholder content is marked in <green>.

add

URL:http://<Citrix-ADC-IP-address(NSIP)/nitro/v1/config/appflowaction_analyticsprofile_binding

HTTP Method:PUT

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Content-Type:application/json

Request Payload:

```
{

"appflowaction_analyticsprofile_binding":{

<b>"name":<String_value>,

</b>"analyticsprofile":<String_value>

}}
```

Response:

HTTP Status Code on Success: 201 Created HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error

delete

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/appflowaction_analyticsprofile_binding/name_value<String>

Query-parameters:

args

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/appflowaction_analyticsprofile_binding/name_value<String>?**args=analyticsprofile:<String_value>**

HTTP Method:DELETE

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue>

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error

get

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/appflowaction_analyticsprofile_binding/name_value<String>

Query-parameters:

filter

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/appflowaction_analyticsprofile_binding/name_value<String>?**filter=property-name1:property-value1,property-name2:property-value2**

Use this query-parameter to get the filtered set of appflowaction_analyticsprofile_binding resources configured on Citrix ADC. Filtering can be done on any of the properties of the resource.

pagination

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/appflowaction_analyticsprofile_binding/name_value<String>?**pagesize=#no;pageno=#no**

Use this query-parameter to get the appflowaction_analyticsprofile_binding resources in chunks.

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "appflowaction_analyticsprofile_binding": [ {

"name":<String_value>,

"analyticsprofile":<String_value>

}]}
```

get (all)

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/appflowaction_analyticsprofile_binding

Query-parameters:

bulkbindings

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/appflowaction_analyticsprofile_binding?**bulkbindings=yes**

NITRO allows you to fetch bindings in bulk.

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "appflowaction_analyticsprofile_binding": [ {

"name":<String_value>,

"analyticsprofile":<String_value>

}]}
```

count

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/appflowaction_analyticsprofile_binding/name_value<String>?**count=yes**

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{"appflowaction_analyticsprofile_binding": [ { "__count": "#no"} ] }
```

appflowaction_binding

Binding object which returns the resources bound to appflowaction.

Properties

(click to see Operations)

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
name	<String>	Read-write	Name of the action about which to display information.
appflowaction_analyticsprofile_binding	<appflowaction_analyticsprofile_binding[]>	Read-only	analyticsprofile that can be bound to appflowaction.

Operations

(click to see Properties)

GET| GET (ALL)

Some options that you can use for each operations:

- **Getting warnings in response:**NITRO allows you to get warnings in an operation by specifying the "warning" query parameter as "yes". For example, to get warnings while connecting to the Citrix ADC appliance, the URL is as follows:

  http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/login?warning=yes

  If any, the warnings are displayed in the response payload with the HTTP code "209 X-NITRO-WARNING".

- **Authenticated access for individual NITRO operations:**NITRO allows you to logon to the Citrix ADC appliance to perform individual operations. You can use this option instead of creating a NITRO session (using the login object) and then using that session to perform all operations,

  To do this, you must specify the username and password in the request header of the NITRO request as follows:

  X-NITRO-USER:<username>

X-NITRO-PASS:<password>

Note: In such cases, make sure that the request header DOES not include the following:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue>

`*Note:`*

Mandatory parameters are marked in redand placeholder content is marked in <green>.

get

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/appflowaction_binding/name_value<String>

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "appflowaction_binding": [ {

"name":<String_value>,

"appflowaction_analyticsprofile_binding":<appflowaction_analyticsprofile_binding[]_value>

}]}
```

get (all)

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/appflowaction_binding

Query-parameters:

bulkbindings

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/appflowaction_binding?**bulkbindings=yes**

NITRO allows you to fetch bindings in bulk.

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "appflowaction_binding": [ {

"name":<String_value>,

"appflowaction_analyticsprofile_binding":<appflowaction_analyticsprofile_b
inding[]_value>

}]}
```

Application Firewall

Application Firewall

Application Firewall configuration.

NAME	DESCRIPTION
appfwarchive	Configuration for archive
appfwconfidfield	Configuration for configured confidential form fields
appfwcustomsettings	Configuration for application firewall custom settings XML configuration
appfwfieldtype	Configuration for application firewall form field type
appfwglobal_appfwpolicy_binding	Binding object showing the appfwpolicy that can be bound to appfwglobal
appfwglobal_auditnslogpolicy_binding	Binding object showing the auditnslogpolicy that can be bound to appfwglobal
appfwglobal_auditsyslogpolicy_binding	Binding object showing the auditsyslogpolicy that can be bound to appfwglobal
appfwglobal_binding	Binding object showing the resources that can be bound to appfwglobal
appfwhtmlerrorpage	Configuration for HTML error page
appfwjsoncontenttype	Configuration for JSON content type
appfwlearningdata	Configuration for learning data
appfwlearningsettings	Configuration for learning settings
appfwpolicy	Configuration for application firewall policy
appfwpolicy_appfwglobal_binding	Binding object showing the appfwglobal that can be bound to appfwpolicy

NAME	DESCRIPTION
appfwpolicy_appfwpolicylabel_binding	Binding object showing the appfwpolicylabel that can be bound to appfwpolicy
appfwpolicy_binding	Binding object showing the resources that can be bound to appfwpolicy
appfwpolicy_csvserver_binding	Binding object showing the csvserver that can be bound to appfwpolicy
appfwpolicy_lbvserver_binding	Binding object showing the lbvserver that can be bound to appfwpolicy
appfwpolicylabel	Configuration for application firewall policy label
appfwpolicylabel_appfwpolicy_binding	Binding object showing the appfwpolicy that can be bound to appfwpolicylabel
appfwpolicylabel_binding	Binding object showing the resources that can be bound to appfwpolicylabel
appfwpolicylabel_policybinding_binding	Binding object showing the policybinding that can be bound to appfwpolicylabel
appfwprofile	Configuration for application firewall profile
appfwprofile_binding	Binding object showing the resources that can be bound to appfwprofile
appfwprofile_contenttype_binding	Binding object showing the contenttype that can be bound to appfwprofile
appfwprofile_cookieconsistency_binding	Binding object showing the cookieconsistency that can be bound to appfwprofile
appfwprofile_creditcardnumber_binding	Binding object showing the creditcardnumber that can be bound to appfwprofile
appfwprofile_crosssitescripting_binding	Binding object showing the crosssitescripting that can be bound to appfwprofile

NAME	DESCRIPTION
appfwprofile_csrftag_binding	Binding object showing the csrftag that can be bound to appfwprofile
appfwprofile_denyurl_binding	Binding object showing the denyurl that can be bound to appfwprofile
appfwprofile_excluderescontenttype_binding	Binding object showing the excluderescontenttype that can be bound to appfwprofile
appfwprofile_fieldconsistency_binding	Binding object showing the fieldconsistency that can be bound to appfwprofile
appfwprofile_fieldformat_binding	Binding object showing the fieldformat that can be bound to appfwprofile
appfwprofile_logexpression_binding	Binding object showing the logexpression that can be bound to appfwprofile
appfwprofile_safeobject_binding	Binding object showing the safeobject that can be bound to appfwprofile
appfwprofile_sqlinjection_binding	Binding object showing the sqlinjection that can be bound to appfwprofile
appfwprofile_starturl_binding	Binding object showing the starturl that can be bound to appfwprofile
appfwprofile_trustedlearningclients_binding	Binding object showing the trustedlearningclients that can be bound to appfwprofile
appfwprofile_xmlattachmenturl_binding	Binding object showing the xmlattachmenturl that can be bound to appfwprofile
appfwprofile_xmldosurl_binding	Binding object showing the xmldosurl that can be bound to appfwprofile
appfwprofile_xmlsqlinjection_binding	Binding object showing the xmlsqlinjection that can be bound to appfwprofile

NAME	DESCRIPTION
appfwprofile_xmlvalidationurl_binding	Binding object showing the xmlvalidationurl that can be bound to appfwprofile
appfwprofile_xmlwsiurl_binding	Binding object showing the xmlwsiurl that can be bound to appfwprofile
appfwprofile_xmlxss_binding	Binding object showing the xmlxss that can be bound to appfwprofile
appfwsettings	Configuration for AS settings
appfwsignatures	Configuration for application firewall signatures XML configuration
appfwtransactionrecords	Configuration for Application firewall transaction record
appfwwsdl	Configuration for WSDL file
appfwxmlcontenttype	Configuration for XML Content type
appfwxmlerrorpage	Configuration for xml error page
appfwxmlschema	Configuration for XML schema

appfwarchive

Configuration for archive resource.

Properties

(click to see Operations)

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
name	<String>	Read-write	Name of tar archive. Minimum length = 1 Maximum length = 31

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
target	<String>	Read-write	Path to the file to be exported. Minimum length = 1 Maximum length = 2047
src	<String>	Read-write	Indicates the source of the tar archive file as a URL of the form <protocol>://<host>[:<port>][/<path>] <protocol> is http or https. <host> is the DNS name or IP address of the http or https server. <port> is the port number of the server. If omitted, the default port for http or https will be used. <path> is the path of the file on the server.

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
			Import will fail if an https server requires client certificate authentication. . Minimum length = 1 Maximum length = 2047
comment	<String>	Read-write	Comments associated with this archive. Maximum length = 128
response	<String>	Read-only	.

Operations

(click to see Properties)

EXPORT| IMPORT| DELETE| GET (ALL)

Some options that you can use for each operations:

- **Getting warnings in response:**NITRO allows you to get warnings in an operation by specifying the "warning" query parameter as "yes". For example, to get warnings while connecting to the Citrix ADC appliance, the URL is as follows:

  http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/login?warning=yes

  If any, the warnings are displayed in the response payload with the HTTP code "209 X-NITRO-WARNING".

- **Authenticated access for individual NITRO operations:**NITRO allows you to logon to the Citrix ADC appliance to perform individual operations. You can use this option instead of

creating a NITRO session (using the login object) and then using that session to perform all operations,

To do this, you must specify the username and password in the request header of the NITRO request as follows:

X-NITRO-USER:<username>

X-NITRO-PASS:<password>

Note: In such cases, make sure that the request header DOES not include the following:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue>

`*Note:`*

Mandatory parameters are marked in redand placeholder content is marked in <green>.

export

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/appfwarchive?**action=export**

HTTP Method:POST

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Content-Type:application/json

Request Payload:

```
{"appfwarchive":{

<b>"name":<String_value>,

</b><b>"target":<String_value>

</b>}}
```

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error

Import

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/appfwarchive?**action=Import**

HTTP Method:POST

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Content-Type:application/json

Request Payload:

```
{"appfwarchive":{

<b>"src":<String_value>,

</b><b>"name":<String_value>,

</b>"comment":<String_value>

}}
```

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error

delete

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/appfwarchive/name_value<String>

HTTP Method:DELETE

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue>

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error

get (all)

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/appfwarchive

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "appfwarchive": [ {

"response":<String_value>

}]}
```

appfwconfidfield

Configuration for configured confidential form fields resource.

Properties

(click to see Operations)

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
fieldname	<String>	Read-write	Name of the form field to designate as confidential. Minimum length = 1
url	<String>	Read-write	URL of the web page that contains the web form. Minimum length = 1
isregex	<String>	Read-write	Method of specifying the form field name. Available settings function as follows: * REGEX. Form field is a regular expression. * NOTREGEX. Form field is a literal string. Default value: NOTREGEX Possible values = REGEX, NOTREGEX
comment	<String>	Read-write	Any comments to preserve information about the form field designation.
state	<String>	Read-write	Enable or disable the confidential field designation. Default value: ENABLED

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
			Possible values = ENABLED, DISABLED
__count	<Double>	Read-only	count parameter

Operations

(click to see Properties)

ADD| DELETE| UPDATE| UNSET| GET (ALL)| COUNT

Some options that you can use for each operations:

- **Getting warnings in response:**NITRO allows you to get warnings in an operation by specifying the "warning" query parameter as "yes". For example, to get warnings while connecting to the Citrix ADC appliance, the URL is as follows:

  http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/login?warning=yes

  If any, the warnings are displayed in the response payload with the HTTP code "209 X-NITRO-WARNING".

- **Authenticated access for individual NITRO operations:**NITRO allows you to logon to the Citrix ADC appliance to perform individual operations. You can use this option instead of creating a NITRO session (using the login object) and then using that session to perform all operations,

  To do this, you must specify the username and password in the request header of the NITRO request as follows:

  X-NITRO-USER:<username>

  X-NITRO-PASS:<password>

  Note: In such cases, make sure that the request header DOES not include the following:

  Cookie:NITRO_AUTH_TOKEN=<tokenvalue>

`*Note:`*

Mandatory parameters are marked in redand placeholder content is marked in <green>.

add

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/appfwconfidfield

HTTP Method:POST

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Content-Type:application/json

Request Payload:

```
{"appfwconfidfield":{

<b>"fieldname":<String_value>,

</b><b>"url":<String_value>,

</b>"isregex":<String_value>,

"comment":<String_value>,

"state":<String_value>

}}
```

Response:

HTTP Status Code on Success: 201 Created HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error

delete

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/appfwconfidfield/fieldname_value<String>

HTTP Method:DELETE

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue>

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error

update

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/appfwconfidfield

HTTP Method:PUT

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Content-Type:application/json

Request Payload:

```
{"appfwconfidfield":{

<b>"fieldname":<String_value>,

</b><b>"url":<String_value>,

</b>"comment":<String_value>,

"isregex":<String_value>,

"state":<String_value>

}}
```

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error

unset

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/appfwconfidfield?**action=unset**

HTTP Method:POST

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Content-Type:application/json

Request Payload:

```
{"appfwconfidfield":{

<b>"fieldname":<String_value>,

</b><b>"url":<String_value>,

</b>"comment":true,

"isregex":true,

"state":true

}}
```

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error

get (all)

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/appfwconfidfield

Query-parameters:

attrs

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/appfwconfidfield?**attrs=property-name1,property-name2**

Use this query parameter to specify the resource details that you want to retrieve.

filter

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/appfwconfidfield?**filter=property-name1:property-val1,property-name2:property-val2**

Use this query-parameter to get the filtered set of appfwconfidfield resources configured on Citrix ADC. Filtering can be done on any of the properties of the resource.

view

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/appfwconfidfield?**view=summary**

Note: By default, the retrieved results are displayed in detail view (?view=detail).

pagination

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/appfwconfidfield?**pagesize=#no;pageno=#no**

Use this query-parameter to get the appfwconfidfield resources in chunks.

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "appfwconfidfield": [ {

"fieldname":<String_value>,

"url":<String_value>,
```

```
"isregex":<String_value>,

"comment":<String_value>,

"state":<String_value>

}]}
```

count

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/appfwconfidfield?**count=yes**

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "appfwconfidfield": [ { "__count": "#no"} ] }
```

appfwcustomsettings

Configuration for application firewall custom settings XML configuration resource.

Properties

(click to see Operations)

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
name	<String>	Read-write	. Minimum length = 1 Maximum length = 31
target	<String>	Read-write	. Minimum length = 1 Maximum length = 2047

Operations

(click to see Properties)

EXPORT

Some options that you can use for each operations:

- **Getting warnings in response:**NITRO allows you to get warnings in an operation by specifying the "warning" query parameter as "yes". For example, to get warnings while connecting to the Citrix ADC appliance, the URL is as follows:

  http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/login?warning=yes

  If any, the warnings are displayed in the response payload with the HTTP code "209 X-NITRO-WARNING".

- **Authenticated access for individual NITRO operations:**NITRO allows you to logon to the Citrix ADC appliance to perform individual operations. You can use this option instead of creating a NITRO session (using the login object) and then using that session to perform all operations,

  To do this, you must specify the username and password in the request header of the NITRO request as follows:

  X-NITRO-USER:<username>

  X-NITRO-PASS:<password>

  Note: In such cases, make sure that the request header DOES not include the following:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue>

`*Note:` *

Mandatory parameters are marked in redand placeholder content is marked in <green>.

export

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/appfwcustomsettings?**action=export**

HTTP Method:POST

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Content-Type:application/json

Request Payload:

```
{"appfwcustomsettings":{

<b>"name":<String_value>,

</b><b>"target":<String_value>

</b>}}
```

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error

appfwfieldtype

Configuration for application firewall form field type resource.

Properties

(click to see Operations)

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
name	<String>	Read-write	Name for the field type. Must begin with a letter, number, or the underscore character (_), and must contain only letters, numbers, and the hyphen (-), period (.) pound (#), space ( ), at (@), equals (=), colon (:), and underscore characters. Cannot be changed after the field type is added. The following requirement applies only to the Citrix ADC CLI: If the name includes one or more spaces, enclose the name in double or single quotation marks (for example, "my field type" or 'my field type'). Minimum length = 1
regex	<String>	Read-write	PCRE - format regular expression defining the characters and

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
			length allowed for this field type. Minimum length = 1
priority	<Double>	Read-write	Positive integer specifying the priority of the field type. A lower number specifies a higher priority. Field types are checked in the order of their priority numbers. Minimum value = 0 Maximum value = 64000
comment	<String>	Read-write	Comment describing the type of field that this field type is intended to match.
nocharmaps	<Boolean>	Read-write	will not show internal field types added as part of FieldFormat learn rules deployment.
__count	<Double>	Read-only	count parameter

Operations

(click to see Properties)

ADD| DELETE| UPDATE| GET (ALL)| GET| COUNT

Some options that you can use for each operations:

- **Getting warnings in response:**NITRO allows you to get warnings in an operation by specifying the "warning" query parameter as "yes". For example, to get warnings while connecting to the Citrix ADC appliance, the URL is as follows:

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/login?warning=yes

If any, the warnings are displayed in the response payload with the HTTP code "209 X-NITRO-WARNING".

- **Authenticated access for individual NITRO operations:**NITRO allows you to logon to the Citrix ADC appliance to perform individual operations. You can use this option instead of creating a NITRO session (using the login object) and then using that session to perform all operations,

  To do this, you must specify the username and password in the request header of the NITRO request as follows:

  X-NITRO-USER:<username>

  X-NITRO-PASS:<password>

  Note: In such cases, make sure that the request header DOES not include the following:

  Cookie:NITRO_AUTH_TOKEN=<tokenvalue>

`*Note:`*

Mandatory parameters are marked in redand placeholder content is marked in <green>.

add

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/appfwfieldtype

HTTP Method:POST

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Content-Type:application/json

Request Payload:

```
{"appfwfieldtype":{

<b>"name":<String_value>,

</b><b>"regex":<String_value>,

</b><b>"priority":<Double_value>,
```

```
</b>"comment":<String_value>

}}
```

Response:

HTTP Status Code on Success: 201 Created HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error

delete

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/appfwfieldtype/name_value<String>

HTTP Method:DELETE

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue>

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error

update

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/appfwfieldtype

HTTP Method:PUT

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Content-Type:application/json

Request Payload:

```
{"appfwfieldtype":{

<b>"name":<String_value>,
```

```
</b><b>"regex":<String_value>,

</b><b>"priority":<Double_value>,

</b>"comment":<String_value>

}}
```

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error

get (all)

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/appfwfieldtype

Query-parameters:

args

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/appfwfieldtype?**args=name:<String_value>,nocharmaps:<Boolean_value>**

Use this query-parameter to get appfwfieldtype resources based on additional properties.

attrs

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/appfwfieldtype?**attrs=property-name1,property-name2**

Use this query parameter to specify the resource details that you want to retrieve.

filter

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/appfwfieldtype?**filter=property-name1:property-val1,property-name2:property-val2**

Use this query-parameter to get the filtered set of appfwfieldtype resources configured on Citrix ADC. Filtering can be done on any of the properties of the resource.

view

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/appfwfieldtype?**view=summary**

Note: By default, the retrieved results are displayed in detail view (?view=detail).

pagination

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/appfwfieldtype?**pagesize=#no;pageno=#no**

Use this query-parameter to get the appfwfieldtype resources in chunks.

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "appfwfieldtype": [ {

name:<String_value>,nocharmaps:<Boolean_value>"regex":<String_value>,

"priority":<Double_value>,

"comment":<String_value>

}]}
```

get

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/appfwfieldtype/name_value<String>

Query-parameters:

attrs

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/appfwfieldtype/name_value<String>?**attrs=property-name1,property-name2**

Use this query parameter to specify the resource details that you want to retrieve.

view

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/appfwfieldtype/name_value<String>?**view=summary**

Note: By default, the retrieved results are displayed in detail view (?view=detail).

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "appfwfieldtype": [ {

name:<String_value>,nocharmaps:<Boolean_value>"regex":<String_value>,

"priority":<Double_value>,

"comment":<String_value>

}]}
```

count

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/appfwfieldtype?**count=yes**

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "appfwfieldtype": [ { "__count": "#no"} ] }
```

appfwglobal_appfwpolicy_binding

Binding object showing the appfwpolicy that can be bound to appfwglobal.

Properties

(click to see Operations)

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
priority	<Double>	Read-write	The priority of the policy.
globalbindtype	<String>	Read-write	. Default value: SYSTEM_GLOBAL Possible values = SYSTEM_GLOBAL,

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
			VPN_GLOBAL, RNAT_GLOBAL
policyname	<String>	Read-write	Name of the policy.
labelname	<String>	Read-write	Name of the policy label to invoke if the current policy evaluates to TRUE, the invoke parameter is set, and Label Type is set to Policy Label.
state	<String>	Read-write	Enable or disable the binding to activate or deactivate the policy. This is applicable to classic policies only. Possible values = ENABLED, DISABLED
gotopriorityexpression	<String>	Read-write	Expression specifying the priority of the next policy which will get evaluated if the current policy rule evaluates to TRUE.
type	<String>	Read-write	Bind point to which to policy is bound. Possible values = REQ_OVERRIDE, REQ_DEFAULT, NONE
invoke	<Boolean>	Read-write	If the current policy evaluates to TRUE,

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
			terminate evaluation of policies bound to the current policy label, and then forward the request to the specified virtual server or evaluate the specified policy label.
labeltype	<String>	Read-write	Type of policy label invocation. Possible values = reqvserver, policylabel
flowtype	<Double>	Read-only	flowtype of the bound application firewall policy.
numpol	<Double>	Read-only	The number of policies bound to the bindpoint.
policytype	<String>	Read-only	. Possible values = Classic Policy, Advanced Policy
__count	<Double>	Read-only	count parameter

Operations

(click to see Properties)

ADD:| DELETE:| GET| GET (ALL)| COUNT

Some options that you can use for each operations:

- **Getting warnings in response:**NITRO allows you to get warnings in an operation by specifying the "warning" query parameter as "yes". For example, to get warnings while connecting to the Citrix ADC appliance, the URL is as follows:

  http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/login?warning=yes

  If any, the warnings are displayed in the response payload with the HTTP code "209 X-NITRO-WARNING".

- **Authenticated access for individual NITRO operations:**NITRO allows you to logon to the Citrix ADC appliance to perform individual operations. You can use this option instead of creating a NITRO session (using the login object) and then using that session to perform all operations,

  To do this, you must specify the username and password in the request header of the NITRO request as follows:

  X-NITRO-USER:<username>

  X-NITRO-PASS:<password>

  Note: In such cases, make sure that the request header DOES not include the following:

  Cookie:NITRO_AUTH_TOKEN=<tokenvalue>

`*Note:`*

Mandatory parameters are marked in redand placeholder content is marked in <green>.

add

URL:http://<Citrix-ADC-IP-address(NSIP)/nitro/v1/config/appfwglobal_appfwpolicy_binding

HTTP Method:PUT

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Content-Type:application/json

Request Payload:

```
{
```

```
"appfwglobal_appfwpolicy_binding":{

<b>"policyname":<String_value>,

</b><b>"priority":<Double_value>,

</b>"state":<String_value>,

"gotopriorityexpression":<String_value>,

"type":<String_value>,

"invoke":<Boolean_value>,

"labeltype":<String_value>,

"labelname":<String_value>

}}
```

Response:

HTTP Status Code on Success: 201 Created HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error

delete

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/appfwglobal_appfwpolicy_binding

Query-parameters:

args

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/appfwglobal_appfwpolicy_binding?**args=policyname:<String_value>,type:<String_value>,priority:<Double_value>**

HTTP Method:DELETE

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue>

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error

get

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/appfwglobal_appfwpolicy_binding

Query-parameters:

filter

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/appfwglobal_appfwpolicy_binding?**filter=property-name1:property-value1,property-name2:property-value2**

Use this query-parameter to get the filtered set of appfwglobal_appfwpolicy_binding resources configured on Citrix ADC. Filtering can be done on any of the properties of the resource.

pagination

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/appfwglobal_appfwpolicy_binding?**pagesize=#no;pageno=#no**

Use this query-parameter to get the appfwglobal_appfwpolicy_binding resources in chunks.

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "appfwglobal_appfwpolicy_binding": [ {

"priority":<Double_value>,

"globalbindtype":<String_value>,

"policyname":<String_value>,

"labelname":<String_value>,

"state":<String_value>,

"gotopriorityexpression":<String_value>,

"type":<String_value>,

"invoke":<Boolean_value>,

"labeltype":<String_value>,

"flowtype":<Double_value>,

"numpol":<Double_value>,

"policytype":<String_value>

}]}
```

get (all)

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/appfwglobal_appfwpolicy_binding

Query-parameters:

bulkbindings

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/appfwglobal_appfwpolicy_binding?**bulkbindings=yes**

NITRO allows you to fetch bindings in bulk.

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "appfwglobal_appfwpolicy_binding": [ {

"priority":<Double_value>,

"globalbindtype":<String_value>,

"policyname":<String_value>,

"labelname":<String_value>,

"state":<String_value>,

"gotopriorityexpression":<String_value>,

"type":<String_value>,

"invoke":<Boolean_value>,

"labeltype":<String_value>,

"flowtype":<Double_value>,

"numpol":<Double_value>,

"policytype":<String_value>
```

```
}]}
```

count

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/appfwglobal_appfwpolicy_binding?**count=yes**

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "appfwglobal_appfwpolicy_binding": [ { "__count": "#no"} ] }
```

appfwglobal_auditnslogpolicy_binding

Binding object showing the auditnslogpolicy that can be bound to appfwglobal.

Properties

(click to see Operations)

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
priority	<Double>	Read-write	The priority of the policy.
policyname	<String>	Read-write	Name of the policy.

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
labelname	<String>	Read-write	Name of the policy label to invoke if the current policy evaluates to TRUE, the invoke parameter is set, and Label Type is set to Policy Label.
state	<String>	Read-write	Enable or disable the binding to activate or deactivate the policy. This is applicable to classic policies only. Possible values = ENABLED, DISABLED
gotopriorityexpression	<String>	Read-write	Expression or other value specifying the next policy to evaluate if the current policy evaluates to TRUE. Specify one of the following values: * NEXT - Evaluate the policy with the next higher priority number. * END - End policy evaluation. * USE_INVOCATION_RESULT - Applicable if this policy invokes another policy label. If the final goto in the invoked policy label has a value of END, the evaluation stops. If the final goto is

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
			anything other than END, the current policy label performs a NEXT. * An expression that evaluates to a number. If you specify an expression, the number to which it evaluates determines the next policy to evaluate, as follows: * If the expression evaluates to a higher numbered priority, the policy with that priority is evaluated next. * If the expression evaluates to the priority of the current policy, the policy with the next higher numbered priority is evaluated next. * If the expression evaluates to a number that is larger than the largest numbered priority, policy evaluation ends. An UNDEF event is triggered if: * The expression is invalid. * The expression evaluates to a priority number that is smaller than the current policy's

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
			priority number. * The expression evaluates to a priority number that is between the current policy's priority number (say, 30) and the highest priority number (say, 100), but does not match any configured priority number (for example, the expression evaluates to the number 85). This example assumes that the priority number increments by 10 for every successive policy, and therefore a priority number of 85 does not exist in the policy label.
invoke	<Boolean>	Read-write	If the current policy evaluates to TRUE, terminate evaluation of policies bound to the current policy label, and then forward the request to the specified virtual server or evaluate the specified policy label.
type	<String>	Read-write	Bind point to which to policy is bound.

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
			Possible values = REQ_OVERRIDE, REQ_DEFAULT, NONE
labeltype	<String>	Read-write	Type of policy label to invoke if the current policy evaluates to TRUE and the invoke parameter is set. Available settings function as follows: * reqvserver. Invoke the unnamed policy label associated with the specified request virtual server. * policylabel. Invoke the specified user-defined policy label. Possible values = reqvserver, policylabel
policytype	<String>	Read-only	. Possible values = Classic Policy, Advanced Policy
__count	<Double>	Read-only	count parameter

Operations

(click to see Properties)

ADD:| DELETE:| GET| GET (ALL)| COUNT

Some options that you can use for each operations:

- **Getting warnings in response:**NITRO allows you to get warnings in an operation by specifying the "warning" query parameter as "yes". For example, to get warnings while connecting to the Citrix ADC appliance, the URL is as follows:

  http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/login?warning=yes

  If any, the warnings are displayed in the response payload with the HTTP code "209 X-NITRO-WARNING".

- **Authenticated access for individual NITRO operations:**NITRO allows you to logon to the Citrix ADC appliance to perform individual operations. You can use this option instead of creating a NITRO session (using the login object) and then using that session to perform all operations,

  To do this, you must specify the username and password in the request header of the NITRO request as follows:

  X-NITRO-USER:<username>

  X-NITRO-PASS:<password>

  Note: In such cases, make sure that the request header DOES not include the following:

  Cookie:NITRO_AUTH_TOKEN=<tokenvalue>

`*Note:`*

Mandatory parameters are marked in redand placeholder content is marked in <green>.

add

URL:http://<Citrix-ADC-IP-address(NSIP)/nitro/v1/config/appfwglobal_auditnslogpolicy_binding

HTTP Method:PUT

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Content-Type:application/json

Request Payload:

```
{
```

```
"appfwglobal_auditnslogpolicy_binding":{

<b>"policyname":<String_value>,

</b><b>"priority":<Double_value>,

</b>"state":<String_value>,

"gotopriorityexpression":<String_value>,

"type":<String_value>,

"invoke":<Boolean_value>,

"labeltype":<String_value>,

"labelname":<String_value>

}}
```

Response:

HTTP Status Code on Success: 201 Created HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error

delete

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/appfwglobal_auditnslogpolicy_binding

Query-parameters:

args

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/appfwglobal_auditnslogpolicy_binding?**args=policyname:<String_value>,type:<String_value>,priority:<Double_value>**

HTTP Method:DELETE

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue>

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error

get

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/appfwglobal_auditnslogpolicy_binding

Query-parameters:

filter

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/appfwglobal_auditnslogpolicy_binding?**filter=property-name1:property-value1,property-name2:property-value2**

Use this query-parameter to get the filtered set of appfwglobal_auditnslogpolicy_binding resources configured on Citrix ADC. Filtering can be done on any of the properties of the resource.

pagination

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/appfwglobal_auditnslogpolicy_binding?**pagesize=#no;pageno=#no**

Use this query-parameter to get the appfwglobal_auditnslogpolicy_binding resources in chunks.

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "appfwglobal_auditnslogpolicy_binding": [ {

"priority":<Double_value>,

"policyname":<String_value>,

"labelname":<String_value>,

"state":<String_value>,

"gotopriorityexpression":<String_value>,

"invoke":<Boolean_value>,

"type":<String_value>,

"labeltype":<String_value>,

"policytype":<String_value>

}]}
```

get (all)

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/appfwglobal_auditnslogpolicy_binding

Query-parameters:

bulkbindings

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/appfwglobal_auditnslogpolicy_binding?**bulkbindings=yes**

NITRO allows you to fetch bindings in bulk.

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "appfwglobal_auditnslogpolicy_binding": [ {

"priority":<Double_value>,

"policyname":<String_value>,

"labelname":<String_value>,

"state":<String_value>,

"gotopriorityexpression":<String_value>,

"invoke":<Boolean_value>,

"type":<String_value>,

"labeltype":<String_value>,

"policytype":<String_value>

}]}
```

count

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/appfwglobal_auditnslogpolicy_binding?**count=yes**

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "appfwglobal_auditnslogpolicy_binding": [ { "__count": "#no"} ] }
```

appfwglobal_auditsyslogpolicy_binding

Binding object showing the auditsyslogpolicy that can be bound to appfwglobal.

Properties

(click to see Operations)

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
priority	<Double>	Read-write	The priority of the policy.
policyname	<String>	Read-write	Name of the policy.
labelname	<String>	Read-write	Name of the policy label to invoke if the current policy evaluates to TRUE, the invoke parameter is set, and Label Type is set to Policy Label.
state	<String>	Read-write	Enable or disable the binding to activate or deactivate the policy.

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
			This is applicable to classic policies only. Possible values = ENABLED, DISABLED
gotopriorityexpression	<String>	Read-write	Expression or other value specifying the next policy to evaluate if the current policy evaluates to TRUE. Specify one of the following values: * NEXT - Evaluate the policy with the next higher priority number. * END - End policy evaluation. * USE_INVOCATION_RESULT - Applicable if this policy invokes another policy label. If the final goto in the invoked policy label has a value of END, the evaluation stops. If the final goto is anything other than END, the current policy label performs a NEXT. * An expression that evaluates to a number. If you specify an expression, the number to which it evaluates determines the next policy to evaluate, as

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
			follows: * If the expression evaluates to a higher numbered priority, the policy with that priority is evaluated next. * If the expression evaluates to the priority of the current policy, the policy with the next higher numbered priority is evaluated next. * If the expression evaluates to a number that is larger than the largest numbered priority, policy evaluation ends. An UNDEF event is triggered if: * The expression is invalid. * The expression evaluates to a priority number that is smaller than the current policy's priority number. * The expression evaluates to a priority number that is between the current policy's priority number (say, 30) and the highest priority number (say, 100), but does not match any configured priority number (for

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
			example, the expression evaluates to the number 85). This example assumes that the priority number increments by 10 for every successive policy, and therefore a priority number of 85 does not exist in the policy label.
invoke	<Boolean>	Read-write	If the current policy evaluates to TRUE, terminate evaluation of policies bound to the current policy label, and then forward the request to the specified virtual server or evaluate the specified policy label.
type	<String>	Read-write	Bind point to which to policy is bound. Possible values = REQ_OVERRIDE, REQ_DEFAULT, NONE
labeltype	<String>	Read-write	Type of policy label to invoke if the current policy evaluates to TRUE and the invoke parameter is set. Available settings function as follows: * reqvserver. Invoke

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
			the unnamed policy label associated with the specified request virtual server. * policylabel. Invoke the specified user-defined policy label. Possible values = reqvserver, policylabel
policytype	<String>	Read-only	. Possible values = Classic Policy, Advanced Policy
__count	<Double>	Read-only	count parameter

Operations

(click to see Properties)

ADD:| DELETE:| GET| GET (ALL)| COUNT

Some options that you can use for each operations:

- **Getting warnings in response:**NITRO allows you to get warnings in an operation by specifying the "warning" query parameter as "yes". For example, to get warnings while connecting to the Citrix ADC appliance, the URL is as follows:

  http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/login?warning=yes

  If any, the warnings are displayed in the response payload with the HTTP code "209 X-NITRO-WARNING".
- **Authenticated access for individual NITRO operations:**NITRO allows you to logon to the Citrix ADC appliance to perform individual operations. You can use this option instead of creating a NITRO session (using the login object) and then using that session to perform all operations,

To do this, you must specify the username and password in the request header of the NITRO request as follows:

X-NITRO-USER:<username>

X-NITRO-PASS:<password>

Note: In such cases, make sure that the request header DOES not include the following:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue>

`*Note:`*

Mandatory parameters are marked in redand placeholder content is marked in <green>.

add

URL:http://<Citrix-ADC-IP-address(NSIP)/nitro/v1/config/appfwglobal_auditsyslogpolicy_binding

HTTP Method:PUT

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Content-Type:application/json

Request Payload:

```
{

"appfwglobal_auditsyslogpolicy_binding":{

<b>"policyname":<String_value>,

</b><b>"priority":<Double_value>,

</b>"state":<String_value>,

"gotopriorityexpression":<String_value>,

"type":<String_value>,

"invoke":<Boolean_value>,
```

```
"labeltype":<String_value>,

"labelname":<String_value>

}}
```

Response:

HTTP Status Code on Success: 201 Created HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error

delete

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/appfwglobal_auditsyslogpolicy_binding

Query-parameters:

args

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/appfwglobal_auditsyslogpolicy_binding?**args=policyname:<String_value>,type:<String_value>,priority:<Double_value>**

HTTP Method:DELETE

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue>

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error

get

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/appfwglobal_auditsyslogpolicy_binding

Query-parameters:

filter

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/appfwglobal_auditsyslogpolicy_binding?**filter=property-name1:property-value1,property-name2:property-value2**

Use this query-parameter to get the filtered set of appfwglobal_auditsyslogpolicy_binding resources configured on Citrix ADC. Filtering can be done on any of the properties of the resource.

pagination

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/appfwglobal_auditsyslogpolicy_binding?**pagesize=#no;pageno=#no**

Use this query-parameter to get the appfwglobal_auditsyslogpolicy_binding resources in chunks.

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "appfwglobal_auditsyslogpolicy_binding": [ {

"priority":<Double_value>,

"policyname":<String_value>,

"labelname":<String_value>,

"state":<String_value>,

"gotopriorityexpression":<String_value>,

"invoke":<Boolean_value>,
```

```
"type":<String_value>,

"labeltype":<String_value>,

"policytype":<String_value>

}]}
```

get (all)

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/appfwglobal_auditsyslogpolicy_binding

Query-parameters:

bulkbindings

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/appfwglobal_auditsyslogpolicy_binding?**bulkbindings=yes**

NITRO allows you to fetch bindings in bulk.

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "appfwglobal_auditsyslogpolicy_binding": [ {

"priority":<Double_value>,
```

```
"policyname":<String_value>,

"labelname":<String_value>,

"state":<String_value>,

"gotopriorityexpression":<String_value>,

"invoke":<Boolean_value>,

"type":<String_value>,

"labeltype":<String_value>,

"policytype":<String_value>

}]}
```

count

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/appfwglobal_auditsyslogpolicy_binding?**count=yes**

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "appfwglobal_auditsyslogpolicy_binding": [ { "__count": "#no"} ] }
```

appfwglobal_binding

Binding object which returns the resources bound to appfwglobal.

Properties

(click to see Operations)

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
appfwglobal_auditnslogpolicy_binding	<appfwglobal_auditnslogpolicy_binding[]>	Read-only	auditnslogpolicy that can be bound to appfwglobal.
appfwglobal_auditsyslogpolicy_binding	<appfwglobal_auditsyslogpolicy_binding[]>	Read-only	auditsyslogpolicy that can be bound to appfwglobal.
appfwglobal_appfwpolicy_binding	<appfwglobal_appfwpolicy_binding[]>	Read-only	appfwpolicy that can be bound to appfwglobal.

Operations

(click to see Properties)

GET| GET (ALL)

Some options that you can use for each operations:

- **Getting warnings in response:**NITRO allows you to get warnings in an operation by specifying the "warning" query parameter as "yes". For example, to get warnings while connecting to the Citrix ADC appliance, the URL is as follows:

  http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/login?warning=yes

  If any, the warnings are displayed in the response payload with the HTTP code "209 X-NITRO-WARNING".
- **Authenticated access for individual NITRO operations:**NITRO allows you to logon to the Citrix ADC appliance to perform individual operations. You can use this option instead of

creating a NITRO session (using the login object) and then using that session to perform all operations,

To do this, you must specify the username and password in the request header of the NITRO request as follows:

X-NITRO-USER:<username>

X-NITRO-PASS:<password>

Note: In such cases, make sure that the request header DOES not include the following:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue>

`*Note:`*

Mandatory parameters are marked in redand placeholder content is marked in <green>.

get

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/appfwglobal_binding

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "appfwglobal_binding": [ {

"appfwglobal_appfwpolicy_binding":<appfwglobal_appfwpolicy_binding[]_value
>,

"appfwglobal_auditsyslogpolicy_binding":<appfwglobal_auditsyslogpolicy_bin
```

```
ding[]_value>,

"appfwglobal_auditnslogpolicy_binding":<appfwglobal_auditnslogpolicy_bindi
ng[]_value>

}]}
```

get (all)

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/appfwglobal_binding

Query-parameters:

bulkbindings

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/appfwglobal_binding?**bulkbindings=yes**

NITRO allows you to fetch bindings in bulk.

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "appfwglobal_binding": [ {

"appfwglobal_appfwpolicy_binding":<appfwglobal_appfwpolicy_binding[]_value
>,

"appfwglobal_auditsyslogpolicy_binding":<appfwglobal_auditsyslogpolicy_bin
ding[]_value>,
```

```
"appfwglobal_auditnslogpolicy_binding":<appfwglobal_auditnslogpolicy_binding[]_value>

}]}
```

appfwhtmlerrorpage

Configuration for HTML error page resource.

Properties

(click to see Operations)

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
name	<String>	Read-write	Name of the XML error object to remove. Minimum length = 1 Maximum length = 31
src	<String>	Read-write	URL (protocol, host, path, and name) for the location at which to store the imported HTML error object. NOTE: The import fails if the object to be imported is on an HTTPS server that requires client certificate authentication for access. Minimum length = 1

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
			Maximum length = 2047
comment	<String>	Read-write	Any comments to preserve information about the HTML error object. Maximum length = 128
overwrite	<Boolean>	Read-write	Overwrite any existing HTML error object of the same name.
response	<String>	Read-only	.

Operations

(click to see Properties)

DELETE| IMPORT| CHANGE| GET (ALL)| GET

Some options that you can use for each operations:

- **Getting warnings in response:**NITRO allows you to get warnings in an operation by specifying the "warning" query parameter as "yes". For example, to get warnings while connecting to the Citrix ADC appliance, the URL is as follows:

  http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/login?warning=yes

  If any, the warnings are displayed in the response payload with the HTTP code "209 X-NITRO-WARNING".

- **Authenticated access for individual NITRO operations:**NITRO allows you to logon to the Citrix ADC appliance to perform individual operations. You can use this option instead of creating a NITRO session (using the login object) and then using that session to perform all operations,

  To do this, you must specify the username and password in the request header of the NITRO request as follows:

X-NITRO-USER:<username>

X-NITRO-PASS:<password>

Note: In such cases, make sure that the request header DOES not include the following:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue>

`*Note:`*

Mandatory parameters are marked in redand placeholder content is marked in <green>.

delete

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/appfwhtmlerrorpage/name_value<String>

HTTP Method:DELETE

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue>

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error

Import

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/appfwhtmlerrorpage?**action=Import**

HTTP Method:POST

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Content-Type:application/json

Request Payload:

```
{"appfwhtmlerrorpage":{

<b>"src":<String_value>,
```

```
</b><b>"name":<String_value>,

</b>"comment":<String_value>,

"overwrite":<Boolean_value>

}}
```

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error

change

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/appfwhtmlerrorpage?**action=update**

HTTP Method:POST

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Content-Type:application/json

Request Payload:

```
{"appfwhtmlerrorpage":{

<b>"name":<String_value>

</b>}}
```

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error

get (all)

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/appfwhtmlerrorpage

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "appfwhtmlerrorpage": [ {

"name":<String_value>,

"response":<String_value>,

"src":<String_value>

}]}
```

get

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/appfwhtmlerrorpage/name_value<String>

Query-parameters:

attrs

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/appfwhtmlerrorpage/name_value<String>?**attrs=property-name1,property-name2**

Use this query parameter to specify the resource details that you want to retrieve.

view

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/appfwhtmlerrorpage/name_value<String>?**view=summary**

Note: By default, the retrieved results are displayed in detail view (?view=detail).

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "appfwhtmlerrorpage": [ {

"name":<String_value>,

"response":<String_value>,

"src":<String_value>

}]}
```

appfwjsoncontenttype

Configuration for JSON content type resource.

Properties

(click to see Operations)

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
jsoncontenttypevalue	<String>	Read-write	Content type to be classified as JSON. Minimum length = 1
isregex	<String>	Read-write	Is json content type a regular expression?. Default value: NOTREGEX Possible values = REGEX, NOTREGEX
__count	<Double>	Read-only	count parameter

Operations

(click to see Properties)

ADD| DELETE| GET (ALL)| GET| COUNT

Some options that you can use for each operations:

- **Getting warnings in response:**NITRO allows you to get warnings in an operation by specifying the "warning" query parameter as "yes". For example, to get warnings while connecting to the Citrix ADC appliance, the URL is as follows:

  http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/login?warning=yes

  If any, the warnings are displayed in the response payload with the HTTP code "209 X-NITRO-WARNING".
- **Authenticated access for individual NITRO operations:**NITRO allows you to logon to the Citrix ADC appliance to perform individual operations. You can use this option instead of creating a NITRO session (using the login object) and then using that session to perform all operations,

To do this, you must specify the username and password in the request header of the NITRO request as follows:

X-NITRO-USER:<username>

X-NITRO-PASS:<password>

Note: In such cases, make sure that the request header DOES not include the following:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue>

`*Note:`*

Mandatory parameters are marked in redand placeholder content is marked in <green>.

add

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/appfwjsoncontenttype

HTTP Method:POST

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Content-Type:application/json

Request Payload:

```
{"appfwjsoncontenttype":{

<b>"jsoncontenttypevalue":<String_value>,

</b>"isregex":<String_value>

}}
```

Response:

HTTP Status Code on Success: 201 Created HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error

delete

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/appfwjsoncontenttype/jsoncontenttypevalue_value<String>

HTTP Method:DELETE

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue>

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error

get (all)

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/appfwjsoncontenttype

Query-parameters:

attrs

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/appfwjsoncontenttype?**attrs=property-name1,property-name2**

Use this query parameter to specify the resource details that you want to retrieve.

filter

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/appfwjsoncontenttype?**filter=property-name1:property-val1,property-name2:property-val2**

Use this query-parameter to get the filtered set of appfwjsoncontenttype resources configured on Citrix ADC. Filtering can be done on any of the properties of the resource.

view

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/appfwjsoncontenttype?**view=summary**

Note: By default, the retrieved results are displayed in detail view (?view=detail).

pagination

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/appfwjsoncontenttype?**pagesize=#no;pageno=#no**

Use this query-parameter to get the appfwjsoncontenttype resources in chunks.

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "appfwjsoncontenttype": [ {

"jsoncontenttypevalue":<String_value>,

"isregex":<String_value>

}]}
```

get

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/appfwjsoncontenttype/jsoncontenttypevalue_value<String>

Query-parameters:

attrs

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/appfwjsoncontenttype/jsoncontenttypevalue_value<String>?**attrs=property-name1,property-name2**

Use this query parameter to specify the resource details that you want to retrieve.

view

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/appfwjsoncontenttype/jsoncontenttypevalue_value<String>?**view=summary**

Note: By default, the retrieved results are displayed in detail view (?view=detail).

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "appfwjsoncontenttype": [ {

"jsoncontenttypevalue":<String_value>,

"isregex":<String_value>

}]}
```

count

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/appfwjsoncontenttype?**count=yes**

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "appfwjsoncontenttype": [ { "__count": "#no"} ] }
```

appfwlearningdata

Configuration for learning data resource.

Properties

(click to see Operations)

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
profilename	<String>	Read-write	Name of the profile.
starturl	<String>	Read-write	Start URL configuration. Minimum length = 1
cookieconsistency	<String>	Read-write	Cookie Name. Minimum length = 1
fieldconsistency	<String>	Read-write	Form field name. Minimum length = 1
formactionurl_ffc	<String>	Read-write	Form action URL.
contenttype	<String>	Read-write	Content Type Name. Minimum length = 1
crosssitescripting	<String>	Read-write	Cross-site scripting.

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
			Minimum length = 1
formactionurl_xss	<String>	Read-write	Form action URL. Minimum length = 1
as_scan_location_xss	<String>	Read-write	Location of cross-site scripting exception - form field, header or cookie. Possible values = FORMFIELD, HEADER, COOKIE
as_value_type_xss	<String>	Read-write	XSS value type. (Tag | Attribute | Pattern). Possible values = Tag, Attribute, Pattern
as_value_expr_xss	<String>	Read-write	XSS value expressions consistituting expressions for Tag, Attribute or Pattern.
sqlinjection	<String>	Read-write	Form field name. Minimum length = 1
formactionurl_sql	<String>	Read-write	Form action URL. Minimum length = 1
as_scan_location_sql	<String>	Read-write	Location of sql injection exception - form field, header or cookie. Possible values = FORMFIELD, HEADER, COOKIE

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
as_value_type_sql	<String>	Read-write	SQL value type. Keyword, SpecialString or Wildchar. Possible values = Keyword, SpecialString, Wildchar
as_value_expr_sql	<String>	Read-write	SQL value expressions consistituting expressions for Keyword, SpecialString or Wildchar.
fieldformat	<String>	Read-write	Field format name. Minimum length = 1
formactionurl_ff	<String>	Read-write	Form action URL. Minimum length = 1
csrftag	<String>	Read-write	CSRF Form Action URL. Minimum length = 1
csrfformoriginurl	<String>	Read-write	CSRF Form Origin URL. Minimum length = 1
creditcardnumber	<String>	Read-write	The object expression that is to be excluded from safe commerce check. Minimum length = 1
creditcardnumberurl	<String>	Read-write	The url for which the list of credit card

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
			numbers are needed to be bypassed from inspection. Minimum length = 1
xmldoscheck	<String>	Read-write	XML Denial of Service check, one of MaxAttributes MaxAttributeNameLength MaxAttributeValueLength MaxElementNameLength MaxFileSize MinFileSize MaxCDATALength MaxElements MaxElementDepth MaxElementChildren NumDTDs NumProcessingInstructions NumExternalEntities

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
			MaxEntityExpansions MaxEntityExpansionDepth MaxNamespaces MaxNamespaceUriLength MaxSOAPArraySize MaxSOAPArrayRank . Minimum length = 1
xmlwsicheck	<String>	Read-write	Web Services Interoperability Rule ID. Minimum length = 1
xmlattachmentcheck	<String>	Read-write	XML Attachment Content-Type. Minimum length = 1
totalxmlrequests	<Boolean>	Read-write	Total XML requests.
securitycheck	<String>	Read-write	Name of the security check. Possible values = startURL, cookieConsistency, fieldConsistency, crossSiteScripting, SQLInjection, fieldFormat, CSRFtag,

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
			XMLDoSCheck, XMLWSICheck, XMLAttachmentCheck, TotalXMLRequests, creditCardNumber, ContentType
target	<String>	Read-write	Target filename for data to be exported. Minimum length = 1 Maximum length = 127
url	<String>	Read-only	Learnt url.
name	<String>	Read-only	Learnt field name.
fieldtype	<String>	Read-only	Learnt field type.
fieldformatminlength	<Double>	Read-only	The minimum allowed length for data in this form field.
fieldformatmaxlength	<Double>	Read-only	The maximum allowed length for data in this form field.
fieldformatcharmappcre	<String>	Read-only	Form field value allowed character map.
value_type	<String>	Read-only	Learnt field value type.
value	<String>	Read-only	Learnt field value.

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
hits	<Double>	Read-only	Learnt entity hit count.
data	<String>	Read-only	Learned data.
__count	<Double>	Read-only	count parameter

Operations

(click to see Properties)

DELETE| RESET| EXPORT| GET (ALL)| COUNT

Some options that you can use for each operations:

- **Getting warnings in response:**NITRO allows you to get warnings in an operation by specifying the "warning" query parameter as "yes". For example, to get warnings while connecting to the Citrix ADC appliance, the URL is as follows:

  http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/login?warning=yes

  If any, the warnings are displayed in the response payload with the HTTP code "209 X-NITRO-WARNING".

- **Authenticated access for individual NITRO operations:**NITRO allows you to logon to the Citrix ADC appliance to perform individual operations. You can use this option instead of creating a NITRO session (using the login object) and then using that session to perform all operations,

  To do this, you must specify the username and password in the request header of the NITRO request as follows:

  X-NITRO-USER:<username>

  X-NITRO-PASS:<password>

  Note: In such cases, make sure that the request header DOES not include the following:

  Cookie:NITRO_AUTH_TOKEN=<tokenvalue>

`*Note:` *

Mandatory parameters are marked in redand placeholder content is marked in <green>.

delete

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/appfwlearningdata

HTTP Method:DELETE

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue>

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error

reset

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/appfwlearningdata?**action=reset**

HTTP Method:POST

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Content-Type:application/json

Request Payload:

```
{"appfwlearningdata":{

}}
```

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error

export

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/appfwlearningdata?**action=export**

HTTP Method:POST

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Content-Type:application/json

Request Payload:

```
{"appfwlearningdata":{

<b>"profilename":<String_value>,

</b><b>"securitycheck":<String_value>,

</b>"target":<String_value>

}}
```

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error

get (all)

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/appfwlearningdata

Query-parameters:

args

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/appfwlearningdata?**args=profilename:<String_value>,securitycheck:<String_value>,**

Use this query-parameter to get appfwlearningdata resources based on additional properties.

attrs

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/appfwlearningdata?**attrs=property-name1,property-name2**

Use this query parameter to specify the resource details that you want to retrieve.

filter

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/appfwlearningdata?**filter=property-name1:property-val1,property-name2:property-val2**

Use this query-parameter to get the filtered set of appfwlearningdata resources configured on Citrix ADC. Filtering can be done on any of the properties of the resource.

view

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/appfwlearningdata?**view=summary**

Note: By default, the retrieved results are displayed in detail view (?view=detail).

pagination

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/appfwlearningdata?**pagesize=#no;pageno=#no**

Use this query-parameter to get the appfwlearningdata resources in chunks.

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "appfwlearningdata": [ {

<b>profilename:<String_value>,</b><b>securitycheck:<String_value>,</
b>"url":<String_value>,

"name":<String_value>,
```

```
"fieldtype":<String_value>,

"fieldformatminlength":<Double_value>,

"fieldformatmaxlength":<Double_value>,

"fieldformatcharmappcre":<String_value>,

"value_type":<String_value>,

"value":<String_value>,

"hits":<Double_value>,

"data":<String_value>

}]}
```

count

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/appfwlearningdata?**args=profilename:<String_value>,securitycheck:<String_value>,;count=yes**

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "appfwlearningdata": [ { "__count": "#no"} ] }
```

appfwlearningsettings

Configuration for learning settings resource.

Properties

(click to see Operations)

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
profilename	<String>	Read-write	Name of the profile. Minimum length = 1
starturlminthreshold	<Double>	Read-write	Minimum number of application firewall sessions that the learning engine must observe to learn start URLs. Default value: 1 Minimum value = 1
starturlpercentthreshold	<Double>	Read-write	Minimum percentage of application firewall sessions that must contain a particular start URL pattern for the learning engine to learn that start URL. Default value: 0 Minimum value = 0 Maximum value = 100
cookieconsistencyminthreshold	<Double>	Read-write	Minimum number of application firewall sessions that the

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
			learning engine must observe to learn cookies. Default value: 1 Minimum value = 1
cookieconsistencypercentthreshold	<Double>	Read-write	Minimum percentage of application firewall sessions that must contain a particular cookie pattern for the learning engine to learn that cookie. Default value: 0 Minimum value = 0 Maximum value = 100
csrftagminthreshold	<Double>	Read-write	Minimum number of application firewall sessions that the learning engine must observe to learn cross-site request forgery (CSRF) tags. Default value: 1 Minimum value = 1
csrftagpercentthreshold	<Double>	Read-write	Minimum percentage of application firewall sessions that must contain a particular CSRF tag for the learning engine to learn that CSRF tag. Default value: 0

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
			Minimum value = 0 Maximum value = 100
fieldconsistencymintreshold	<Double>	Read-write	Minimum number of application firewall sessions that the learning engine must observe to learn field consistency information. Default value: 1 Minimum value = 1
fieldconsistencypercentthreshold	<Double>	Read-write	Minimum percentage of application firewall sessions that must contain a particular field consistency pattern for the learning engine to learn that field consistency pattern. Default value: 0 Minimum value = 0 Maximum value = 100
crosssitescriptingminthreshold	<Double>	Read-write	Minimum number of application firewall sessions that the learning engine must observe to learn HTML cross-site scripting patterns. Default value: 1

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
			Minimum value = 1
crosssitescriptingpercentthreshold	<Double>	Read-write	Minimum percentage of application firewall sessions that must contain a particular cross-site scripting pattern for the learning engine to learn that cross-site scripting pattern. Default value: 0 Minimum value = 0 Maximum value = 100
sqlinjectionminthreshold	<Double>	Read-write	Minimum number of application firewall sessions that the learning engine must observe to learn HTML SQL injection patterns. Default value: 1 Minimum value = 1
sqlinjectionpercentthreshold	<Double>	Read-write	Minimum percentage of application firewall sessions that must contain a particular HTML SQL injection pattern for the learning engine to learn that HTML SQL injection pattern. Default value: 0

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
			Minimum value = 0 Maximum value = 100
fieldformatminthreshold	<Double>	Read-write	Minimum number of application firewall sessions that the learning engine must observe to learn field formats. Default value: 1 Minimum value = 1
fieldformatpercentthreshold	<Double>	Read-write	Minimum percentage of application firewall sessions that must contain a particular web form field pattern for the learning engine to recommend a field format for that form field. Default value: 0 Minimum value = 0 Maximum value = 100
creditcardnumberminthreshold	<Double>	Read-write	Minimum threshold to learn Credit Card information. Default value: 1 Minimum value = 1
creditcardnumberpercentthreshold	<Double>	Read-write	Minimum threshold in percent to learn

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
			Credit Card information. Default value: 0 Minimum value = 0 Maximum value = 100
contenttypeminthreshold	<Double>	Read-write	Minimum threshold to learn Content Type information. Default value: 1 Minimum value = 1
contenttypepercentthreshold	<Double>	Read-write	Minimum threshold in percent to learn Content Type information. Default value: 0 Minimum value = 0 Maximum value = 100
xmlwsiminthreshold	<Double>	Read-write	Minimum number of application firewall sessions that the learning engine must observe to learn web services interoperability (WSI) information. Default value: 1 Minimum value = 1
xmlwsipercentthreshold	<Double>	Read-write	Minimum percentage of application firewall sessions that must

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
			contain a particular pattern for the learning engine to learn a web services interoperability (WSI) pattern. Default value: 0 Minimum value = 0 Maximum value = 100
xmlattachmentminthreshold	<Double>	Read-write	Minimum number of application firewall sessions that the learning engine must observe to learn XML attachment patterns. Default value: 1 Minimum value = 1
xmlattachmentpercentthreshold	<Double>	Read-write	Minimum percentage of application firewall sessions that must contain a particular XML attachment pattern for the learning engine to learn that XML attachment pattern. Default value: 0 Minimum value = 0 Maximum value = 100
__count	<Double>	Read-only	count parameter

Operations

(click to see Properties)

UPDATE| UNSET| GET (ALL)| GET| COUNT

Some options that you can use for each operations:

- **Getting warnings in response:**NITRO allows you to get warnings in an operation by specifying the "warning" query parameter as "yes". For example, to get warnings while connecting to the Citrix ADC appliance, the URL is as follows:

  http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/login?warning=yes

  If any, the warnings are displayed in the response payload with the HTTP code "209 X-NITRO-WARNING".

- **Authenticated access for individual NITRO operations:**NITRO allows you to logon to the Citrix ADC appliance to perform individual operations. You can use this option instead of creating a NITRO session (using the login object) and then using that session to perform all operations,

  To do this, you must specify the username and password in the request header of the NITRO request as follows:

  X-NITRO-USER:<username>

  X-NITRO-PASS:<password>

  Note: In such cases, make sure that the request header DOES not include the following:

  Cookie:NITRO_AUTH_TOKEN=<tokenvalue>

`*Note:`*

Mandatory parameters are marked in redand placeholder content is marked in <green>.

update

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/appfwlearningsettings

HTTP Method:PUT

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Content-Type:application/json

Request Payload:

```
{"appfwlearningsettings":{

<b>"profilename":<String_value>,

</b>"starturlminthreshold":<Double_value>,

"starturlpercentthreshold":<Double_value>,

"cookieconsistencyminthreshold":<Double_value>,

"cookieconsistencypercentthreshold":<Double_value>,

"csrftagminthreshold":<Double_value>,

"csrftagpercentthreshold":<Double_value>,

"fieldconsistencyminthreshold":<Double_value>,

"fieldconsistencypercentthreshold":<Double_value>,

"crosssitescriptingminthreshold":<Double_value>,

"crosssitescriptingpercentthreshold":<Double_value>,

"sqlinjectionminthreshold":<Double_value>,

"sqlinjectionpercentthreshold":<Double_value>,

"fieldformatminthreshold":<Double_value>,

"fieldformatpercentthreshold":<Double_value>,

"creditcardnumberminthreshold":<Double_value>,

"creditcardnumberpercentthreshold":<Double_value>,

"contenttypeminthreshold":<Double_value>,

"contenttypepercentthreshold":<Double_value>,
```

```
"xmlwsiminthreshold":<Double_value>,

"xmlwsipercentthreshold":<Double_value>,

"xmlattachmentminthreshold":<Double_value>,

"xmlattachmentpercentthreshold":<Double_value>

}}
```

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error

unset

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/appfwlearningsettings?**action=unset**

HTTP Method:POST

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Content-Type:application/json

Request Payload:

```
{"appfwlearningsettings":{

<b>"profilename":<String_value>,

</b>"starturlminthreshold":true,

"starturlpercentthreshold":true,

"cookieconsistencyminthreshold":true,

"cookieconsistencypercentthreshold":true,
```

```
"csrftagminthreshold":true,

"csrftagpercentthreshold":true,

"fieldconsistencyminthreshold":true,

"fieldconsistencypercentthreshold":true,

"crosssitescriptingminthreshold":true,

"crosssitescriptingpercentthreshold":true,

"sqlinjectionminthreshold":true,

"sqlinjectionpercentthreshold":true,

"fieldformatminthreshold":true,

"fieldformatpercentthreshold":true,

"creditcardnumberminthreshold":true,

"creditcardnumberpercentthreshold":true,

"contenttypeminthreshold":true,

"contenttypepercentthreshold":true,

"xmlwsiminthreshold":true,

"xmlwsipercentthreshold":true,

"xmlattachmentminthreshold":true,

"xmlattachmentpercentthreshold":true

}}
```


Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error

get (all)

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/appfwlearningsettings

Query-parameters:

attrs

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/appfwlearningsettings?**attrs=property-name1,property-name2**

Use this query parameter to specify the resource details that you want to retrieve.

filter

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/appfwlearningsettings?**filter=property-name1:property-val1,property-name2:property-val2**

Use this query-parameter to get the filtered set of appfwlearningsettings resources configured on Citrix ADC. Filtering can be done on any of the properties of the resource.

view

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/appfwlearningsettings?**view=summary**

Note: By default, the retrieved results are displayed in detail view (?view=detail).

pagination

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/appfwlearningsettings?**pagesize=#no;pageno=#no**

Use this query-parameter to get the appfwlearningsettings resources in chunks.

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "appfwlearningsettings": [ {

"profilename":<String_value>,

"starturlminthreshold":<Double_value>,

"starturlpercentthreshold":<Double_value>,

"cookieconsistencyminthreshold":<Double_value>,

"cookieconsistencypercentthreshold":<Double_value>,

"csrftagminthreshold":<Double_value>,

"csrftagpercentthreshold":<Double_value>,

"fieldconsistencyminthreshold":<Double_value>,

"fieldconsistencypercentthreshold":<Double_value>,

"crosssitescriptingminthreshold":<Double_value>,

"crosssitescriptingpercentthreshold":<Double_value>,

"sqlinjectionminthreshold":<Double_value>,

"sqlinjectionpercentthreshold":<Double_value>,

"fieldformatminthreshold":<Double_value>,

"fieldformatpercentthreshold":<Double_value>,

"creditcardnumberminthreshold":<Double_value>,
```

```
"creditcardnumberpercentthreshold":<Double_value>,

"contenttypeminthreshold":<Double_value>,

"contenttypepercentthreshold":<Double_value>,

"xmlwsiminthreshold":<Double_value>,

"xmlwsipercentthreshold":<Double_value>,

"xmlattachmentminthreshold":<Double_value>,

"xmlattachmentpercentthreshold":<Double_value>

}]}
```

get

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/appfwlearningsettings/profilename_value<String>

Query-parameters:

attrs

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/appfwlearningsettings/profilename_value<String>?**attrs=property-name1,property-name2**

Use this query parameter to specify the resource details that you want to retrieve.

view

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/appfwlearningsettings/profilename_value<String>?**view=summary**

Note: By default, the retrieved results are displayed in detail view (?view=detail).

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "appfwlearningsettings": [ {

"profilename":<String_value>,

"starturlminthreshold":<Double_value>,

"starturlpercentthreshold":<Double_value>,

"cookieconsistencyminthreshold":<Double_value>,

"cookieconsistencypercentthreshold":<Double_value>,

"csrftagminthreshold":<Double_value>,

"csrftagpercentthreshold":<Double_value>,

"fieldconsistencyminthreshold":<Double_value>,

"fieldconsistencypercentthreshold":<Double_value>,

"crosssitescriptingminthreshold":<Double_value>,

"crosssitescriptingpercentthreshold":<Double_value>,

"sqlinjectionminthreshold":<Double_value>,

"sqlinjectionpercentthreshold":<Double_value>,

"fieldformatminthreshold":<Double_value>,

"fieldformatpercentthreshold":<Double_value>,
```

```
"creditcardnumberminthreshold":<Double_value>,

"creditcardnumberpercentthreshold":<Double_value>,

"contenttypeminthreshold":<Double_value>,

"contenttypepercentthreshold":<Double_value>,

"xmlwsiminthreshold":<Double_value>,

"xmlwsipercentthreshold":<Double_value>,

"xmlattachmentminthreshold":<Double_value>,

"xmlattachmentpercentthreshold":<Double_value>

}]}
```

count

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/appfwlearningsettings?**count=yes**

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "appfwlearningsettings": [ { "__count": "#no"} ] }
```

appfwpolicy

Configuration for application firewall policy resource.

Properties

(click to see Operations)

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
name	<String>	Read-write	Name for the policy. Must begin with a letter, number, or the underscore character \(_\), and must contain only letters, numbers, and the hyphen \(-\), period \(.\) pound \(#\), space \(\), at (@), equals \(=\), colon \(:\), and underscore characters. Can be changed after the policy is created. The following requirement applies only to the Citrix ADC CLI: If the name includes one or more spaces, enclose the name in double or single quotation marks \(for

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
			example, "my policy" or 'my policy'\). Minimum length = 1
rule	<String>	Read-write	Name of the Citrix ADC named rule, or a Citrix ADC expression, that the policy uses to determine whether to filter the connection through the application firewall with the designated profile.
profilename	<String>	Read-write	Name of the application firewall profile to use if the policy matches. Minimum length = 1
comment	<String>	Read-write	Any comments to preserve information about the policy for later reference.
logaction	<String>	Read-write	Where to log information for connections that match this policy.
newname	<String>	Read-write	New name for the policy. Must begin with a letter, number, or the underscore character (_), and must contain only letters, numbers, and

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
			the hyphen (-), period (.) pound (#), space ( ), at (@), equals (=), colon (:), and underscore characters. The following requirement applies only to the Citrix ADC CLI: If the name includes one or more spaces, enclose the name in double or single quotation marks (for example, "my policy" or 'my policy'). Minimum length = 1
hits	<Double>	Read-only	Number of hits.
undefhits	<Double>	Read-only	Number of Undef hits.
policytype	<String>	Read-only	. Possible values = Classic Policy, Advanced Policy
__count	<Double>	Read-only	count parameter

Operations

(click to see Properties)

ADD| DELETE| UPDATE| UNSET| RENAME| GET (ALL)| GET| COUNT

Some options that you can use for each operations:

- **Getting warnings in response:**NITRO allows you to get warnings in an operation by specifying the "warning" query parameter as "yes". For example, to get warnings while connecting to the Citrix ADC appliance, the URL is as follows:

  http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/login?warning=yes

  If any, the warnings are displayed in the response payload with the HTTP code "209 X-NITRO-WARNING".

- **Authenticated access for individual NITRO operations:**NITRO allows you to logon to the Citrix ADC appliance to perform individual operations. You can use this option instead of creating a NITRO session (using the login object) and then using that session to perform all operations,

  To do this, you must specify the username and password in the request header of the NITRO request as follows:

  X-NITRO-USER:<username>

  X-NITRO-PASS:<password>

  Note: In such cases, make sure that the request header DOES not include the following:

  Cookie:NITRO_AUTH_TOKEN=<tokenvalue>

`*Note:`*

Mandatory parameters are marked in redand placeholder content is marked in <green>.

add

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/appfwpolicy

HTTP Method:POST

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Content-Type:application/json

Request Payload:

```
{"appfwpolicy":{

<b>"name":<String_value>,

</b><b>"rule":<String_value>,

</b><b>"profilename":<String_value>,

</b>"comment":<String_value>,

"logaction":<String_value>

}}
```

Response:

HTTP Status Code on Success: 201 Created HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error

delete

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/appfwpolicy/name_value<String>

HTTP Method:DELETE

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue>

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error

update

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/appfwpolicy

HTTP Method:PUT

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Content-Type:application/json

Request Payload:

```
{"appfwpolicy":{

<b>"name":<String_value>,

</b>"rule":<String_value>,

"profilename":<String_value>,

"comment":<String_value>,

"logaction":<String_value>

}}
```

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error

unset

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/appfwpolicy?**action=unset**

HTTP Method:POST

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Content-Type:application/json

Request Payload:

```
{"appfwpolicy":{
```

```
<b>"name":<String_value>,

</b>"comment":true,

"logaction":true

}}
```

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error

rename

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/appfwpolicy?**action=rename**

HTTP Method:POST

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Content-Type:application/json

Request Payload:

```
{"appfwpolicy":{

<b>"name":<String_value>,

</b><b>"newname":<String_value>

</b>}}
```

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error

get (all)

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/appfwpolicy

Query-parameters:

attrs

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/appfwpolicy?**attrs=property-name1,property-name2**

Use this query parameter to specify the resource details that you want to retrieve.

filter

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/appfwpolicy?**filter=property-name1:property-val1,property-name2:property-val2**

Use this query-parameter to get the filtered set of appfwpolicy resources configured on Citrix ADC. Filtering can be done on any of the properties of the resource.

view

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/appfwpolicy?**view=summary**

Note: By default, the retrieved results are displayed in detail view (?view=detail).

pagination

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/appfwpolicy?**pagesize=#no;pageno=#no**

Use this query-parameter to get the appfwpolicy resources in chunks.

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "appfwpolicy": [ {

"name":<String_value>,

"rule":<String_value>,

"profilename":<String_value>,

"hits":<Double_value>,

"undefhits":<Double_value>,

"comment":<String_value>,

"logaction":<String_value>,

"policytype":<String_value>

}]}
```

get

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/appfwpolicy/name_value<String>

Query-parameters:

attrs

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/appfwpolicy/name_value<String>?**attrs=property-name1,property-name2**

Use this query parameter to specify the resource details that you want to retrieve.

view

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/appfwpolicy/name_value<String>?**view=summary**

Note: By default, the retrieved results are displayed in detail view (?view=detail).

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "appfwpolicy": [ {

"name":<String_value>,

"rule":<String_value>,

"profilename":<String_value>,

"hits":<Double_value>,

"undefhits":<Double_value>,

"comment":<String_value>,

"logaction":<String_value>,

"policytype":<String_value>

}]}
```

count

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/appfwpolicy?**count=yes**

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "appfwpolicy": [ { "__count": "#no"} ] }
```

appfwpolicylabel

Configuration for application firewall policy label resource.

Properties

(click to see Operations)

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
labelname	<String>	Read-write	Name for the policy label. Must begin with a letter, number, or the underscore character (_), and must contain only letters, numbers, and the hyphen (-), period (.) pound (#), space ( ), at (@), equals (=), colon (:), and underscore characters. Can be changed after the

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
			policy label is created. The following requirement applies only to the Citrix ADC CLI: If the name includes one or more spaces, enclose the name in double or single quotation marks (for example, "my policy label" or 'my policy label').
policylabeltype	<String>	Read-write	Type of transformations allowed by the policies bound to the label. Always http_req for application firewall policy labels. Possible values = http_req
newname	<String>	Read-write	The new name of the application firewall policylabel. Minimum length = 1
numpol	<Double>	Read-only	Number of polices bound to label.
hits	<Double>	Read-only	Number of times policy label was invoked.

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
priority	<Double>	Read-only	Positive integer specifying the priority of the policy. A lower number specifies a higher priority. Must be unique within a group of policies that are bound to the same bind point or label. Policies are evaluated in the order of their priority numbers.
gotopriorityexpression	<String>	Read-only	Expression specifying the priority of the next policy which will get evaluated if the current policy rule evaluates to TRUE.
labeltype	<String>	Read-only	Type of policy label to invoke if the current policy evaluates to TRUE and the invoke parameter is set. Available settings function as follows: * reqvserver. Invoke the unnamed policy label associated with the specified request virtual server.

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
			* policylabel. Invoke the specified user-defined policy label. Possible values = reqvserver, policylabel
invoke_labelname	<String>	Read-only	Name of the policy label to invoke if the current policy evaluates to TRUE, the invoke parameter is set, and Label Type is set to Policy Label.
description	<String>	Read-only	Description of the policylabel.
policytype	<String>	Read-only	. Possible values = Classic Policy, Advanced Policy
__count	<Double>	Read-only	count parameter

Operations

(click to see Properties)

ADD| DELETE| RENAME| GET (ALL)| GET| COUNT

Some options that you can use for each operations:

- **Getting warnings in response:**NITRO allows you to get warnings in an operation by specifying the "warning" query parameter as "yes". For example, to get warnings while connecting to the Citrix ADC appliance, the URL is as follows:

  http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/login?warning=yes

If any, the warnings are displayed in the response payload with the HTTP code "209 X-NITRO-WARNING".

- **Authenticated access for individual NITRO operations:**NITRO allows you to logon to the Citrix ADC appliance to perform individual operations. You can use this option instead of creating a NITRO session (using the login object) and then using that session to perform all operations,

  To do this, you must specify the username and password in the request header of the NITRO request as follows:

  X-NITRO-USER:<username>

  X-NITRO-PASS:<password>

  Note: In such cases, make sure that the request header DOES not include the following:

  Cookie:NITRO_AUTH_TOKEN=<tokenvalue>

`*Note:`*

Mandatory parameters are marked in redand placeholder content is marked in <green>.

add

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/appfwpolicylabel

HTTP Method:POST

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Content-Type:application/json

Request Payload:

```
{"appfwpolicylabel":{

<b>"labelname":<String_value>,

</b><b>"policylabeltype":<String_value>

</b>}}
```

Response:

HTTP Status Code on Success: 201 Created HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error

delete

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/appfwpolicylabel/labelname_value<String>

HTTP Method:DELETE

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue>

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error

rename

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/appfwpolicylabel?**action=rename**

HTTP Method:POST

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Content-Type:application/json

Request Payload:

```
{"appfwpolicylabel":{

<b>"labelname":<String_value>,

</b><b>"newname":<String_value>
```

```
</b>}}
```

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error

get (all)

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/appfwpolicylabel

Query-parameters:

attrs

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/appfwpolicylabel?**attrs=property-name1,property-name2**

Use this query parameter to specify the resource details that you want to retrieve.

filter

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/appfwpolicylabel?**filter=property-name1:property-val1,property-name2:property-val2**

Use this query-parameter to get the filtered set of appfwpolicylabel resources configured on Citrix ADC. Filtering can be done on any of the properties of the resource.

view

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/appfwpolicylabel?**view=summary**

Note: By default, the retrieved results are displayed in detail view (?view=detail).

pagination

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/appfwpolicylabel?**pagesize=#no;pageno=#no**

Use this query-parameter to get the appfwpolicylabel resources in chunks.

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "appfwpolicylabel": [ {

"labelname":<String_value>,

"policylabeltype":<String_value>,

"numpol":<Double_value>,

"hits":<Double_value>,

"priority":<Double_value>,

"gotopriorityexpression":<String_value>,

"labeltype":<String_value>,

"invoke_labelname":<String_value>,

"description":<String_value>,

"policytype":<String_value>

}]}
```

get

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/appfwpolicylabel/labelname_value<String>

Query-parameters:

attrs

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/appfwpolicylabel/labelname_value<String>?**attrs=property-name1,property-name2**

Use this query parameter to specify the resource details that you want to retrieve.

view

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/appfwpolicylabel/labelname_value<String>?**view=summary**

Note: By default, the retrieved results are displayed in detail view (?view=detail).

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "appfwpolicylabel": [ {

"labelname":<String_value>,

"policylabeltype":<String_value>,
```

```
"numpol":<Double_value>,

"hits":<Double_value>,

"priority":<Double_value>,

"gotopriorityexpression":<String_value>,

"labeltype":<String_value>,

"invoke_labelname":<String_value>,

"description":<String_value>,

"policytype":<String_value>

}]}
```

count

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/appfwpolicylabel?**count=yes**

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "appfwpolicylabel": [ { "__count": "#no"} ] }
```

appfwpolicylabel_appfwpolicy_binding

Binding object showing the appfwpolicy that can be bound to appfwpolicylabel.

Properties

(click to see Operations)

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
priority	<Double>	Read-write	Positive integer specifying the priority of the policy. A lower number specifies a higher priority. Must be unique within a group of policies that are bound to the same bind point or label. Policies are evaluated in the order of their priority numbers.
policyname	<String>	Read-write	Name of the application firewall policy to bind to the policy label.
labelname	<String>	Read-write	Name of the application firewall policy label.
invoke_labelname	<String>	Read-write	Name of the policy label to invoke if the current policy evaluates to TRUE, the invoke parameter is set, and Label Type is set to Policy Label.

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
gotopriorityexpression	<String>	Read-write	Expression specifying the priority of the next policy which will get evaluated if the current policy rule evaluates to TRUE.
invoke	<Boolean>	Read-write	If the current policy evaluates to TRUE, terminate evaluation of policies bound to the current policy label, and then forward the request to the specified virtual server or evaluate the specified policy label.
labeltype	<String>	Read-write	Type of policy label to invoke if the current policy evaluates to TRUE and the invoke parameter is set. Available settings function as follows: * reqvserver. Invoke the unnamed policy label associated with the specified request virtual server. * policylabel. Invoke the specified user-defined policy label. Possible values = reqvserver, policylabel

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
__count	<Double>	Read-write	count parameter

Operations

(click to see Properties)

ADD:| DELETE:| GET| GET (ALL)| COUNT

Some options that you can use for each operations:

- **Getting warnings in response:**NITRO allows you to get warnings in an operation by specifying the "warning" query parameter as "yes". For example, to get warnings while connecting to the Citrix ADC appliance, the URL is as follows:

  http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/login?warning=yes

  If any, the warnings are displayed in the response payload with the HTTP code "209 X-NITRO-WARNING".
- **Authenticated access for individual NITRO operations:**NITRO allows you to logon to the Citrix ADC appliance to perform individual operations. You can use this option instead of creating a NITRO session (using the login object) and then using that session to perform all operations,

  To do this, you must specify the username and password in the request header of the NITRO request as follows:

  X-NITRO-USER:<username>

  X-NITRO-PASS:<password>

  Note: In such cases, make sure that the request header DOES not include the following:

  Cookie:NITRO_AUTH_TOKEN=<tokenvalue>

`*Note:`*

Mandatory parameters are marked in redand placeholder content is marked in <green>.

add

URL:http://<Citrix-ADC-IP-address(NSIP)/nitro/v1/config/appfwpolicylabel_appfwpolicy_binding

HTTP Method:PUT

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Content-Type:application/json

Request Payload:

```
{

"appfwpolicylabel_appfwpolicy_binding":{

<b>"labelname":<String_value>,

</b><b>"policyname":<String_value>,

</b><b>"priority":<Double_value>,

</b>"gotopriorityexpression":<String_value>,

"invoke":<Boolean_value>,

"labeltype":<String_value>,

"invoke_labelname":<String_value>

}}
```

Response:

HTTP Status Code on Success: 201 Created HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error

delete

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/appfwpolicylabel_appfwpolicy_binding/labelname_value<String>

Query-parameters:

args

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/appfwpolicylabel_appfwpolicy_binding/labelname_value<String>?**args=policyname:<String_value>,priority:<Double_value>**

HTTP Method:DELETE

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue>

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error

get

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/appfwpolicylabel_appfwpolicy_binding/labelname_value<String>

Query-parameters:

filter

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/appfwpolicylabel_appfwpolicy_binding/labelname_value<String>?**filter=property-name1:property-value1,property-name2:property-value2**

Use this query-parameter to get the filtered set of appfwpolicylabel_appfwpolicy_binding resources configured on Citrix ADC. Filtering can be done on any of the properties of the resource.

pagination

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/appfwpolicylabel_appfwpolicy_binding/labelname_value<String>?**pagesize=#no;pageno=#no**

Use this query-parameter to get the appfwpolicylabel_appfwpolicy_binding resources in chunks.

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "appfwpolicylabel_appfwpolicy_binding": [ {

"priority":<Double_value>,

"policyname":<String_value>,

"labelname":<String_value>,

"invoke_labelname":<String_value>,

"gotopriorityexpression":<String_value>,

"invoke":<Boolean_value>,

"labeltype":<String_value>

}]}
```

get (all)

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/appfwpolicylabel_appfwpolicy_binding

Query-parameters:

bulkbindings

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/appfwpolicylabel_appfwpolicy_binding?**bulkbindings=yes**

NITRO allows you to fetch bindings in bulk.

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "appfwpolicylabel_appfwpolicy_binding": [ {

"priority":<Double_value>,

"policyname":<String_value>,

"labelname":<String_value>,

"invoke_labelname":<String_value>,

"gotopriorityexpression":<String_value>,

"invoke":<Boolean_value>,

"labeltype":<String_value>

}]}
```

count

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/appfwpolicylabel_appfwpolicy_binding/labelname_value<String>?**count=yes**

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{"appfwpolicylabel_appfwpolicy_binding": [ { "__count": "#no"} ] }
```

appfwpolicylabel_binding

Binding object which returns the resources bound to appfwpolicylabel.

Properties

(click to see Operations)

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
labelname	<String>	Read-write	Name of the application firewall policy label.
appfwpolicylabel_policybinding_binding	<appfwpolicylabel_policybinding_binding[]>	Read-only	policybinding that can be bound to appfwpolicylabel.
appfwpolicylabel_appfwpolicy_binding	<appfwpolicylabel_appfwpolicy_binding[]>	Read-only	appfwpolicy that can be bound to appfwpolicylabel.

Operations

(click to see Properties)

GET| GET (ALL)

Some options that you can use for each operations:

- **Getting warnings in response:**NITRO allows you to get warnings in an operation by specifying the "warning" query parameter as "yes". For example, to get warnings while connecting to the Citrix ADC appliance, the URL is as follows:

  http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/login?warning=yes

  If any, the warnings are displayed in the response payload with the HTTP code "209 X-NITRO-WARNING".

- **Authenticated access for individual NITRO operations:**NITRO allows you to logon to the Citrix ADC appliance to perform individual operations. You can use this option instead of creating a NITRO session (using the login object) and then using that session to perform all operations,

  To do this, you must specify the username and password in the request header of the NITRO request as follows:

  X-NITRO-USER:<username>

  X-NITRO-PASS:<password>

  Note: In such cases, make sure that the request header DOES not include the following:

  Cookie:NITRO_AUTH_TOKEN=<tokenvalue>

`*Note:`*

Mandatory parameters are marked in redand placeholder content is marked in <green>.

get

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/appfwpolicylabel_binding/labelname_value<String>

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "appfwpolicylabel_binding": [ {

"labelname":<String_value>,

"appfwpolicylabel_appfwpolicy_binding":<appfwpolicylabel_appfwpolicy_binding[]_value>,

"appfwpolicylabel_policybinding_binding":<appfwpolicylabel_policybinding_binding[]_value>

}]}
```

get (all)

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/appfwpolicylabel_binding

Query-parameters:

bulkbindings

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/appfwpolicylabel_binding?**bulkbindings=yes**

NITRO allows you to fetch bindings in bulk.

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "appfwpolicylabel_binding": [ {

"labelname":<String_value>,

"appfwpolicylabel_appfwpolicy_binding":<appfwpolicylabel_appfwpolicy_binding[]_value>,

"appfwpolicylabel_policybinding_binding":<appfwpolicylabel_policybinding_binding[]_value>

}]}
```

appfwpolicylabel_policybinding_binding

Binding object showing the policybinding that can be bound to appfwpolicylabel.

Properties

(click to see Operations)

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
priority	<Double>	Read-write	Positive integer specifying the priority of the policy. A lower number specifies a higher priority. Must be unique within a group of policies that are bound to the same bind point or label. Policies are

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
			evaluated in the order of their priority numbers.
policyname	<String>	Read-write	Name of the application firewall policy to bind to the policy label.
labelname	<String>	Read-write	Name of the application firewall policy label.
invoke_labelname	<String>	Read-write	Name of the policy label to invoke if the current policy evaluates to TRUE, the invoke parameter is set, and Label Type is set to Policy Label.
gotopriorityexpression	<String>	Read-write	Expression specifying the priority of the next policy which will get evaluated if the current policy rule evaluates to TRUE.
invoke	<Boolean>	Read-write	If the current policy evaluates to TRUE, terminate evaluation of policies bound to the current policy label, and then forward the request to the specified virtual server or evaluate the specified policy label.

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
labeltype	<String>	Read-write	Type of policy label to invoke if the current policy evaluates to TRUE and the invoke parameter is set. Available settings function as follows: * reqvserver. Invoke the unnamed policy label associated with the specified request virtual server. * policylabel. Invoke the specified user-defined policy label. Possible values = reqvserver, policylabel
__count	<Double>	Read-write	count parameter

Operations

(click to see Properties)

GET| GET (ALL)| COUNT

Some options that you can use for each operations:

- **Getting warnings in response:**NITRO allows you to get warnings in an operation by specifying the "warning" query parameter as "yes". For example, to get warnings while connecting to the Citrix ADC appliance, the URL is as follows:

  http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/login?warning=yes

  If any, the warnings are displayed in the response payload with the HTTP code "209 X-NITRO-WARNING".

- **Authenticated access for individual NITRO operations:**NITRO allows you to logon to the Citrix ADC appliance to perform individual operations. You can use this option instead of

creating a NITRO session (using the login object) and then using that session to perform all operations,

To do this, you must specify the username and password in the request header of the NITRO request as follows:

X-NITRO-USER:<username>

X-NITRO-PASS:<password>

Note: In such cases, make sure that the request header DOES not include the following:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue>

`*Note:`*

Mandatory parameters are marked in redand placeholder content is marked in <green>.

get

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/appfwpolicylabel_policybinding_binding/labelname_value<String>

Query-parameters:

filter

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/appfwpolicylabel_policybinding_binding/labelname_value<String>?**filter=property-name1:property-value1,property-name2:property-value2**

Use this query-parameter to get the filtered set of appfwpolicylabel_policybinding_binding resources configured on Citrix ADC. Filtering can be done on any of the properties of the resource.

pagination

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/appfwpolicylabel_policybinding_binding/labelname_value<String>?**pagesize=#no;pageno=#no**

Use this query-parameter to get the appfwpolicylabel_policybinding_binding resources in chunks.

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "appfwpolicylabel_policybinding_binding": [ {

"priority":<Double_value>,

"policyname":<String_value>,

"labelname":<String_value>,

"invoke_labelname":<String_value>,

"gotopriorityexpression":<String_value>,

"invoke":<Boolean_value>,

"labeltype":<String_value>

}]}
```

get (all)

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/appfwpolicylabel_policybinding_binding

Query-parameters:

bulkbindings

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/appfwpolicylabel_policybinding_binding?**bulkbindings=yes**

NITRO allows you to fetch bindings in bulk.

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "appfwpolicylabel_policybinding_binding": [ {

"priority":<Double_value>,

"policyname":<String_value>,

"labelname":<String_value>,

"invoke_labelname":<String_value>,

"gotopriorityexpression":<String_value>,

"invoke":<Boolean_value>,

"labeltype":<String_value>

}]}
```

count

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/appfwpolicylabel_policybinding_binding/labelname_value<String>?**count=yes**

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{"appfwpolicylabel_policybinding_binding": [ { "__count": "#no"} ] }
```

appfwpolicy_appfwglobal_binding

Binding object showing the appfwglobal that can be bound to appfwpolicy.

Properties

(click to see Operations)

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
name	<String>	Read-write	Name of the policy. Minimum length = 1
boundto	<String>	Read-write	The entity name to which policy is bound.
priority	<Double>	Read-only	Specifies the priority of the policy.
gotopriorityexpression	<String>	Read-only	Expression specifying the priority of the next policy which will get evaluated if the

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
			current policy rule evaluates to TRUE.
labeltype	<String>	Read-only	Type of policy label invocation. Possible values = reqvserver, resvserver, policylabel
activepolicy	<Integer>	Read-only	Indicates whether policy is bound or not.
labelname	<String>	Read-only	Name of the label to invoke if the current policy rule evaluates to TRUE.
__count	<Double>	Read-only	count parameter

Operations

(click to see Properties)

GET| GET (ALL)| COUNT

Some options that you can use for each operations:

- **Getting warnings in response:**NITRO allows you to get warnings in an operation by specifying the "warning" query parameter as "yes". For example, to get warnings while connecting to the Citrix ADC appliance, the URL is as follows:

  http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/login?warning=yes

  If any, the warnings are displayed in the response payload with the HTTP code "209 X-NITRO-WARNING".
- **Authenticated access for individual NITRO operations:**NITRO allows you to logon to the Citrix ADC appliance to perform individual operations. You can use this option instead of

creating a NITRO session (using the login object) and then using that session to perform all operations,

To do this, you must specify the username and password in the request header of the NITRO request as follows:

X-NITRO-USER:<username>

X-NITRO-PASS:<password>

Note: In such cases, make sure that the request header DOES not include the following:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue>

`*Note:`*

Mandatory parameters are marked in redand placeholder content is marked in <green>.

get

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/appfwpolicy_appfwglobal_binding/name_value<String>

Query-parameters:

filter

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/appfwpolicy_appfwglobal_binding/name_value<String>?**filter=property-name1:property-value1,property-name2:property-value2**

Use this query-parameter to get the filtered set of appfwpolicy_appfwglobal_binding resources configured on Citrix ADC. Filtering can be done on any of the properties of the resource.

pagination

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/appfwpolicy_appfwglobal_binding/name_value<String>?**pagesize=#no;pageno=#no**

Use this query-parameter to get the appfwpolicy_appfwglobal_binding resources in chunks.

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "appfwpolicy_appfwglobal_binding": [ {

"name":<String_value>,

"boundto":<String_value>,

"priority":<Double_value>,

"gotopriorityexpression":<String_value>,

"labeltype":<String_value>,

"activepolicy":<Integer_value>,

"labelname":<String_value>

}]}
```

get (all)

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/appfwpolicy_appfwglobal_binding

Query-parameters:

bulkbindings

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/appfwpolicy_appfwglobal_binding?**bulkbindings=yes**

NITRO allows you to fetch bindings in bulk.

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "appfwpolicy_appfwglobal_binding": [ {

"name":<String_value>,

"boundto":<String_value>,

"priority":<Double_value>,

"gotopriorityexpression":<String_value>,

"labeltype":<String_value>,

"activepolicy":<Integer_value>,

"labelname":<String_value>

}]}
```

count

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/appfwpolicy_appfwglobal_binding/name_value<String>?**count=yes**

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{"appfwpolicy_appfwglobal_binding": [ { "__count": "#no"} ] }
```

appfwpolicy_appfwpolicylabel_binding

Binding object showing the appfwpolicylabel that can be bound to appfwpolicy.

Properties

(click to see Operations)

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
name	<String>	Read-write	Name of the policy. Minimum length = 1
boundto	<String>	Read-write	The entity name to which policy is bound.
priority	<Double>	Read-only	Specifies the priority of the policy.
gotopriorityexpression	<String>	Read-only	Expression specifying the priority of the next policy which will get evaluated if the

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
			current policy rule evaluates to TRUE.
labeltype	<String>	Read-only	Type of policy label invocation. Possible values = reqvserver, resvserver, policylabel
activepolicy	<Integer>	Read-only	Indicates whether policy is bound or not.
labelname	<String>	Read-only	Name of the label to invoke if the current policy rule evaluates to TRUE.
__count	<Double>	Read-only	count parameter

Operations

(click to see Properties)

GET| GET (ALL)| COUNT

Some options that you can use for each operations:

- **Getting warnings in response:**NITRO allows you to get warnings in an operation by specifying the "warning" query parameter as "yes". For example, to get warnings while connecting to the Citrix ADC appliance, the URL is as follows:

  http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/login?warning=yes

  If any, the warnings are displayed in the response payload with the HTTP code "209 X-NITRO-WARNING".
- **Authenticated access for individual NITRO operations:**NITRO allows you to logon to the Citrix ADC appliance to perform individual operations. You can use this option instead of

creating a NITRO session (using the login object) and then using that session to perform all operations,

To do this, you must specify the username and password in the request header of the NITRO request as follows:

X-NITRO-USER:<username>

X-NITRO-PASS:<password>

Note: In such cases, make sure that the request header DOES not include the following:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue>

`*Note:`*

Mandatory parameters are marked in redand placeholder content is marked in <green>.

get

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/appfwpolicy_appfwpolicylabel_binding/name_value<String>

Query-parameters:

filter

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/appfwpolicy_appfwpolicylabel_binding/name_value<String>?**filter=property-name1:property-value1,property-name2:property-value2**

Use this query-parameter to get the filtered set of appfwpolicy_appfwpolicylabel_binding resources configured on Citrix ADC. Filtering can be done on any of the properties of the resource.

pagination

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/appfwpolicy_appfwpolicylabel_binding/name_value<String>?**pagesize=#no;pageno=#no**

Use this query-parameter to get the appfwpolicy_appfwpolicylabel_binding resources in chunks.

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "appfwpolicy_appfwpolicylabel_binding": [ {

"name":<String_value>,

"boundto":<String_value>,

"priority":<Double_value>,

"gotopriorityexpression":<String_value>,

"labeltype":<String_value>,

"activepolicy":<Integer_value>,

"labelname":<String_value>

}]}
```

get (all)

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/appfwpolicy_appfwpolicylabel_binding

Query-parameters:

bulkbindings

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/appfwpolicy_appfwpolicylabel_binding?**bulkbindings=yes**

NITRO allows you to fetch bindings in bulk.

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "appfwpolicy_appfwpolicylabel_binding": [ {

"name":<String_value>,

"boundto":<String_value>,

"priority":<Double_value>,

"gotopriorityexpression":<String_value>,

"labeltype":<String_value>,

"activepolicy":<Integer_value>,

"labelname":<String_value>

}]}
```

count

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/appfwpolicy_appfwpolicylabel_binding/name_value<String>?**count=yes**

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{"appfwpolicy_appfwpolicylabel_binding": [ { "__count": "#no"} ] }
```

appfwpolicy_binding

Binding object which returns the resources bound to appfwpolicy.

Properties

(click to see Operations)

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
name	<String>	Read-write	Name of the policy. Minimum length = 1
appfwpolicy_appfwglobal_binding	<appfwpolicy_appfwglobal_binding[]>	Read-only	appfwglobal that can be bound to appfwpolicy.
appfwpolicy_lbvserver_binding	<appfwpolicy_lbvserver_binding[]>	Read-only	lbvserver that can be bound to appfwpolicy.
appfwpolicy_csvserver_binding	<appfwpolicy_csvserver_binding[]>	Read-only	csvserver that can be bound to appfwpolicy.

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
appfwpolicy_appfwpolicylabel_binding	<appfwpolicy_appfwpolicylabel_binding[]>	Read-only	appfwpolicylabel that can be bound to appfwpolicy.

Operations

(click to see Properties)

GET| GET (ALL)

Some options that you can use for each operations:

- **Getting warnings in response:**NITRO allows you to get warnings in an operation by specifying the "warning" query parameter as "yes". For example, to get warnings while connecting to the Citrix ADC appliance, the URL is as follows:

  http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/login?warning=yes

  If any, the warnings are displayed in the response payload with the HTTP code "209 X-NITRO-WARNING".

- **Authenticated access for individual NITRO operations:**NITRO allows you to logon to the Citrix ADC appliance to perform individual operations. You can use this option instead of creating a NITRO session (using the login object) and then using that session to perform all operations,

  To do this, you must specify the username and password in the request header of the NITRO request as follows:

  X-NITRO-USER:<username>

  X-NITRO-PASS:<password>

  Note: In such cases, make sure that the request header DOES not include the following:

  Cookie:NITRO_AUTH_TOKEN=<tokenvalue>

`*Note:`*

Mandatory parameters are marked in redand placeholder content is marked in <green>.

get

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/appfwpolicy_binding/name_value<String>

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "appfwpolicy_binding": [ {

"name":<String_value>,

"appfwpolicy_appfwpolicylabel_binding":<appfwpolicy_appfwpolicylabel_binding[]_value>,

"appfwpolicy_csvserver_binding":<appfwpolicy_csvserver_binding[]_value>,

"appfwpolicy_appfwglobal_binding":<appfwpolicy_appfwglobal_binding[]_value>,

"appfwpolicy_lbvserver_binding":<appfwpolicy_lbvserver_binding[]_value>

}]}
```

get (all)

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/appfwpolicy_binding

Query-parameters:

bulkbindings

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/appfwpolicy_binding?**bulkbindings=yes**

NITRO allows you to fetch bindings in bulk.

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "appfwpolicy_binding": [ {

"name":<String_value>,

"appfwpolicy_appfwpolicylabel_binding":<appfwpolicy_appfwpolicylabel_bindi
ng[]_value>,

"appfwpolicy_csvserver_binding":<appfwpolicy_csvserver_binding[]_value>,

"appfwpolicy_appfwglobal_binding":<appfwpolicy_appfwglobal_binding[]_value
>,

"appfwpolicy_lbvserver_binding":<appfwpolicy_lbvserver_binding[]_value>

}]}
```

appfwpolicy_csvserver_binding

Binding object showing the csvserver that can be bound to appfwpolicy.

Properties

(click to see Operations)

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
name	<String>	Read-write	Name of the policy. Minimum length = 1
boundto	<String>	Read-write	The entity name to which policy is bound.
priority	<Double>	Read-only	Specifies the priority of the policy.
gotopriorityexpression	<String>	Read-only	Expression specifying the priority of the next policy which will get evaluated if the current policy rule evaluates to TRUE.
labeltype	<String>	Read-only	Type of policy label invocation. Possible values = reqvserver, resvserver, policylabel
activepolicy	<Integer>	Read-only	Indicates whether policy is bound or not.
labelname	<String>	Read-only	Name of the label to invoke if the current policy rule evaluates to TRUE.
__count	<Double>	Read-only	count parameter

Operations

(click to see Properties)

GET| GET (ALL)| COUNT

Some options that you can use for each operations:

- **Getting warnings in response:**NITRO allows you to get warnings in an operation by specifying the "warning" query parameter as "yes". For example, to get warnings while connecting to the Citrix ADC appliance, the URL is as follows:

  http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/login?warning=yes

  If any, the warnings are displayed in the response payload with the HTTP code "209 X-NITRO-WARNING".

- **Authenticated access for individual NITRO operations:**NITRO allows you to logon to the Citrix ADC appliance to perform individual operations. You can use this option instead of creating a NITRO session (using the login object) and then using that session to perform all operations,

  To do this, you must specify the username and password in the request header of the NITRO request as follows:

  X-NITRO-USER:<username>

  X-NITRO-PASS:<password>

  Note: In such cases, make sure that the request header DOES not include the following:

  Cookie:NITRO_AUTH_TOKEN=<tokenvalue>

`*Note:`*

Mandatory parameters are marked in redand placeholder content is marked in <green>.

get

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/appfwpolicy_csvserver_binding/name_value<String>

Query-parameters:

filter

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/appfwpolicy_csvserver_binding/name_value<String>?**filter=property-name1:property-value1,property-name2:property-value2**

Use this query-parameter to get the filtered set of appfwpolicy_csvserver_binding resources configured on Citrix ADC. Filtering can be done on any of the properties of the resource.

pagination

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/appfwpolicy_csvserver_binding/name_value<String>?**pagesize=#no;pageno=#no**

Use this query-parameter to get the appfwpolicy_csvserver_binding resources in chunks.

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "appfwpolicy_csvserver_binding": [ {

"name":<String_value>,

"boundto":<String_value>,

"priority":<Double_value>,

"gotopriorityexpression":<String_value>,

"labeltype":<String_value>,

"activepolicy":<Integer_value>,
```

```
"labelname":<String_value>

}]}
```

get (all)

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/appfwpolicy_csvserver_binding

Query-parameters:

bulkbindings

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/appfwpolicy_csvserver_binding?**bulkbindings=yes**

NITRO allows you to fetch bindings in bulk.

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "appfwpolicy_csvserver_binding": [ {

"name":<String_value>,

"boundto":<String_value>,

"priority":<Double_value>,
```

```
"gotopriorityexpression":<String_value>,

"labeltype":<String_value>,

"activepolicy":<Integer_value>,

"labelname":<String_value>

}]}
```

count

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/appfwpolicy_csvserver_binding/name_value<String>?**count=yes**

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{"appfwpolicy_csvserver_binding": [ { "__count": "#no"} ] }
```

appfwpolicy_lbvserver_binding

Binding object showing the lbvserver that can be bound to appfwpolicy.

Properties

(click to see Operations)

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
name	<String>	Read-write	Name of the policy. Minimum length = 1
boundto	<String>	Read-write	The entity name to which policy is bound.
priority	<Double>	Read-only	Specifies the priority of the policy.
gotopriorityexpression	<String>	Read-only	Expression specifying the priority of the next policy which will get evaluated if the current policy rule evaluates to TRUE.
labeltype	<String>	Read-only	Type of policy label invocation. Possible values = reqvserver, resvserver, policylabel
activepolicy	<Integer>	Read-only	Indicates whether policy is bound or not.
labelname	<String>	Read-only	Name of the label to invoke if the current policy rule evaluates to TRUE.
__count	<Double>	Read-only	count parameter

Operations

(click to see Properties)

GET| GET (ALL)| COUNT

Some options that you can use for each operations:

- **Getting warnings in response:**NITRO allows you to get warnings in an operation by specifying the "warning" query parameter as "yes". For example, to get warnings while connecting to the Citrix ADC appliance, the URL is as follows:

  http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/login?warning=yes

  If any, the warnings are displayed in the response payload with the HTTP code "209 X-NITRO-WARNING".

- **Authenticated access for individual NITRO operations:**NITRO allows you to logon to the Citrix ADC appliance to perform individual operations. You can use this option instead of creating a NITRO session (using the login object) and then using that session to perform all operations,

  To do this, you must specify the username and password in the request header of the NITRO request as follows:

  X-NITRO-USER:<username>

  X-NITRO-PASS:<password>

  Note: In such cases, make sure that the request header DOES not include the following:

  Cookie:NITRO_AUTH_TOKEN=<tokenvalue>

`*Note:`*

Mandatory parameters are marked in redand placeholder content is marked in <green>.

get

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/appfwpolicy_lbvserver_binding/name_value<String>

Query-parameters:

filter

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/appfwpolicy_lbvserver_binding/name_value<String>?**filter=property-name1:property-value1,property-name2:property-value2**

Use this query-parameter to get the filtered set of appfwpolicy_lbvserver_binding resources configured on Citrix ADC. Filtering can be done on any of the properties of the resource.

pagination

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/appfwpolicy_lbvserver_binding/name_value<String>?**pagesize=#no;pageno=#no**

Use this query-parameter to get the appfwpolicy_lbvserver_binding resources in chunks.

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "appfwpolicy_lbvserver_binding": [ {

"name":<String_value>,

"boundto":<String_value>,

"priority":<Double_value>,

"gotopriorityexpression":<String_value>,

"labeltype":<String_value>,

"activepolicy":<Integer_value>,
```

```
"labelname":<String_value>

}]}
```

get (all)

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/appfwpolicy_lbvserver_binding

Query-parameters:

bulkbindings

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/appfwpolicy_lbvserver_binding?**bulkbindings=yes**

NITRO allows you to fetch bindings in bulk.

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "appfwpolicy_lbvserver_binding": [ {

"name":<String_value>,

"boundto":<String_value>,

"priority":<Double_value>,
```

```
"gotopriorityexpression":<String_value>,

"labeltype":<String_value>,

"activepolicy":<Integer_value>,

"labelname":<String_value>

}]}
```

count

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/appfwpolicy_lbvserver_binding/name_value<String>?**count=yes**

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{"appfwpolicy_lbvserver_binding": [ { "__count": "#no"} ] }
```

appfwprofile

Configuration for application firewall profile resource.

Properties

(click to see Operations)

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
name	<String>	Read-write	Name for the profile. Must begin with a letter, number, or the underscore character (_), and must contain only letters, numbers, and the hyphen (-), period (.), pound (#), space ( ), at (@), equals (=), colon (:), and underscore (_) characters. Cannot be changed after the profile is added. The following requirement applies only to the Citrix ADC CLI: If the name includes one or more spaces, enclose the name in double or single quotation marks (for example, "my profile" or 'my profile'). Minimum length = 1
defaults	<String>	Read-write	Default configuration to apply to the profile. Basic defaults are intended for standard content

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
			that requires little further configuration, such as static web site content. Advanced defaults are intended for specialized content that requires significant specialized configuration, such as heavily scripted or dynamic content. CLI users: When adding an application firewall profile, you can set either the defaults or the type, but not both. To set both options, create the profile by using the add appfw profile command, and then use the set appfw profile command to configure the other option. Possible values = basic, advanced
starturlaction	<String[]>	Read-write	One or more Start URL actions. Available settings function as follows: * Block - Block connections that

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
			violate this security check. * Learn - Use the learning engine to generate a list of exceptions to this security check. * Log - Log violations of this security check. * Stats - Generate statistics for this security check. * None - Disable all actions for this security check. CLI users: To enable one or more actions, type "set appfw profile -startURLaction" followed by the actions to be enabled. To turn off all actions, type "set appfw profile -startURLaction none". Possible values = none, block, learn, log, stats

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
contenttypeaction	<String[]>	Read-write	One or more Content-type actions. Available settings function as follows: * Block - Block connections that violate this security check. * Learn - Use the learning engine to generate a list of exceptions to this security check. * Log - Log violations of this security check. * Stats - Generate statistics for this security check. * None - Disable all actions for this security check. CLI users: To enable one or more actions, type "set appfw profile -contentTypeaction" followed by the actions to be enabled. To turn off all actions, type "set appfw profile

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
			-contentTypeaction none". Possible values = none, block, learn, log, stats
inspectcontenttypes	<String[]>	Read-write	One or more InspectContentType lists. * application/x-www-form-urlencoded * multipart/form-data * text/x-gwt-rpc CLI users: To enable, type "set appfw profile -InspectContentTypes" followed by the content types to be inspected. Possible values = none, application/x-www-form-urlencoded, multipart/form-data, text/x-gwt-rpc
starturlclosure	<String>	Read-write	Toggle the state of Start URL Closure. Default value: OFF

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
			Possible values = ON, OFF
denyurlaction	<String[]>	Read-write	One or more Deny URL actions. Available settings function as follows: * Block - Block connections that violate this security check. * Log - Log violations of this security check. * Stats - Generate statistics for this security check. * None - Disable all actions for this security check. NOTE: The Deny URL check takes precedence over the Start URL check. If you enable blocking for the Deny URL check, the application firewall blocks any URL that is explicitly blocked by a Deny URL, even if the same URL would otherwise be

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
			allowed by the Start URL check. CLI users: To enable one or more actions, type "set appfw profile -denyURLaction" followed by the actions to be enabled. To turn off all actions, type "set appfw profile -denyURLaction none". Possible values = none, block, log, stats
refererheadercheck	<String>	Read-write	Enable validation of Referer headers. Referer validation ensures that a web form that a user sends to your web site originally came from your web site, not an outside attacker. Although this parameter is part of the Start URL check, referer validation protects against cross-site

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
			request forgery (CSRF) attacks, not Start URL attacks. Default value: OFF Possible values = OFF, if_present, AlwaysExceptStart URLs, AlwaysExceptFirstRequest

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
cookieconsistencyaction	<String[]>	Read-write	One or more Cookie Consistency actions. Available settings function as follows: * Block - Block connections that violate this security check. * Learn - Use the learning engine to generate a list of exceptions to this security check. * Log - Log violations of this security check. * Stats - Generate statistics for this security check. * None - Disable all actions for this security check. CLI users: To enable one or more actions, type "set appfw profile -cookieConsistency Action" followed by the actions to be enabled. To turn off all actions, type "set appfw profile

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
			-cookieConsistency Action none". Default value: none Possible values = none, block, learn, log, stats
cookietransforms	<String>	Read-write	Perform the specified type of cookie transformation. Available settings function as follows: * Encryption - Encrypt cookies. * Proxying - Mask contents of server cookies by sending proxy cookie to users. * Cookie flags - Flag cookies as HTTP only to prevent scripts on user's browser from accessing and possibly modifying them. CAUTION: Make sure that this parameter is set to ON if you are configuring any cookie

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
			transformations. If it is set to OFF, no cookie transformations are performed regardless of any other settings. Default value: OFF Possible values = ON, OFF
cookieencryption	<String>	Read-write	Type of cookie encryption. Available settings function as follows: * None - Do not encrypt cookies. * Decrypt Only - Decrypt encrypted cookies, but do not encrypt cookies. * Encrypt Session Only - Encrypt session cookies, but not permanent cookies. * Encrypt All - Encrypt all cookies. Default value: none Possible values = none, decryptOnly,

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
			encryptSessionOnly , encryptAll
cookieproxying	<String>	Read-write	Cookie proxy setting. Available settings function as follows: * None - Do not proxy cookies. * Session Only - Proxy session cookies by using the Citrix ADC session ID, but do not proxy permanent cookies. Default value: none Possible values = none, sessionOnly
addcookieflags	<String>	Read-write	Add the specified flags to cookies. Available settings function as follows: * None - Do not add flags to cookies. * HTTP Only - Add the HTTP Only flag to cookies, which prevents scripts from accessing cookies.

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
			* Secure - Add Secure flag to cookies. * All - Add both HTTPOnly and Secure flags to cookies. Default value: none Possible values = none, httpOnly, secure, all
fieldconsistencyaction	<String[]>	Read-write	One or more Form Field Consistency actions. Available settings function as follows: * Block - Block connections that violate this security check. * Learn - Use the learning engine to generate a list of exceptions to this security check. * Log - Log violations of this security check. * Stats - Generate statistics for this security check.

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
			* None - Disable all actions for this security check. CLI users: To enable one or more actions, type "set appfw profile -fieldConsistencyaction" followed by the actions to be enabled. To turn off all actions, type "set appfw profile -fieldConsistencyAction none". Default value: none Possible values = none, block, learn, log, stats
csrftagaction	<String[]>	Read-write	One or more Cross-Site Request Forgery (CSRF) Tagging actions. Available settings function as follows: * Block - Block connections that violate this security check. * Learn - Use the learning engine to generate a list of

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
			exceptions to this security check. * Log - Log violations of this security check. * Stats - Generate statistics for this security check. * None - Disable all actions for this security check. CLI users: To enable one or more actions, type "set appfw profile -CSRFTagAction" followed by the actions to be enabled. To turn off all actions, type "set appfw profile -CSRFTagAction none". Default value: none Possible values = none, block, learn, log, stats

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
crosssitescriptingaction	<String[]>	Read-write	One or more Cross-Site Scripting (XSS) actions. Available settings function as follows: * Block - Block connections that violate this security check. * Learn - Use the learning engine to generate a list of exceptions to this security check. * Log - Log violations of this security check. * Stats - Generate statistics for this security check. * None - Disable all actions for this security check. CLI users: To enable one or more actions, type "set appfw profile -crossSiteScriptingAction" followed by the actions to be enabled. To turn off all actions, type "set appfw profile

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
			-crossSiteScriptingAction none". Possible values = none, block, learn, log, stats
crosssitescriptingtransformunsafehtml	<String>	Read-write	Transform cross-site scripts. This setting configures the application firewall to disable dangerous HTML instead of blocking the request. CAUTION: Make sure that this parameter is set to ON if you are configuring any cross-site scripting transformations. If it is set to OFF, no cross-site scripting transformations are performed regardless of any other settings. Default value: OFF Possible values = ON, OFF
crosssitescriptingcheckcompleteurls	<String>	Read-write	Check complete URLs for cross-site scripts, instead of just the query portions of URLs.

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
			Default value: OFF Possible values = ON, OFF
sqlinjectionaction	<String[]>	Read-write	One or more HTML SQL Injection actions. Available settings function as follows: * Block - Block connections that violate this security check. * Learn - Use the learning engine to generate a list of exceptions to this security check. * Log - Log violations of this security check. * Stats - Generate statistics for this security check. * None - Disable all actions for this security check. CLI users: To enable one or more actions, type "set appfw profile -SQLInjectionAction " followed by the

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
			actions to be enabled. To turn off all actions, type "set appfw profile -SQLInjectionAction none". Possible values = none, block, learn, log, stats
sqlinjectiontransformspecialchars	<String>	Read-write	Transform injected SQL code. This setting configures the application firewall to disable SQL special strings instead of blocking the request. Since most SQL servers require a special string to activate an SQL keyword, in most cases a request that contains injected SQL code is safe if special strings are disabled. CAUTION: Make sure that this parameter is set to ON if you are configuring any SQL injection transformations. If it is set to OFF, no SQL injection transformations are performed

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
			regardless of any other settings. Default value: OFF Possible values = ON, OFF
sqlinjectiononlycheck fieldswithsqlchars	<String>	Read-write	Check only form fields that contain SQL special strings (characters) for injected SQL code. Most SQL servers require a special string to activate an SQL request, so SQL code without a special string is harmless to most SQL servers. Default value: ON Possible values = ON, OFF
sqlinjectiontype	<String>	Read-write	Available SQL injection types. -SQLSplChar : Checks for SQL Special Chars -SQLKeyword : Checks for SQL Keywords -SQLSplCharANDKeyword : Checks for

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
			both and blocks if both are found -SQLSplCharORKeyword : Checks for both and blocks if anyone is found. Default value: SQLSplCharANDKeyword Possible values = SQLSplChar, SQLKeyword, SQLSplCharORKeyword, SQLSplCharANDKeyword
sqlinjectionchecksqlwildchars	<String>	Read-write	Check for form fields that contain SQL wild chars . Default value: OFF Possible values = ON, OFF
fieldformataction	<String[]>	Read-write	One or more Field Format actions. Available settings function as follows: * Block - Block connections that violate this security check.

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
			* Learn - Use the learning engine to generate a list of suggested web form fields and field format assignments. * Log - Log violations of this security check. * Stats - Generate statistics for this security check. * None - Disable all actions for this security check. CLI users: To enable one or more actions, type "set appfw profile -fieldFormatAction" followed by the actions to be enabled. To turn off all actions, type "set appfw profile -fieldFormatAction none". Possible values = none, block, learn, log, stats
defaultfieldformattype	<String>	Read-write	Designate a default field type to be

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
			applied to web form fields that do not have a field type explicitly assigned to them. Minimum length = 1
defaultfieldformatminlength	<Double>	Read-write	Minimum length, in characters, for data entered into a field that is assigned the default field type. To disable the minimum and maximum length settings and allow data of any length to be entered into the field, set this parameter to zero (0). Default value: 0 Minimum value = 0 Maximum value = 2147483647
defaultfieldformatmaxlength	<Double>	Read-write	Maximum length, in characters, for data entered into a field that is assigned the default field type. Default value: 65535 Minimum value = 1

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
			Maximum value = 2147483647
bufferoverflowaction	<String[]>	Read-write	One or more Buffer Overflow actions. Available settings function as follows: * Block - Block connections that violate this security check. * Log - Log violations of this security check. * Stats - Generate statistics for this security check. * None - Disable all actions for this security check. CLI users: To enable one or more actions, type "set appfw profile -bufferOverflowAction" followed by the actions to be enabled. To turn off all actions, type "set appfw profile -bufferOverflowAction none".

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
			Possible values = none, block, log, stats
bufferoverflowmaxurllength	<Double>	Read-write	Maximum length, in characters, for URLs on your protected web sites. Requests with longer URLs are blocked. Default value: 1024 Minimum value = 0 Maximum value = 65535
bufferoverflowmaxheaderlength	<Double>	Read-write	Maximum length, in characters, for HTTP headers in requests sent to your protected web sites. Requests with longer headers are blocked. Default value: 4096 Minimum value = 0 Maximum value = 65535
bufferoverflowmaxcookielength	<Double>	Read-write	Maximum length, in characters, for cookies sent to your protected web sites. Requests with longer cookies are blocked. Default value: 4096 Minimum value = 0

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
			Maximum value = 65535
creditcardaction	<String[]>	Read-write	One or more Credit Card actions. Available settings function as follows: * Block - Block connections that violate this security check. * Log - Log violations of this security check. * Stats - Generate statistics for this security check. * None - Disable all actions for this security check. CLI users: To enable one or more actions, type "set appfw profile -creditCardAction" followed by the actions to be enabled. To turn off all actions, type "set appfw profile -creditCardAction none". Default value: none

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
			Possible values = none, block, learn, log, stats
creditcard	<String[]>	Read-write	Credit card types that the application firewall should protect. Default value: none Possible values = none, visa, mastercard, discover, amex, jcb, dinersclub
creditcardmaxallowed	<Double>	Read-write	This parameter value is used by the block action. It represents the maximum number of credit card numbers that can appear on a web page served by your protected web sites. Pages that contain more credit card numbers are blocked. Minimum value = 0 Maximum value = 255
creditcardxout	<String>	Read-write	Mask any credit card number detected in a response by replacing each digit, except the digits in

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
			the final group, with the letter "X.". Default value: OFF Possible values = ON, OFF
dosecurecreditcardlogging	<String>	Read-write	Setting this option logs credit card numbers in the response when the match is found. Default value: ON Possible values = ON, OFF
streaming	<String>	Read-write	Setting this option converts content-length form submission requests (requests with content-type "application/x-www-form-urlencoded" or "multipart/form-data") to chunked requests when atleast one of the following protections : SQL injection protection, XSS protection, form field consistency protection, starturl closure, CSRF tagging is enabled. Please make sure that the backend server accepts chunked requests

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
			before enabling this option. Default value: OFF Possible values = ON, OFF
trace	<String>	Read-write	Toggle the state of trace. Default value: OFF Possible values = ON, OFF
requestcontenttype	<String>	Read-write	Default Content-Type header for requests. A Content-Type header can contain 0-255 letters, numbers, and the hyphen (-) and underscore (_) characters. Minimum length = 1 Maximum length = 255
responsecontenttype	<String>	Read-write	Default Content-Type header for responses. A Content-Type header can contain 0-255 letters, numbers, and the hyphen (-) and

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
			underscore (_) characters. Minimum length = 1 Maximum length = 255
xmldosaction	<String[]>	Read-write	One or more XML Denial-of-Service (XDoS) actions. Available settings function as follows: * Block - Block connections that violate this security check. * Learn - Use the learning engine to generate a list of exceptions to this security check. * Log - Log violations of this security check. * Stats - Generate statistics for this security check. * None - Disable all actions for this security check. CLI users: To enable one or more actions, type "set appfw profile

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
			-XMLDoSAction" followed by the actions to be enabled. To turn off all actions, type "set appfw profile -XMLDoSAction none". Possible values = none, block, learn, log, stats
xmlformataction	<String[]>	Read-write	One or more XML Format actions. Available settings function as follows: * Block - Block connections that violate this security check. * Log - Log violations of this security check. * Stats - Generate statistics for this security check. * None - Disable all actions for this security check. CLI users: To enable one or more actions, type "set appfw profile -XMLFormatAction"

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
			followed by the actions to be enabled. To turn off all actions, type "set appfw profile -XMLFormatAction none". Possible values = none, block, log, stats
xmlsqlinjectionaction	<String[]>	Read-write	One or more XML SQL Injection actions. Available settings function as follows: * Block - Block connections that violate this security check. * Log - Log violations of this security check. * Stats - Generate statistics for this security check. * None - Disable all actions for this security check. CLI users: To enable one or more actions, type "set appfw profile -XMLSQLInjectionA

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
			ction" followed by the actions to be enabled. To turn off all actions, type "set appfw profile -XMLSQLInjectionAction none". Possible values = none, block, log, stats
xmlsqlinjectiononlycheckfieldswithsqlchars	<String>	Read-write	Check only form fields that contain SQL special characters, which most SQL servers require before accepting an SQL command, for injected SQL. Default value: ON Possible values = ON, OFF
xmlsqlinjectiontype	<String>	Read-write	Available SQL injection types. -SQLSplChar : Checks for SQL Special Chars -SQLKeyword : Checks for SQL Keywords -SQLSplCharANDKeyword : Checks for

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
			both and blocks if both are found -SQLSplCharORKeyword : Checks for both and blocks if anyone is found. Default value: SQLSplCharANDKeyword Possible values = SQLSplChar, SQLKeyword, SQLSplCharORKeyword, SQLSplCharANDKeyword
xmlsqlinjectioncheck sqlwildchars	<String>	Read-write	Check for form fields that contain SQL wild chars . Default value: OFF Possible values = ON, OFF
xmlsqlinjectionparse comments	<String>	Read-write	Parse comments in XML Data and exempt those sections of the request that are from the XML SQL Injection check. You must configure the type of comments that the application firewall is to detect and exempt from this

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
			security check. Available settings function as follows: * Check all - Check all content. * ANSI - Exempt content that is part of an ANSI (Mozilla-style) comment. * Nested - Exempt content that is part of a nested (Microsoft-style) comment. * ANSI Nested - Exempt content that is part of any type of comment. Default value: checkall Possible values = checkall, ansi, nested, ansinested
xmlxssaction	<String[]>	Read-write	One or more XML Cross-Site Scripting actions. Available settings function as follows: * Block - Block connections that violate this security check.

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
			* Log - Log violations of this security check. * Stats - Generate statistics for this security check. * None - Disable all actions for this security check. CLI users: To enable one or more actions, type "set appfw profile -XMLXSSAction" followed by the actions to be enabled. To turn off all actions, type "set appfw profile -XMLXSSAction none". Possible values = none, block, learn, log, stats
xmlwsiaction	<String[]>	Read-write	One or more Web Services Interoperability (WSI) actions. Available settings function as follows: * Block - Block connections that

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
			violate this security check. * Learn - Use the learning engine to generate a list of exceptions to this security check. * Log - Log violations of this security check. * Stats - Generate statistics for this security check. * None - Disable all actions for this security check. CLI users: To enable one or more actions, type "set appfw profile -XMLWSIAction" followed by the actions to be enabled. To turn off all actions, type "set appfw profile -XMLWSIAction none". Possible values = none, block, learn, log, stats

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
xmlattachmentaction	<String[]>	Read-write	One or more XML Attachment actions. Available settings function as follows: * Block - Block connections that violate this security check. * Learn - Use the learning engine to generate a list of exceptions to this security check. * Log - Log violations of this security check. * Stats - Generate statistics for this security check. * None - Disable all actions for this security check. CLI users: To enable one or more actions, type "set appfw profile -XMLAttachmentAction" followed by the actions to be enabled. To turn off all actions, type "set appfw profile

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
			-XMLAttachmentAction none". Possible values = none, block, learn, log, stats
xmlvalidationaction	<String[]>	Read-write	One or more XML Validation actions. Available settings function as follows: * Block - Block connections that violate this security check. * Log - Log violations of this security check. * Stats - Generate statistics for this security check. * None - Disable all actions for this security check. CLI users: To enable one or more actions, type "set appfw profile -XMLValidationAction" followed by the actions to be enabled. To turn off all actions, type "set appfw profile

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
			-XMLValidationAction none". Possible values = none, block, log, stats
xmlerrorobject	<String>	Read-write	Name to assign to the XML Error Object, which the application firewall displays when a user request is blocked. Must begin with a letter, number, or the underscore character \(_\), and must contain only letters, numbers, and the hyphen \(-\), period \(.\) pound \(#\), space \(\), at (@), equals \(=\), colon \(:\), and underscore characters. Cannot be changed after the XML error object is added. The following requirement applies only to the Citrix ADC CLI: If the name includes one or more spaces, enclose the

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
			name in double or single quotation marks \(for example, "my XML error object" or 'my XML error object'\). Minimum length = 1
customsettings	<String>	Read-write	Object name for custom settings. This check is applicable to Profile Type: HTML, XML. . Minimum length = 1
signatures	<String>	Read-write	Object name for signatures. This check is applicable to Profile Type: HTML, XML. . Minimum length = 1
xmlsoapfaultaction	<String[]>	Read-write	One or more XML SOAP Fault Filtering actions. Available settings function as follows: * Block - Block connections that violate this security check. * Log - Log violations of this security check.

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
			* Stats - Generate statistics for this security check. * None - Disable all actions for this security check. * Remove - Remove all violations for this security check. CLI users: To enable one or more actions, type "set appfw profile -XMLSOAPFaultAction" followed by the actions to be enabled. To turn off all actions, type "set appfw profile -XMLSOAPFaultAction none". Possible values = none, block, log, remove, stats
usehtmlerrorobject	<String>	Read-write	Send an imported HTML Error object to a user when a request is blocked, instead of redirecting the user to the designated Error URL. Default value: OFF

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
			Possible values = ON, OFF
errorurl	<String>	Read-write	URL that application firewall uses as the Error URL. Minimum length = 1
htmlerrorobject	<String>	Read-write	Name to assign to the HTML Error Object. Must begin with a letter, number, or the underscore character \(_\), and must contain only letters, numbers, and the hyphen \(-\), period \(.\) pound \(#\), space \(\), at (@), equals \(=\), colon \(:\), and underscore characters. Cannot be changed after the HTML error object is added. The following requirement applies only to the Citrix ADC CLI: If the name includes one or more spaces, enclose the name in double or single quotation

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
			marks \(for example, "my HTML error object" or 'my HTML error object'\). Minimum length = 1
logeverypolicyhit	<String>	Read-write	Log every profile match, regardless of security checks results. Default value: OFF Possible values = ON, OFF
stripcomments	<String>	Read-write	Strip HTML comments. This check is applicable to Profile Type: HTML. . Default value: OFF Possible values = ON, OFF
striphtmlcomments	<String>	Read-write	Strip HTML comments before forwarding a web page sent by a protected web site in response to a user request. Default value: none Possible values = none, all, exclude_script_tag

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
stripxmlcomments	<String>	Read-write	Strip XML comments before forwarding a web page sent by a protected web site in response to a user request. Default value: none Possible values = none, all
exemptclosureurlsfromsecuritychecks	<String>	Read-write	Exempt URLs that pass the Start URL closure check from SQL injection, cross-site script, field format and field consistency security checks at locations other than headers. Default value: ON Possible values = ON, OFF
defaultcharset	<String>	Read-write	Default character set for protected web pages. Web pages sent by your protected web sites in response to user requests are assigned this character set if the page does not already specify a character set. The character sets supported by the

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
			application firewall are: * iso-8859-1 (English US) * big5 (Chinese Traditional) * gb2312 (Chinese Simplified) * sjis (Japanese Shift-JIS) * euc-jp (Japanese EUC-JP) * iso-8859-9 (Turkish) * utf-8 (Unicode) * euc-kr (Korean). Minimum length = 1 Maximum length = 31
postbodylimit	<Double>	Read-write	Maximum allowed HTTP post body size, in bytes. Default value: 20000000
postbodylimitsignature	<Double>	Read-write	Maximum allowed HTTP post body size for signature inspection for location HTTP_POST_BODY

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
			in the signatures, in bytes. Default value: 8096
fileuploadmaxnum	<Double>	Read-write	Maximum allowed number of file uploads per form-submission request. The maximum setting (65535) allows an unlimited number of uploads. Default value: 65535 Minimum value = 0 Maximum value = 65535
canonicalizehtmlresponse	<String>	Read-write	Perform HTML entity encoding for any special characters in responses sent by your protected web sites. Default value: ON Possible values = ON, OFF
enableformtagging	<String>	Read-write	Enable tagging of web form fields for use by the Form Field Consistency and CSRF Form Tagging checks. Default value: ON

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
			Possible values = ON, OFF
sessionlessfieldconsistency	<String>	Read-write	Perform sessionless Field Consistency Checks. Default value: OFF Possible values = OFF, ON, postOnly
sessionlessurlclosure	<String>	Read-write	Enable session less URL Closure Checks. This check is applicable to Profile Type: HTML. . Default value: OFF Possible values = ON, OFF
semicolonfieldseparator	<String>	Read-write	Allow ';' as a form field separator in URL queries and POST form bodies. . Default value: OFF Possible values = ON, OFF
excludefileuploadfromchecks	<String>	Read-write	Exclude uploaded files from Form checks. Default value: OFF Possible values = ON, OFF

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
sqlinjectionparsecomments	<String>	Read-write	Parse HTML comments and exempt them from the HTML SQL Injection check. You must specify the type of comments that the application firewall is to detect and exempt from this security check. Available settings function as follows: * Check all - Check all content. * ANSI - Exempt content that is part of an ANSI (Mozilla-style) comment. * Nested - Exempt content that is part of a nested (Microsoft-style) comment. * ANSI Nested - Exempt content that is part of any type of comment. Possible values = checkall, ansi, nested, ansinested
invalidpercenthandling	<String>	Read-write	Configure the method that the application firewall

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
			uses to handle percent-encoded names and values. Available settings function as follows: * apache_mode - Apache format. * asp_mode - Microsoft ASP format. * secure_mode - Secure format. Default value: secure_mode Possible values = apache_mode, asp_mode, secure_mode
type	<String[]>	Read-write	Application firewall profile type, which controls which security checks and settings are applied to content that is filtered with the profile. Available settings function as follows: * HTML - HTML-based web sites. * XML - XML-based web sites and services.

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
			* HTML XML (Web 2.0) - Sites that contain both HTML and XML content, such as ATOM feeds, blogs, and RSS feeds. Default value: HTML Possible values = HTML, XML
checkrequestheaders	<String>	Read-write	Check request headers as well as web forms for injected SQL and cross-site scripts. Default value: OFF Possible values = ON, OFF
optimizepartialreqs	<String>	Read-write	Optimize handle of HTTP partial requests i.e. those with range headers. Available settings are as follows: * ON - Partial requests by the client result in partial requests to the backend server in most cases.

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
			* OFF - Partial requests by the client are changed to full requests to the backend server. Default value: ON Possible values = ON, OFF
urldecoderequestcookies	<String>	Read-write	URL Decode request cookies before subjecting them to SQL and cross-site scripting checks. Default value: OFF Possible values = ON, OFF
comment	<String>	Read-write	Any comments about the purpose of profile, or other useful information about the profile.
percentdecoderecursively	<String>	Read-write	Configure whether the application firewall should use percentage recursive decoding. Default value: ON Possible values = ON, OFF
multipleheaderaction	<String[]>	Read-write	One or more multiple header actions.

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
			Available settings function as follows: * Block - Block connections that have multiple headers. * Log - Log connections that have multiple headers. * KeepLast - Keep only last header when multiple headers are present. CLI users: To enable one or more actions, type "set appfw profile -multipleHeaderAction" followed by the actions to be enabled. Possible values = block, keepLast, log, none
rfcprofile	<String>	Read-write	Object name of the rfc profile. Minimum length = 1
archivename	<String>	Read-write	Source for tar archive. Minimum length = 1

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
			Maximum length = 31
state	<String>	Read-only	Enabled. Possible values = ENABLED, DISABLED
csrftag	<String>	Read-only	The web form originating URL.
builtin_depr	<Boolean>	Read-only	Indicates that a profile is a built-in entity.
__count	<Double>	Read-only	count parameter

Operations

(click to see Properties)

ADD| DELETE| UPDATE| UNSET| RESTORE| GET (ALL)| GET| COUNT

Some options that you can use for each operations:

- **Getting warnings in response:**NITRO allows you to get warnings in an operation by specifying the "warning" query parameter as "yes". For example, to get warnings while connecting to the Citrix ADC appliance, the URL is as follows:

  http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/login?warning=yes

  If any, the warnings are displayed in the response payload with the HTTP code "209 X-NITRO-WARNING".

- **Authenticated access for individual NITRO operations:**NITRO allows you to logon to the Citrix ADC appliance to perform individual operations. You can use this option instead of creating a NITRO session (using the login object) and then using that session to perform all operations,

  To do this, you must specify the username and password in the request header of the NITRO request as follows:

X-NITRO-USER:<username>

X-NITRO-PASS:<password>

Note: In such cases, make sure that the request header DOES not include the following:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue>

`*Note:` *

Mandatory parameters are marked in redand placeholder content is marked in <green>.

add

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/appfwprofile

HTTP Method:POST

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Content-Type:application/json

Request Payload:

```
{"appfwprofile":{

<b>"name":<String_value>,

</b>"defaults":<String_value>,

"starturlaction":<String[]_value>,

"contenttypeaction":<String[]_value>,

"inspectcontenttypes":<String[]_value>,

"starturlclosure":<String_value>,

"denyurlaction":<String[]_value>,

"refererheadercheck":<String_value>,

"cookieconsistencyaction":<String[]_value>,
```

```
"cookietransforms":<String_value>,

"cookieencryption":<String_value>,

"cookieproxying":<String_value>,

"addcookieflags":<String_value>,

"fieldconsistencyaction":<String[]_value>,

"csrftagaction":<String[]_value>,

"crosssitescriptingaction":<String[]_value>,

"crosssitescriptingtransformunsafehtml":<String_value>,

"crosssitescriptingcheckcompleteurls":<String_value>,

"sqlinjectionaction":<String[]_value>,

"sqlinjectiontransformspecialchars":<String_value>,

"sqlinjectiononlycheckfieldswithsqlchars":<String_value>,

"sqlinjectiontype":<String_value>,

"sqlinjectionchecksqlwildchars":<String_value>,

"fieldformataction":<String[]_value>,

"defaultfieldformattype":<String_value>,

"defaultfieldformatminlength":<Double_value>,

"defaultfieldformatmaxlength":<Double_value>,

"bufferoverflowaction":<String[]_value>,

"bufferoverflowmaxurllength":<Double_value>,

"bufferoverflowmaxheaderlength":<Double_value>,

"bufferoverflowmaxcookielength":<Double_value>,
```

```
"creditcardaction":<String[]_value>,

"creditcard":<String[]_value>,

"creditcardmaxallowed":<Double_value>,

"creditcardxout":<String_value>,

"dosecurecreditcardlogging":<String_value>,

"streaming":<String_value>,

"trace":<String_value>,

"requestcontenttype":<String_value>,

"responsecontenttype":<String_value>,

"xmldosaction":<String[]_value>,

"xmlformataction":<String[]_value>,

"xmlsqlinjectionaction":<String[]_value>,

"xmlsqlinjectiononlycheckfieldswithsqlchars":<String_value>,

"xmlsqlinjectiontype":<String_value>,

"xmlsqlinjectionchecksqlwildchars":<String_value>,

"xmlsqlinjectionparsecomments":<String_value>,

"xmlxssaction":<String[]_value>,

"xmlwsiaction":<String[]_value>,

"xmlattachmentaction":<String[]_value>,

"xmlvalidationaction":<String[]_value>,

"xmlerrorobject":<String_value>,

"customsettings":<String_value>,
```

```
"signatures":<String_value>,

"xmlsoapfaultaction":<String[]_value>,

"usehtmlerrorobject":<String_value>,

"errorurl":<String_value>,

"htmlerrorobject":<String_value>,

"logeverypolicyhit":<String_value>,

"stripcomments":<String_value>,

"striphtmlcomments":<String_value>,

"stripxmlcomments":<String_value>,

"exemptclosureurlsfromsecuritychecks":<String_value>,

"defaultcharset":<String_value>,

"postbodylimit":<Double_value>,

"postbodylimitsignature":<Double_value>,

"fileuploadmaxnum":<Double_value>,

"canonicalizehtmlresponse":<String_value>,

"enableformtagging":<String_value>,

"sessionlessfieldconsistency":<String_value>,

"sessionlessurlclosure":<String_value>,

"semicolonfieldseparator":<String_value>,

"excludefileuploadfromchecks":<String_value>,

"sqlinjectionparsecomments":<String_value>,

"invalidpercenthandling":<String_value>,
```

```
"type":<String[]_value>,

"checkrequestheaders":<String_value>,

"optimizepartialreqs":<String_value>,

"urldecoderequestcookies":<String_value>,

"comment":<String_value>,

"percentdecoderecursively":<String_value>,

"multipleheaderaction":<String[]_value>,

"rfcprofile":<String_value>

}}
```

Response:

HTTP Status Code on Success: 201 Created HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error

delete

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/appfwprofile/name_value<String>

HTTP Method:DELETE

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue>

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error

update

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/appfwprofile

HTTP Method:PUT

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Content-Type:application/json

Request Payload:

```
{"appfwprofile":{

<b>"name":<String_value>,

</b>"starturlaction":<String[]_value>,

"contenttypeaction":<String[]_value>,

"inspectcontenttypes":<String[]_value>,

"starturlclosure":<String_value>,

"denyurlaction":<String[]_value>,

"refererheadercheck":<String_value>,

"cookieconsistencyaction":<String[]_value>,

"cookietransforms":<String_value>,

"cookieencryption":<String_value>,

"cookieproxying":<String_value>,

"addcookieflags":<String_value>,

"fieldconsistencyaction":<String[]_value>,

"csrftagaction":<String[]_value>,

"crosssitescriptingaction":<String[]_value>,

"crosssitescriptingtransformunsafehtml":<String_value>,
```

```
"crosssitescriptingcheckcompleteurls":<String_value>,

"sqlinjectionaction":<String[]_value>,

"sqlinjectiontransformspecialchars":<String_value>,

"sqlinjectiononlycheckfieldswithsqlchars":<String_value>,

"sqlinjectiontype":<String_value>,

"sqlinjectionchecksqlwildchars":<String_value>,

"fieldformataction":<String[]_value>,

"defaultfieldformattype":<String_value>,

"defaultfieldformatminlength":<Double_value>,

"defaultfieldformatmaxlength":<Double_value>,

"bufferoverflowaction":<String[]_value>,

"bufferoverflowmaxurllength":<Double_value>,

"bufferoverflowmaxheaderlength":<Double_value>,

"bufferoverflowmaxcookielength":<Double_value>,

"creditcardaction":<String[]_value>,

"creditcard":<String[]_value>,

"creditcardmaxallowed":<Double_value>,

"creditcardxout":<String_value>,

"dosecurecreditcardlogging":<String_value>,

"streaming":<String_value>,

"trace":<String_value>,

"requestcontenttype":<String_value>,
```

```
"responsecontenttype":<String_value>,

"xmldosaction":<String[]_value>,

"xmlformataction":<String[]_value>,

"xmlsqlinjectionaction":<String[]_value>,

"xmlsqlinjectiononlycheckfieldswithsqlchars":<String_value>,

"xmlsqlinjectiontype":<String_value>,

"xmlsqlinjectionchecksqlwildchars":<String_value>,

"xmlsqlinjectionparsecomments":<String_value>,

"xmlxssaction":<String[]_value>,

"xmlwsiaction":<String[]_value>,

"xmlattachmentaction":<String[]_value>,

"xmlvalidationaction":<String[]_value>,

"xmlerrorobject":<String_value>,

"customsettings":<String_value>,

"signatures":<String_value>,

"xmlsoapfaultaction":<String[]_value>,

"usehtmlerrorobject":<String_value>,

"errorurl":<String_value>,

"htmlerrorobject":<String_value>,

"logeverypolicyhit":<String_value>,

"stripcomments":<String_value>,

"striphtmlcomments":<String_value>,
```

```
"stripxmlcomments":<String_value>,

"exemptclosureurlsfromsecuritychecks":<String_value>,

"defaultcharset":<String_value>,

"postbodylimit":<Double_value>,

"postbodylimitsignature":<Double_value>,

"fileuploadmaxnum":<Double_value>,

"canonicalizehtmlresponse":<String_value>,

"enableformtagging":<String_value>,

"sessionlessfieldconsistency":<String_value>,

"sessionlessurlclosure":<String_value>,

"semicolonfieldseparator":<String_value>,

"excludefileuploadfromchecks":<String_value>,

"sqlinjectionparsecomments":<String_value>,

"invalidpercenthandling":<String_value>,

"type":<String[]_value>,

"checkrequestheaders":<String_value>,

"optimizepartialreqs":<String_value>,

"urldecoderequestcookies":<String_value>,

"comment":<String_value>,

"percentdecoderecursively":<String_value>,

"multipleheaderaction":<String[]_value>,

"rfcprofile":<String_value>
```

```
}}
```

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error

unset

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/appfwprofile?**action=unset**

HTTP Method:POST

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Content-Type:application/json

Request Payload:

```
{"appfwprofile":{

<b>"name":<String_value>,

</b>"starturlaction":true,

"contenttypeaction":true,

"inspectcontenttypes":true,

"starturlclosure":true,

"denyurlaction":true,

"refererheadercheck":true,

"cookieconsistencyaction":true,

"cookietransforms":true,
```

```
"cookieencryption":true,

"cookieproxying":true,

"addcookieflags":true,

"fieldconsistencyaction":true,

"csrftagaction":true,

"crosssitescriptingaction":true,

"crosssitescriptingtransformunsafehtml":true,

"crosssitescriptingcheckcompleteurls":true,

"sqlinjectionaction":true,

"sqlinjectiontransformspecialchars":true,

"sqlinjectiononlycheckfieldswithsqlchars":true,

"sqlinjectiontype":true,

"sqlinjectionchecksqlwildchars":true,

"fieldformataction":true,

"defaultfieldformattype":true,

"defaultfieldformatminlength":true,

"defaultfieldformatmaxlength":true,

"bufferoverflowaction":true,

"bufferoverflowmaxurllength":true,

"bufferoverflowmaxheaderlength":true,

"bufferoverflowmaxcookielength":true,

"creditcardaction":true,
```

```
"creditcard":true,

"creditcardmaxallowed":true,

"creditcardxout":true,

"dosecurecreditcardlogging":true,

"streaming":true,

"trace":true,

"requestcontenttype":true,

"responsecontenttype":true,

"xmldosaction":true,

"xmlformataction":true,

"xmlsqlinjectionaction":true,

"xmlsqlinjectiononlycheckfieldswithsqlchars":true,

"xmlsqlinjectiontype":true,

"xmlsqlinjectionchecksqlwildchars":true,

"xmlsqlinjectionparsecomments":true,

"xmlxssaction":true,

"xmlwsiaction":true,

"xmlattachmentaction":true,

"xmlvalidationaction":true,

"xmlerrorobject":true,

"customsettings":true,

"signatures":true,
```

```
"xmlsoapfaultaction":true,

"usehtmlerrorobject":true,

"errorurl":true,

"htmlerrorobject":true,

"logeverypolicyhit":true,

"stripcomments":true,

"striphtmlcomments":true,

"stripxmlcomments":true,

"exemptclosureurlsfromsecuritychecks":true,

"defaultcharset":true,

"postbodylimit":true,

"postbodylimitsignature":true,

"fileuploadmaxnum":true,

"canonicalizehtmlresponse":true,

"enableformtagging":true,

"sessionlessfieldconsistency":true,

"sessionlessurlclosure":true,

"semicolonfieldseparator":true,

"excludefileuploadfromchecks":true,

"sqlinjectionparsecomments":true,

"invalidpercenthandling":true,

"type":true,
```

```
"checkrequestheaders":true,

"optimizepartialreqs":true,

"urldecoderequestcookies":true,

"comment":true,

"percentdecoderecursively":true,

"multipleheaderaction":true,

"rfcprofile":true

}}
```

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error

restore

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/appfwprofile?**action=restore**

HTTP Method:POST

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Content-Type:application/json

Request Payload:

```
{"appfwprofile":{

<b>"archivename":<String_value>

</b>}}
```

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error

get (all)

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/appfwprofile

Query-parameters:

attrs

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/appfwprofile?**attrs=property-name1,property-name2**

Use this query parameter to specify the resource details that you want to retrieve.

filter

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/appfwprofile?**filter=property-name1:property-val1,property-name2:property-val2**

Use this query-parameter to get the filtered set of appfwprofile resources configured on Citrix ADC. Filtering can be done on any of the properties of the resource.

view

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/appfwprofile?**view=summary**

Note: By default, the retrieved results are displayed in detail view (?view=detail).

pagination

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/appfwprofile?**pagesize=#no;pageno=#no**

Use this query-parameter to get the appfwprofile resources in chunks.

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "appfwprofile": [ {

"name":<String_value>,

"type":<String[]_value>,

"state":<String_value>,

"defaults":<String_value>,

"usehtmlerrorobject":<String_value>,

"errorurl":<String_value>,

"htmlerrorobject":<String_value>,

"logeverypolicyhit":<String_value>,

"stripcomments":<String_value>,

"striphtmlcomments":<String_value>,

"stripxmlcomments":<String_value>,

"defaultcharset":<String_value>,

"postbodylimit":<Double_value>,

"postbodylimitsignature":<Double_value>,

"fileuploadmaxnum":<Double_value>,
```

```
"canonicalizehtmlresponse":<String_value>,

"enableformtagging":<String_value>,

"sessionlessfieldconsistency":<String_value>,

"sessionlessurlclosure":<String_value>,

"semicolonfieldseparator":<String_value>,

"excludefileuploadfromchecks":<String_value>,

"sqlinjectionparsecomments":<String_value>,

"checkrequestheaders":<String_value>,

"optimizepartialreqs":<String_value>,

"urldecoderequestcookies":<String_value>,

"starturlaction":<String[]_value>,

"contenttypeaction":<String[]_value>,

"inspectcontenttypes":<String[]_value>,

"starturlclosure":<String_value>,

"denyurlaction":<String[]_value>,

"refererheadercheck":<String_value>,

"csrftagaction":<String[]_value>,

"csrftag":<String_value>,

"crosssitescriptingaction":<String[]_value>,

"crosssitescriptingtransformunsafehtml":<String_value>,

"crosssitescriptingcheckcompleteurls":<String_value>,

"exemptclosureurlsfromsecuritychecks":<String_value>,
```

```
"sqlinjectionaction":<String[]_value>,

"sqlinjectiontransformspecialchars":<String_value>,

"sqlinjectiononlycheckfieldswithsqlchars":<String_value>,

"sqlinjectiontype":<String_value>,

"sqlinjectionchecksqlwildchars":<String_value>,

"invalidpercenthandling":<String_value>,

"fieldconsistencyaction":<String[]_value>,

"cookieconsistencyaction":<String[]_value>,

"cookietransforms":<String_value>,

"cookieencryption":<String_value>,

"cookieproxying":<String_value>,

"addcookieflags":<String_value>,

"bufferoverflowaction":<String[]_value>,

"bufferoverflowmaxurllength":<Double_value>,

"bufferoverflowmaxheaderlength":<Double_value>,

"bufferoverflowmaxcookielength":<Double_value>,

"fieldformataction":<String[]_value>,

"defaultfieldformattype":<String_value>,

"defaultfieldformatminlength":<Double_value>,

"defaultfieldformatmaxlength":<Double_value>,

"creditcardaction":<String[]_value>,

"creditcard":<String[]_value>,
```

```
"creditcardmaxallowed":<Double_value>,

"creditcardxout":<String_value>,

"dosecurecreditcardlogging":<String_value>,

"streaming":<String_value>,

"trace":<String_value>,

"requestcontenttype":<String_value>,

"responsecontenttype":<String_value>,

"xmlerrorobject":<String_value>,

"signatures":<String_value>,

"xmlformataction":<String[]_value>,

"xmldosaction":<String[]_value>,

"xmlsqlinjectionaction":<String[]_value>,

"xmlsqlinjectiononlycheckfieldswithsqlchars":<String_value>,

"xmlsqlinjectiontype":<String_value>,

"xmlsqlinjectionchecksqlwildchars":<String_value>,

"xmlsqlinjectionparsecomments":<String_value>,

"xmlxssaction":<String[]_value>,

"xmlwsiaction":<String[]_value>,

"xmlattachmentaction":<String[]_value>,

"xmlvalidationaction":<String[]_value>,

"xmlsoapfaultaction":<String[]_value>,

"builtin_depr":<Boolean_value>,
```

```
"comment":<String_value>,

"percentdecoderecursively":<String_value>,

"multipleheaderaction":<String[]_value>,

"rfcprofile":<String_value>

}]}
```

get

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/appfwprofile/name_value<String>

Query-parameters:

attrs

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/appfwprofile/name_value<String>?**attrs=property-name1,property-name2**

Use this query parameter to specify the resource details that you want to retrieve.

view

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/appfwprofile/name_value<String>?**view=summary**

Note: By default, the retrieved results are displayed in detail view (?view=detail).

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "appfwprofile": [ {

"name":<String_value>,

"type":<String[]_value>,

"state":<String_value>,

"defaults":<String_value>,

"usehtmlerrorobject":<String_value>,

"errorurl":<String_value>,

"htmlerrorobject":<String_value>,

"logeverypolicyhit":<String_value>,

"stripcomments":<String_value>,

"striphtmlcomments":<String_value>,

"stripxmlcomments":<String_value>,

"defaultcharset":<String_value>,

"postbodylimit":<Double_value>,

"postbodylimitsignature":<Double_value>,

"fileuploadmaxnum":<Double_value>,

"canonicalizehtmlresponse":<String_value>,

"enableformtagging":<String_value>,

"sessionlessfieldconsistency":<String_value>,

"sessionlessurlclosure":<String_value>,
```

```
"semicolonfieldseparator":<String_value>,

"excludefileuploadfromchecks":<String_value>,

"sqlinjectionparsecomments":<String_value>,

"checkrequestheaders":<String_value>,

"optimizepartialreqs":<String_value>,

"urldecoderequestcookies":<String_value>,

"starturlaction":<String[]_value>,

"contenttypeaction":<String[]_value>,

"inspectcontenttypes":<String[]_value>,

"starturlclosure":<String_value>,

"denyurlaction":<String[]_value>,

"refererheadercheck":<String_value>,

"csrftagaction":<String[]_value>,

"csrftag":<String_value>,

"crosssitescriptingaction":<String[]_value>,

"crosssitescriptingtransformunsafehtml":<String_value>,

"crosssitescriptingcheckcompleteurls":<String_value>,

"exemptclosureurlsfromsecuritychecks":<String_value>,

"sqlinjectionaction":<String[]_value>,

"sqlinjectiontransformspecialchars":<String_value>,

"sqlinjectiononlycheckfieldswithsqlchars":<String_value>,

"sqlinjectiontype":<String_value>,
```

```
"sqlinjectionchecksqlwildchars":<String_value>,

"invalidpercenthandling":<String_value>,

"fieldconsistencyaction":<String[]_value>,

"cookieconsistencyaction":<String[]_value>,

"cookietransforms":<String_value>,

"cookieencryption":<String_value>,

"cookieproxying":<String_value>,

"addcookieflags":<String_value>,

"bufferoverflowaction":<String[]_value>,

"bufferoverflowmaxurllength":<Double_value>,

"bufferoverflowmaxheaderlength":<Double_value>,

"bufferoverflowmaxcookielength":<Double_value>,

"fieldformataction":<String[]_value>,

"defaultfieldformattype":<String_value>,

"defaultfieldformatminlength":<Double_value>,

"defaultfieldformatmaxlength":<Double_value>,

"creditcardaction":<String[]_value>,

"creditcard":<String[]_value>,

"creditcardmaxallowed":<Double_value>,

"creditcardxout":<String_value>,

"dosecurecreditcardlogging":<String_value>,

"streaming":<String_value>,
```

```
"trace":<String_value>,

"requestcontenttype":<String_value>,

"responsecontenttype":<String_value>,

"xmlerrorobject":<String_value>,

"signatures":<String_value>,

"xmlformataction":<String[]_value>,

"xmldosaction":<String[]_value>,

"xmlsqlinjectionaction":<String[]_value>,

"xmlsqlinjectiononlycheckfieldswithsqlchars":<String_value>,

"xmlsqlinjectiontype":<String_value>,

"xmlsqlinjectionchecksqlwildchars":<String_value>,

"xmlsqlinjectionparsecomments":<String_value>,

"xmlxssaction":<String[]_value>,

"xmlwsiaction":<String[]_value>,

"xmlattachmentaction":<String[]_value>,

"xmlvalidationaction":<String[]_value>,

"xmlsoapfaultaction":<String[]_value>,

"builtin_depr":<Boolean_value>,

"comment":<String_value>,

"percentdecoderecursively":<String_value>,

"multipleheaderaction":<String[]_value>,

"rfcprofile":<String_value>
```

```
}]}
```

count

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/appfwprofile?**count=yes**

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "appfwprofile": [ { "__count": "#no"} ] }
```

appfwprofile_binding

Binding object which returns the resources bound to appfwprofile.

Properties

(click to see Operations)

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
name	<String>	Read-write	Name of the application firewall profile. Minimum length = 1

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
appfwprofile_xmlattachmenturl_binding	<appfwprofile_xmlattachmenturl_binding[]>	Read-only	xmlattachmenturl that can be bound to appfwprofile.
appfwprofile_excluderescontenttype_binding	<appfwprofile_excluderescontenttype_binding[]>	Read-only	excluderescontenttype that can be bound to appfwprofile.
appfwprofile_crosssitescripting_binding	<appfwprofile_crosssitescripting_binding[]>	Read-only	crosssitescripting that can be bound to appfwprofile.
appfwprofile_xmlwsiurl_binding	<appfwprofile_xmlwsiurl_binding[]>	Read-only	xmlwsiurl that can be bound to appfwprofile.
appfwprofile_xmldosurl_binding	<appfwprofile_xmldosurl_binding[]>	Read-only	xmldosurl that can be bound to appfwprofile.
appfwprofile_fieldconsistency_binding	<appfwprofile_fieldconsistency_binding[]>	Read-only	fieldconsistency that can be bound to appfwprofile.
appfwprofile_sqlinjection_binding	<appfwprofile_sqlinjection_binding[]>	Read-only	sqlinjection that can be bound to appfwprofile.
appfwprofile_safeobject_binding	<appfwprofile_safeobject_binding[]>	Read-only	safeobject that can be bound to appfwprofile.
appfwprofile_logexpression_binding	<appfwprofile_logexpression_binding[]>	Read-only	logexpression that can be bound to appfwprofile.
appfwprofile_denyurl_binding	<appfwprofile_denyurl_binding[]>	Read-only	denyurl that can be bound to appfwprofile.

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
appfwprofile_trustedlearningclients_binding	<appfwprofile_trustedlearningclients_binding[]>	Read-only	trustedlearningclients that can be bound to appfwprofile.
appfwprofile_csrftag_binding	<appfwprofile_csrftag_binding[]>	Read-only	csrftag that can be bound to appfwprofile.
appfwprofile_starturl_binding	<appfwprofile_starturl_binding[]>	Read-only	starturl that can be bound to appfwprofile.
appfwprofile_cookieconsistency_binding	<appfwprofile_cookieconsistency_binding[]>	Read-only	cookieconsistency that can be bound to appfwprofile.
appfwprofile_xmlsqlinjection_binding	<appfwprofile_xmlsqlinjection_binding[]>	Read-only	xmlsqlinjection that can be bound to appfwprofile.
appfwprofile_creditcardnumber_binding	<appfwprofile_creditcardnumber_binding[]>	Read-only	creditcardnumber that can be bound to appfwprofile.
appfwprofile_xmlxss_binding	<appfwprofile_xmlxss_binding[]>	Read-only	xmlxss that can be bound to appfwprofile.
appfwprofile_fieldformat_binding	<appfwprofile_fieldformat_binding[]>	Read-only	fieldformat that can be bound to appfwprofile.
appfwprofile_contenttype_binding	<appfwprofile_contenttype_binding[]>	Read-only	contenttype that can be bound to appfwprofile.
appfwprofile_xmlvalidationurl_binding	<appfwprofile_xmlvalidationurl_binding[]>	Read-only	xmlvalidationurl that can be bound to appfwprofile.

Operations

(click to see Properties)

GET| GET (ALL)

Some options that you can use for each operations:

- **Getting warnings in response:**NITRO allows you to get warnings in an operation by specifying the "warning" query parameter as "yes". For example, to get warnings while connecting to the Citrix ADC appliance, the URL is as follows:

  http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/login?warning=yes

  If any, the warnings are displayed in the response payload with the HTTP code "209 X-NITRO-WARNING".

- **Authenticated access for individual NITRO operations:**NITRO allows you to logon to the Citrix ADC appliance to perform individual operations. You can use this option instead of creating a NITRO session (using the login object) and then using that session to perform all operations,

  To do this, you must specify the username and password in the request header of the NITRO request as follows:

  X-NITRO-USER:<username>

  X-NITRO-PASS:<password>

  Note: In such cases, make sure that the request header DOES not include the following:

  Cookie:NITRO_AUTH_TOKEN=<tokenvalue>

`*Note:`*

Mandatory parameters are marked in redand placeholder content is marked in <green>.

get

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/appfwprofile_binding/name_value<String>

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "appfwprofile_binding": [ {

"name":<String_value>,

"appfwprofile_csrftag_binding":<appfwprofile_csrftag_binding[]_value>,

"appfwprofile_starturl_binding":<appfwprofile_starturl_binding[]_value>,

"appfwprofile_xmlvalidationurl_binding":<appfwprofile_xmlvalidationurl_bin
ding[]_value>,

"appfwprofile_logexpression_binding":<appfwprofile_logexpression_binding[]
_value>,

"appfwprofile_xmlattachmenturl_binding":<appfwprofile_xmlattachmenturl_bin
ding[]_value>,

"appfwprofile_safeobject_binding":<appfwprofile_safeobject_binding[]_value
>,

"appfwprofile_cookieconsistency_binding":<appfwprofile_cookieconsistency_b
inding[]_value>,

"appfwprofile_fieldconsistency_binding":<appfwprofile_fieldconsistency_bin
ding[]_value>,

"appfwprofile_xmlwsiurl_binding":<appfwprofile_xmlwsiurl_binding[]_value>,

"appfwprofile_xmlsqlinjection_binding":<appfwprofile_xmlsqlinjection_bindi
```

```
ng[]_value>,

"appfwprofile_sqlinjection_binding":<appfwprofile_sqlinjection_binding[]_v
alue>,

"appfwprofile_trustedlearningclients_binding":<appfwprofile_trustedlearnin
gclients_binding[]_value>,

"appfwprofile_crosssitescripting_binding":<appfwprofile_crosssitescripting
_binding[]_value>,

"appfwprofile_excluderescontenttype_binding":<appfwprofile_excluderesconte
nttype_binding[]_value>,

"appfwprofile_contenttype_binding":<appfwprofile_contenttype_binding[]_val
ue>,

"appfwprofile_denyurl_binding":<appfwprofile_denyurl_binding[]_value>,

"appfwprofile_xmlxss_binding":<appfwprofile_xmlxss_binding[]_value>,

"appfwprofile_fieldformat_binding":<appfwprofile_fieldformat_binding[]_val
ue>,

"appfwprofile_creditcardnumber_binding":<appfwprofile_creditcardnumber_bin
ding[]_value>,

"appfwprofile_xmldosurl_binding":<appfwprofile_xmldosurl_binding[]_value>

}]}
```

get (all)

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/appfwprofile_binding

Query-parameters:

bulkbindings

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/appfwprofile_binding?**bulkbindings=yes**

NITRO allows you to fetch bindings in bulk.

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "appfwprofile_binding": [ {

"name":<String_value>,

"appfwprofile_csrftag_binding":<appfwprofile_csrftag_binding[]_value>,

"appfwprofile_starturl_binding":<appfwprofile_starturl_binding[]_value>,

"appfwprofile_xmlvalidationurl_binding":<appfwprofile_xmlvalidationurl_bin
ding[]_value>,

"appfwprofile_logexpression_binding":<appfwprofile_logexpression_binding[]
_value>,

"appfwprofile_xmlattachmenturl_binding":<appfwprofile_xmlattachmenturl_bin
ding[]_value>,

"appfwprofile_safeobject_binding":<appfwprofile_safeobject_binding[]_value
>,

"appfwprofile_cookieconsistency_binding":<appfwprofile_cookieconsistency_b
inding[]_value>,

"appfwprofile_fieldconsistency_binding":<appfwprofile_fieldconsistency_bin
ding[]_value>,

"appfwprofile_xmlwsiurl_binding":<appfwprofile_xmlwsiurl_binding[]_value>,
```

```
"appfwprofile_xmlsqlinjection_binding":<appfwprofile_xmlsqlinjection_bindi
ng[]_value>,

"appfwprofile_sqlinjection_binding":<appfwprofile_sqlinjection_binding[]_v
alue>,

"appfwprofile_trustedlearningclients_binding":<appfwprofile_trustedlearnin
gclients_binding[]_value>,

"appfwprofile_crosssitescripting_binding":<appfwprofile_crosssitescripting
_binding[]_value>,

"appfwprofile_excluderescontenttype_binding":<appfwprofile_excluderesconte
nttype_binding[]_value>,

"appfwprofile_contenttype_binding":<appfwprofile_contenttype_binding[]_val
ue>,

"appfwprofile_denyurl_binding":<appfwprofile_denyurl_binding[]_value>,

"appfwprofile_xmlxss_binding":<appfwprofile_xmlxss_binding[]_value>,

"appfwprofile_fieldformat_binding":<appfwprofile_fieldformat_binding[]_val
ue>,

"appfwprofile_creditcardnumber_binding":<appfwprofile_creditcardnumber_bin
ding[]_value>,

"appfwprofile_xmldosurl_binding":<appfwprofile_xmldosurl_binding[]_value>

}]}
```

appfwprofile_contenttype_binding

Binding object showing the contenttype that can be bound to appfwprofile.

Properties

(click to see Operations)

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
state	<String>	Read-write	Enabled. Possible values = ENABLED, DISABLED
name	<String>	Read-write	Name of the profile to which to bind an exemption or rule. Minimum length = 1
contenttype	<String>	Read-write	A regular expression that designates a content-type on the content-types list.
comment	<String>	Read-write	Any comments about the purpose of profile, or other useful information about the profile.
__count	<Double>	Read-write	count parameter

Operations

(click to see Properties)

ADD:| DELETE:| GET| GET (ALL)| COUNT

Some options that you can use for each operations:

- **Getting warnings in response:**NITRO allows you to get warnings in an operation by specifying the "warning" query parameter as "yes". For example, to get warnings while connecting to the Citrix ADC appliance, the URL is as follows:

  http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/login?warning=yes

  If any, the warnings are displayed in the response payload with the HTTP code "209 X-NITRO-WARNING".

- **Authenticated access for individual NITRO operations:**NITRO allows you to logon to the Citrix ADC appliance to perform individual operations. You can use this option instead of creating a NITRO session (using the login object) and then using that session to perform all operations,

  To do this, you must specify the username and password in the request header of the NITRO request as follows:

  X-NITRO-USER:<username>

  X-NITRO-PASS:<password>

  Note: In such cases, make sure that the request header DOES not include the following:

  Cookie:NITRO_AUTH_TOKEN=<tokenvalue>

`*Note:`*

Mandatory parameters are marked in redand placeholder content is marked in <green>.

add

URL:http://<Citrix-ADC-IP-address(NSIP)/nitro/v1/config/appfwprofile_contenttype_binding

HTTP Method:PUT

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Content-Type:application/json

Request Payload:

```
{

"appfwprofile_contenttype_binding":{

<b>"name":<String_value>,

</b>"comment":<String_value>,

"state":<String_value>,

"contenttype":<String_value>
```

```
}}
```

Response:

HTTP Status Code on Success: 201 Created HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error

delete

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/appfwprofile_contenttype_binding/name_value<String>

Query-parameters:

args

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/appfwprofile_contenttype_binding/name_value<String>?**args=contenttype:<String_value>**

HTTP Method:DELETE

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue>

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error

get

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/appfwprofile_contenttype_binding/name_value<String>

Query-parameters:

filter

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/appfwprofile_contenttype_binding/name_value<String>?**filter=property-name1:property-value1,property-name2:property-value2**

Use this query-parameter to get the filtered set of appfwprofile_contenttype_binding resources configured on Citrix ADC. Filtering can be done on any of the properties of the resource.

pagination

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/appfwprofile_contenttype_binding/name_value<String>?**pagesize=#no;pageno=#no**

Use this query-parameter to get the appfwprofile_contenttype_binding resources in chunks.

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "appfwprofile_contenttype_binding": [ {

"state":<String_value>,

"name":<String_value>,

"contenttype":<String_value>,

"comment":<String_value>

}]}
```

get (all)

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/appfwprofile_contenttype_binding

Query-parameters:

bulkbindings

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/appfwprofile_contenttype_binding?**bulkbindings=yes**

NITRO allows you to fetch bindings in bulk.

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "appfwprofile_contenttype_binding": [ {

"state":<String_value>,

"name":<String_value>,

"contenttype":<String_value>,

"comment":<String_value>

}]}
```

count

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/appfwprofile_contenttype_binding/name_value<String>?**count=yes**

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{"appfwprofile_contenttype_binding": [ { "__count": "#no"} ] }
```

appfwprofile_cookieconsistency_binding

Binding object showing the cookieconsistency that can be bound to appfwprofile.

Properties

(click to see Operations)

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
state	<String>	Read-write	Enabled. Possible values = ENABLED, DISABLED
name	<String>	Read-write	Name of the profile to which to bind an exemption or rule. Minimum length = 1
isregex	<String>	Read-write	Is the cookie name a regular expression?.

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
			Possible values = REGEX, NOTREGEX
cookieconsistency	<String>	Read-write	The name of the cookie to be checked.
comment	<String>	Read-write	Any comments about the purpose of profile, or other useful information about the profile.
__count	<Double>	Read-write	count parameter

Operations

(click to see Properties)

ADD:| DELETE:| GET| GET (ALL)| COUNT

Some options that you can use for each operations:

- **Getting warnings in response:**NITRO allows you to get warnings in an operation by specifying the "warning" query parameter as "yes". For example, to get warnings while connecting to the Citrix ADC appliance, the URL is as follows:

  http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/login?warning=yes

  If any, the warnings are displayed in the response payload with the HTTP code "209 X-NITRO-WARNING".

- **Authenticated access for individual NITRO operations:**NITRO allows you to logon to the Citrix ADC appliance to perform individual operations. You can use this option instead of creating a NITRO session (using the login object) and then using that session to perform all operations,

  To do this, you must specify the username and password in the request header of the NITRO request as follows:

  X-NITRO-USER:<username>

X-NITRO-PASS:<password>

Note: In such cases, make sure that the request header DOES not include the following:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue>

`*Note:`*

Mandatory parameters are marked in redand placeholder content is marked in <green>.

add

URL:http://<Citrix-ADC-IP-address(NSIP)/nitro/v1/config/appfwprofile_cookieconsistency_binding

HTTP Method:PUT

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Content-Type:application/json

Request Payload:

```
{
"appfwprofile_cookieconsistency_binding":{
<b>"name":<String_value>,
</b>"cookieconsistency":<String_value>,
"isregex":<String_value>,
"comment":<String_value>,
"state":<String_value>
}}
```

Response:

HTTP Status Code on Success: 201 Created HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error

delete

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/appfwprofile_cookieconsistency_binding/name_value<String>

Query-parameters:

args

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/appfwprofile_cookieconsistency_binding/name_value<String>?**args=cookieconsistency:<String_value>**

HTTP Method:DELETE

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue>

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error

get

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/appfwprofile_cookieconsistency_binding/name_value<String>

Query-parameters:

filter

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/appfwprofile_cookieconsistency_binding/name_value<String>?**filter=property-name1:property-value1,property-name2:property-value2**

Use this query-parameter to get the filtered set of appfwprofile_cookieconsistency_binding resources configured on Citrix ADC. Filtering can be done on any of the properties of the resource.

pagination

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/appfwprofile_cookieconsistency_binding/name_value<String>?**pagesize=#no;pageno=#no**

Use this query-parameter to get the appfwprofile_cookieconsistency_binding resources in chunks.

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "appfwprofile_cookieconsistency_binding": [ {

"state":<String_value>,

"name":<String_value>,

"isregex":<String_value>,

"cookieconsistency":<String_value>,

"comment":<String_value>

}]}
```

get (all)

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/appfwprofile_cookieconsistency_binding

Query-parameters:

bulkbindings

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/appfwprofile_cookieconsistency_binding?**bulkbindings=yes**

NITRO allows you to fetch bindings in bulk.

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "appfwprofile_cookieconsistency_binding": [ {

"state":<String_value>,

"name":<String_value>,

"isregex":<String_value>,

"cookieconsistency":<String_value>,

"comment":<String_value>

}]}
```

count

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/appfwprofile_cookieconsistency_binding/name_value<String>?**count=yes**

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{"appfwprofile_cookieconsistency_binding": [ { "__count": "#no"} ] }
```

appfwprofile_creditcardnumber_binding

Binding object showing the creditcardnumber that can be bound to appfwprofile.

Properties

(click to see Operations)

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
creditcardnumberurl	<String>	Read-write	The url for which the list of credit card numbers are needed to be bypassed from inspection.
state	<String>	Read-write	Enabled. Possible values = ENABLED, DISABLED

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
name	<String>	Read-write	Name of the profile to which to bind an exemption or rule. Minimum length = 1
creditcardnumber	<String>	Read-write	The object expression that is to be excluded from safe commerce check.
comment	<String>	Read-write	Any comments about the purpose of profile, or other useful information about the profile.
__count	<Double>	Read-write	count parameter

Operations

(click to see Properties)

ADD:| DELETE:| GET| GET (ALL)| COUNT

Some options that you can use for each operations:

- **Getting warnings in response:**NITRO allows you to get warnings in an operation by specifying the "warning" query parameter as "yes". For example, to get warnings while connecting to the Citrix ADC appliance, the URL is as follows:

  http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/login?warning=yes

  If any, the warnings are displayed in the response payload with the HTTP code "209 X-NITRO-WARNING".
- **Authenticated access for individual NITRO operations:**NITRO allows you to logon to the Citrix ADC appliance to perform individual operations. You can use this option instead of creating a NITRO session (using the login object) and then using that session to perform all operations,

To do this, you must specify the username and password in the request header of the NITRO request as follows:

X-NITRO-USER:<username>

X-NITRO-PASS:<password>

Note: In such cases, make sure that the request header DOES not include the following:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue>

`*Note:`*

Mandatory parameters are marked in redand placeholder content is marked in <green>.

add

URL:http://<Citrix-ADC-IP-address(NSIP)/nitro/v1/config/appfwprofile_creditcardnumber_binding

HTTP Method:PUT

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Content-Type:application/json

Request Payload:

```
{

"appfwprofile_creditcardnumber_binding":{

<b>"name":<String_value>,

</b>"comment":<String_value>,

"state":<String_value>,

"creditcardnumber":<String_value>,

"creditcardnumberurl":<String_value>

}}
```

Response:

HTTP Status Code on Success: 201 Created HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error

delete

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/appfwprofile_creditcardnumber_binding/name_value<String>

Query-parameters:

args

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/appfwprofile_creditcardnumber_binding/name_value<String>?**args=creditcardnumber:<String_value>,creditcardnumberurl:<String_value>**

HTTP Method:DELETE

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue>

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error

get

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/appfwprofile_creditcardnumber_binding/name_value<String>

Query-parameters:

filter

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/appfwprofile_creditcardnumber_binding/name_value<String>?**filter=property-name1:property-value1,property-name2:property-value2**

Use this query-parameter to get the filtered set of appfwprofile_creditcardnumber_binding resources configured on Citrix ADC. Filtering can be done on any of the properties of the resource.

pagination

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/appfwprofile_creditcardnumber_binding/name_value<String>?**pagesize=#no;pageno=#no**

Use this query-parameter to get the appfwprofile_creditcardnumber_binding resources in chunks.

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "appfwprofile_creditcardnumber_binding": [ {

"creditcardnumberurl":<String_value>,

"state":<String_value>,

"name":<String_value>,

"creditcardnumber":<String_value>,

"comment":<String_value>

}]}
```

get (all)

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/appfwprofile_creditcardnumber_binding

Query-parameters:

bulkbindings

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/appfwprofile_creditcardnumber_binding?**bulkbindings=yes**

NITRO allows you to fetch bindings in bulk.

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "appfwprofile_creditcardnumber_binding": [ {

"creditcardnumberurl":<String_value>,

"state":<String_value>,

"name":<String_value>,

"creditcardnumber":<String_value>,

"comment":<String_value>

}]}
```

count

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/appfwprofile_creditcardnumber_binding/name_value<String>?**count=yes**

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{"appfwprofile_creditcardnumber_binding": [ { "__count": "#no"} ] }
```

appfwprofile_crosssitescripting_binding

Binding object showing the crosssitescripting that can be bound to appfwprofile.

Properties

(click to see Operations)

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
crosssitescripting	<String>	Read-write	The web form field name.
name	<String>	Read-write	Name of the profile to which to bind an exemption or rule.

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
			Minimum length = 1
isregex_xss	<String>	Read-write	Is the web form field name a regular expression?. Possible values = REGEX, NOTREGEX
state	<String>	Read-write	Enabled. Possible values = ENABLED, DISABLED
comment	<String>	Read-write	Any comments about the purpose of profile, or other useful information about the profile.
formactionurl_xss	<String>	Read-write	The web form action URL.
as_value_expr_xss	<String>	Read-write	The web form value expression.
as_scan_location_xss	<String>	Read-write	Location of cross-site scripting exception - form field, header or cookie. Possible values = FORMFIELD, HEADER, COOKIE
as_value_type_xss	<String>	Read-write	The web form value type. Possible values = Tag, Attribute, Pattern

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
isvalueregex_xss	<String>	Read-write	Is the web form field value a regular expression?. Possible values = REGEX, NOTREGEX
__count	<Double>	Read-write	count parameter

Operations

(click to see Properties)

ADD:| DELETE:| GET| GET (ALL)| COUNT

Some options that you can use for each operations:

- **Getting warnings in response:**NITRO allows you to get warnings in an operation by specifying the "warning" query parameter as "yes". For example, to get warnings while connecting to the Citrix ADC appliance, the URL is as follows:

  http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/login?warning=yes

  If any, the warnings are displayed in the response payload with the HTTP code "209 X-NITRO-WARNING".

- **Authenticated access for individual NITRO operations:**NITRO allows you to logon to the Citrix ADC appliance to perform individual operations. You can use this option instead of creating a NITRO session (using the login object) and then using that session to perform all operations,

  To do this, you must specify the username and password in the request header of the NITRO request as follows:

  X-NITRO-USER:<username>

  X-NITRO-PASS:<password>

  Note: In such cases, make sure that the request header DOES not include the following:

  Cookie:NITRO_AUTH_TOKEN=<tokenvalue>

`*Note:` *

Mandatory parameters are marked in redand placeholder content is marked in <green>.

add

URL:http://<Citrix-ADC-IP-address(NSIP)/nitro/v1/config/appfwprofile_crosssitescripting_binding

HTTP Method:PUT

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Content-Type:application/json

Request Payload:

```
{

"appfwprofile_crosssitescripting_binding":{

<b>"name":<String_value>,

</b>"crosssitescripting":<String_value>,

"formactionurl_xss":<String_value>,

"isregex_xss":<String_value>,

"as_scan_location_xss":<String_value>,

"as_value_type_xss":<String_value>,

"as_value_expr_xss":<String_value>,

"isvalueregex_xss":<String_value>,

"comment":<String_value>,

"state":<String_value>

}}
```

Response:

HTTP Status Code on Success: 201 Created HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error

delete

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/appfwprofile_crosssitescripting_binding/name_value<String>

Query-parameters:

args

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/appfwprofile_crosssitescripting_binding/name_value<String>?**args=crosssitescripting:<String_value>,formactionurl_xss:<String_value>,as_scan_location_xss:<Strin**

HTTP Method:DELETE

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue>

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error

get

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/appfwprofile_crosssitescripting_binding/name_value<String>

Query-parameters:

filter

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/appfwprofile_crosssitescripting_binding/name_value<String>?**filter=property-name1:property-value1,property-name2:property-value2**

Use this query-parameter to get the filtered set of appfwprofile_crosssitescripting_binding resources configured on Citrix ADC. Filtering can be done on any of the properties of the resource.

pagination

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/appfwprofile_crosssitescripting_binding/name_value<String>?**pagesize=#no;pageno=#no**

Use this query-parameter to get the appfwprofile_crosssitescripting_binding resources in chunks.

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "appfwprofile_crosssitescripting_binding": [ {

"crosssitescripting":<String_value>,

"name":<String_value>,

"isregex_xss":<String_value>,

"state":<String_value>,

"comment":<String_value>,

"formactionurl_xss":<String_value>,

"as_value_expr_xss":<String_value>,
```

```
"as_scan_location_xss":<String_value>,

"as_value_type_xss":<String_value>,

"isvalueregex_xss":<String_value>

}]}
```

get (all)

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/appfwprofile_crosssitescripting_binding

Query-parameters:

bulkbindings

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/appfwprofile_crosssitescripting_binding?**bulkbindings=yes**

NITRO allows you to fetch bindings in bulk.

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "appfwprofile_crosssitescripting_binding": [ {

"crosssitescripting":<String_value>,
```

```
"name":<String_value>,

"isregex_xss":<String_value>,

"state":<String_value>,

"comment":<String_value>,

"formactionurl_xss":<String_value>,

"as_value_expr_xss":<String_value>,

"as_scan_location_xss":<String_value>,

"as_value_type_xss":<String_value>,

"isvalueregex_xss":<String_value>

}]}
```

count

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/appfwprofile_crosssitescripting_binding/name_value<String>?**count=yes**

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{"appfwprofile_crosssitescripting_binding": [ { "__count": "#no"} ] }
```

appfwprofile_csrftag_binding

Binding object showing the csrftag that can be bound to appfwprofile.

Properties

(click to see Operations)

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
state	<String>	Read-write	Enabled. Possible values = ENABLED, DISABLED
name	<String>	Read-write	Name of the profile to which to bind an exemption or rule. Minimum length = 1
csrftag	<String>	Read-write	The web form originating URL.
csrfformactionurl	<String>	Read-write	The web form action URL.
comment	<String>	Read-write	Any comments about the purpose of profile, or other useful information about the profile.
__count	<Double>	Read-write	count parameter

Operations

(click to see Properties)

ADD:| DELETE:| GET| GET (ALL)| COUNT

Some options that you can use for each operations:

- **Getting warnings in response:**NITRO allows you to get warnings in an operation by specifying the "warning" query parameter as "yes". For example, to get warnings while connecting to the Citrix ADC appliance, the URL is as follows:

  http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/login?warning=yes

  If any, the warnings are displayed in the response payload with the HTTP code "209 X-NITRO-WARNING".

- **Authenticated access for individual NITRO operations:**NITRO allows you to logon to the Citrix ADC appliance to perform individual operations. You can use this option instead of creating a NITRO session (using the login object) and then using that session to perform all operations,

  To do this, you must specify the username and password in the request header of the NITRO request as follows:

  X-NITRO-USER:<username>

  X-NITRO-PASS:<password>

  Note: In such cases, make sure that the request header DOES not include the following:

  Cookie:NITRO_AUTH_TOKEN=<tokenvalue>

`*Note:`*

Mandatory parameters are marked in redand placeholder content is marked in <green>.

add

URL:http://<Citrix-ADC-IP-address(NSIP)/nitro/v1/config/appfwprofile_csrftag_binding

HTTP Method:PUT

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Content-Type:application/json

Request Payload:

```
{

"appfwprofile_csrftag_binding":{

<b>"name":<String_value>,

</b>"csrftag":<String_value>,

"csrfformactionurl":<String_value>,

"comment":<String_value>,

"state":<String_value>

}}
```

Response:

HTTP Status Code on Success: 201 Created HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error

delete

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/appfwprofile_csrftag_binding/name_value<String>

Query-parameters:

args

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/appfwprofile_csrftag_binding/name_value<String>?**args=csrftag:<String_value>,csrfformactionurl:<String_value>**

HTTP Method:DELETE

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue>

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error

get

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/appfwprofile_csrftag_binding/name_value<String>

Query-parameters:

filter

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/appfwprofile_csrftag_binding/name_value<String>?**filter=property-name1:property-value1,property-name2:property-value2**

Use this query-parameter to get the filtered set of appfwprofile_csrftag_binding resources configured on Citrix ADC. Filtering can be done on any of the properties of the resource.

pagination

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/appfwprofile_csrftag_binding/name_value<String>?**pagesize=#no;pageno=#no**

Use this query-parameter to get the appfwprofile_csrftag_binding resources in chunks.

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "appfwprofile_csrftag_binding": [ {
```

```
"state":<String_value>,

"name":<String_value>,

"csrftag":<String_value>,

"csrfformactionurl":<String_value>,

"comment":<String_value>

}]}
```

get (all)

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/appfwprofile_csrftag_binding

Query-parameters:

bulkbindings

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/appfwprofile_csrftag_binding?**bulkbindings=yes**

NITRO allows you to fetch bindings in bulk.

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "appfwprofile_csrftag_binding": [ {

"state":<String_value>,

"name":<String_value>,

"csrftag":<String_value>,

"csrfformactionurl":<String_value>,

"comment":<String_value>

}]}
```

count

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/appfwprofile_csrftag_binding/name_value<String>?**count=yes**

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{"appfwprofile_csrftag_binding": [ { "__count": "#no"} ] }
```

appfwprofile_denyurl_binding

Binding object showing the denyurl that can be bound to appfwprofile.

Properties

(click to see Operations)

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
state	<String>	Read-write	Enabled. Possible values = ENABLED, DISABLED
denyurl	<String>	Read-write	A regular expression that designates a URL on the Deny URL list.
name	<String>	Read-write	Name of the profile to which to bind an exemption or rule. Minimum length = 1
comment	<String>	Read-write	Any comments about the purpose of profile, or other useful information about the profile.
__count	<Double>	Read-write	count parameter

Operations

(click to see Properties)

ADD:| DELETE:| GET| GET (ALL)| COUNT

Some options that you can use for each operations:

- **Getting warnings in response:**NITRO allows you to get warnings in an operation by specifying the "warning" query parameter as "yes". For example, to get warnings while connecting to the Citrix ADC appliance, the URL is as follows:

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/login?warning=yes

If any, the warnings are displayed in the response payload with the HTTP code "209 X-NITRO-WARNING".

- **Authenticated access for individual NITRO operations:**NITRO allows you to logon to the Citrix ADC appliance to perform individual operations. You can use this option instead of creating a NITRO session (using the login object) and then using that session to perform all operations,

  To do this, you must specify the username and password in the request header of the NITRO request as follows:

  X-NITRO-USER:<username>

  X-NITRO-PASS:<password>

  Note: In such cases, make sure that the request header DOES not include the following:

  Cookie:NITRO_AUTH_TOKEN=<tokenvalue>

`*Note:`*

Mandatory parameters are marked in redand placeholder content is marked in <green>.

add

URL:http://<Citrix-ADC-IP-address(NSIP)/nitro/v1/config/appfwprofile_denyurl_binding

HTTP Method:PUT

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Content-Type:application/json

Request Payload:

```
{

"appfwprofile_denyurl_binding":{

<b>"name":<String_value>,

</b>"denyurl":<String_value>,
```

```
"comment":<String_value>,

"state":<String_value>

}}
```

Response:

HTTP Status Code on Success: 201 Created HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error

delete

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/appfwprofile_denyurl_binding/name_value<String>

Query-parameters:

args

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/appfwprofile_denyurl_binding/name_value<String>?**args=denyurl:<String_value>**

HTTP Method:DELETE

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue>

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error

get

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/appfwprofile_denyurl_binding/name_value<String>

Query-parameters:

filter

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/appfwprofile_denyurl_binding/name_value<String>?**filter=property-name1:property-value1,property-name2:property-value2**

Use this query-parameter to get the filtered set of appfwprofile_denyurl_binding resources configured on Citrix ADC. Filtering can be done on any of the properties of the resource.

pagination

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/appfwprofile_denyurl_binding/name_value<String>?**pagesize=#no;pageno=#no**

Use this query-parameter to get the appfwprofile_denyurl_binding resources in chunks.

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "appfwprofile_denyurl_binding": [ {

"state":<String_value>,

"denyurl":<String_value>,

"name":<String_value>,

"comment":<String_value>

}]}
```

get (all)

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/appfwprofile_denyurl_binding

Query-parameters:

bulkbindings

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/appfwprofile_denyurl_binding?**bulkbindings=yes**

NITRO allows you to fetch bindings in bulk.

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "appfwprofile_denyurl_binding": [ {

"state":<String_value>,

"denyurl":<String_value>,

"name":<String_value>,

"comment":<String_value>

}]}
```

count

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/appfwprofile_denyurl_binding/name_value<String>?**count=yes**

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{"appfwprofile_denyurl_binding": [ { "__count": "#no"} ] }
```

appfwprofile_excluderescontenttype_binding

Binding object showing the excluderescontenttype that can be bound to appfwprofile.

Properties

(click to see Operations)

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
excluderescontenttype	<String>	Read-write	A regular expression that represents the content type of the response that are to be excluded from inspection.

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
name	<String>	Read-write	Name of the profile to which to bind an exemption or rule. Minimum length = 1
state	<String>	Read-write	Enabled. Possible values = ENABLED, DISABLED
comment	<String>	Read-write	Any comments about the purpose of profile, or other useful information about the profile.
__count	<Double>	Read-write	count parameter

Operations

(click to see Properties)

ADD:| DELETE:| GET| GET (ALL)| COUNT

Some options that you can use for each operations:

- **Getting warnings in response:**NITRO allows you to get warnings in an operation by specifying the "warning" query parameter as "yes". For example, to get warnings while connecting to the Citrix ADC appliance, the URL is as follows:

  http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/login?warning=yes

  If any, the warnings are displayed in the response payload with the HTTP code "209 X-NITRO-WARNING".
- **Authenticated access for individual NITRO operations:**NITRO allows you to logon to the Citrix ADC appliance to perform individual operations. You can use this option instead of creating a NITRO session (using the login object) and then using that session to perform all operations,

To do this, you must specify the username and password in the request header of the NITRO request as follows:

X-NITRO-USER:<username>

X-NITRO-PASS:<password>

Note: In such cases, make sure that the request header DOES not include the following:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue>

`*Note:`*

Mandatory parameters are marked in redand placeholder content is marked in <green>.

add

URL:http://<Citrix-ADC-IP-address(NSIP)/nitro/v1/config/appfwprofile_excluderescontenttype_binding

HTTP Method:PUT

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Content-Type:application/json

Request Payload:

```
{

"appfwprofile_excluderescontenttype_binding":{

<b>"name":<String_value>,

</b>"comment":<String_value>,

"state":<String_value>,

"excluderescontenttype":<String_value>

}}
```

Response:

HTTP Status Code on Success: 201 Created HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error

delete

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/appfwprofile_excluderescontenttype_binding/name_value<String>

Query-parameters:

args

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/appfwprofile_excluderescontenttype_binding/name_value<String>?**args=excluderescontenttype:<String_value>**

HTTP Method:DELETE

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue>

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error

get

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/appfwprofile_excluderescontenttype_binding/name_value<String>

Query-parameters:

filter

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/appfwprofile_excluderescontenttype_binding/name_value<String>?**filter=property-name1:property-value1,property-name2:property-value2**

Use this query-parameter to get the filtered set of appfwprofile_excluderescontenttype_binding resources configured on Citrix ADC. Filtering can be done on any of the properties of the resource.

pagination

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/appfwprofile_excluderescontenttype_binding/name_value<String>?**pagesize=#no;pageno=#no**

Use this query-parameter to get the appfwprofile_excluderescontenttype_binding resources in chunks.

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "appfwprofile_excluderescontenttype_binding": [ {

"excluderescontenttype":<String_value>,

"name":<String_value>,

"state":<String_value>,

"comment":<String_value>

}]}
```

get (all)

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/appfwprofile_excluderescontenttype_binding

Query-parameters:

bulkbindings

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/appfwprofile_excluderescontenttype_binding?**bulkbindings=yes**

NITRO allows you to fetch bindings in bulk.

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "appfwprofile_excluderescontenttype_binding": [ {

"excluderescontenttype":<String_value>,

"name":<String_value>,

"state":<String_value>,

"comment":<String_value>

}]}
```

count

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/appfwprofile_excluderescontenttype_binding/name_value<String>?**count=yes**

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{"appfwprofile_excluderescontenttype_binding": [ { "__count": "#no"} ] }
```

appfwprofile_fieldconsistency_binding

Binding object showing the fieldconsistency that can be bound to appfwprofile.

Properties

(click to see Operations)

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
fieldconsistency	<String>	Read-write	The web form field name.
state	<String>	Read-write	Enabled. Possible values = ENABLED, DISABLED
name	<String>	Read-write	Name of the profile to which to bind an exemption or rule. Minimum length = 1

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
isregex_ffc	<String>	Read-write	Is the web form field name a regular expression?. Possible values = REGEX, NOTREGEX
formactionurl_ffc	<String>	Read-write	The web form action URL.
comment	<String>	Read-write	Any comments about the purpose of profile, or other useful information about the profile.
__count	<Double>	Read-write	count parameter

Operations

(click to see Properties)

ADD:| DELETE:| GET| GET (ALL)| COUNT

Some options that you can use for each operations:

- **Getting warnings in response:**NITRO allows you to get warnings in an operation by specifying the "warning" query parameter as "yes". For example, to get warnings while connecting to the Citrix ADC appliance, the URL is as follows:

  http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/login?warning=yes

  If any, the warnings are displayed in the response payload with the HTTP code "209 X-NITRO-WARNING".
- **Authenticated access for individual NITRO operations:**NITRO allows you to logon to the Citrix ADC appliance to perform individual operations. You can use this option instead of creating a NITRO session (using the login object) and then using that session to perform all operations,

To do this, you must specify the username and password in the request header of the NITRO request as follows:

X-NITRO-USER:<username>

X-NITRO-PASS:<password>

Note: In such cases, make sure that the request header DOES not include the following:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue>

`*Note:`*

Mandatory parameters are marked in redand placeholder content is marked in <green>.

add

URL:http://<Citrix-ADC-IP-address(NSIP)/nitro/v1/config/appfwprofile_fieldconsistency_binding

HTTP Method:PUT

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Content-Type:application/json

Request Payload:

```
{

"appfwprofile_fieldconsistency_binding":{

<b>"name":<String_value>,

</b>"fieldconsistency":<String_value>,

"formactionurl_ffc":<String_value>,

"isregex_ffc":<String_value>,

"comment":<String_value>,

"state":<String_value>
```

```
}}
```

Response:

HTTP Status Code on Success: 201 Created HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error

delete

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/appfwprofile_fieldconsistency_binding/name_value<String>

Query-parameters:

args

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/appfwprofile_fieldconsistency_binding/name_value<String>?**args=fieldconsistency:<String_value>,formactionurl_ffc:<String_value>**

HTTP Method:DELETE

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue>

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error

get

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/appfwprofile_fieldconsistency_binding/name_value<String>

Query-parameters:

filter

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/appfwprofile_fieldconsistency_binding/name_value<String>?**filter=property-name1:property-value1,property-name2:property-value2**

Use this query-parameter to get the filtered set of appfwprofile_fieldconsistency_binding resources configured on Citrix ADC. Filtering can be done on any of the properties of the resource.

pagination

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/appfwprofile_fieldconsistency_binding/name_value<String>?**pagesize=#no;pageno=#no**

Use this query-parameter to get the appfwprofile_fieldconsistency_binding resources in chunks.

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "appfwprofile_fieldconsistency_binding": [ {

"fieldconsistency":<String_value>,

"state":<String_value>,

"name":<String_value>,

"isregex_ffc":<String_value>,

"formactionurl_ffc":<String_value>,

"comment":<String_value>

}]}
```

get (all)

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/appfwprofile_fieldconsistency_binding

Query-parameters:

bulkbindings

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/appfwprofile_fieldconsistency_binding?**bulkbindings=yes**

NITRO allows you to fetch bindings in bulk.

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "appfwprofile_fieldconsistency_binding": [ {

"fieldconsistency":<String_value>,

"state":<String_value>,

"name":<String_value>,

"isregex_ffc":<String_value>,

"formactionurl_ffc":<String_value>,
```

```
"comment":<String_value>

}]}
```

count

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/appfwprofile_fieldconsistency_binding/name_value<String>?**count=yes**

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{"appfwprofile_fieldconsistency_binding": [ { "__count": "#no"} ] }
```

appfwprofile_fieldformat_binding

Binding object showing the fieldformat that can be bound to appfwprofile.

Properties

(click to see Operations)

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
state	<String>	Read-write	Enabled.

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
			Possible values = ENABLED, DISABLED
fieldformatmaxlength	<Double>	Read-write	The maximum allowed length for data in this form field.
isregex_ff	<String>	Read-write	Is the form field name a regular expression?. Possible values = REGEX, NOTREGEX
fieldtype	<String>	Read-write	The field type you are assigning to this form field.
formactionurl_ff	<String>	Read-write	Action URL of the form field to which a field format will be assigned.
name	<String>	Read-write	Name of the profile to which to bind an exemption or rule. Minimum length = 1
fieldformatminlength	<Double>	Read-write	The minimum allowed length for data in this form field.
comment	<String>	Read-write	Any comments about the purpose of profile, or other useful information about the profile.

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
fieldformat	<String>	Read-write	Name of the form field to which a field format will be assigned.
__count	<Double>	Read-write	count parameter

Operations

(click to see Properties)

ADD:| DELETE:| GET| GET (ALL)| COUNT

Some options that you can use for each operations:

- **Getting warnings in response:**NITRO allows you to get warnings in an operation by specifying the "warning" query parameter as "yes". For example, to get warnings while connecting to the Citrix ADC appliance, the URL is as follows:

  http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/login?warning=yes

  If any, the warnings are displayed in the response payload with the HTTP code "209 X-NITRO-WARNING".

- **Authenticated access for individual NITRO operations:**NITRO allows you to logon to the Citrix ADC appliance to perform individual operations. You can use this option instead of creating a NITRO session (using the login object) and then using that session to perform all operations,

  To do this, you must specify the username and password in the request header of the NITRO request as follows:

  X-NITRO-USER:<username>

  X-NITRO-PASS:<password>

  Note: In such cases, make sure that the request header DOES not include the following:

  Cookie:NITRO_AUTH_TOKEN=<tokenvalue>

`*Note:`*

Mandatory parameters are marked in redand placeholder content is marked in <green>.

add

URL:http://<Citrix-ADC-IP-address(NSIP)/nitro/v1/config/appfwprofile_fieldformat_binding

HTTP Method:PUT

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Content-Type:application/json

Request Payload:

```
{

"appfwprofile_fieldformat_binding":{

<b>"name":<String_value>,

</b>"fieldformat":<String_value>,

"formactionurl_ff":<String_value>,

"fieldtype":<String_value>,

"fieldformatminlength":<Double_value>,

"fieldformatmaxlength":<Double_value>,

"isregex_ff":<String_value>,

"comment":<String_value>,

"state":<String_value>

}}
```

Response:

HTTP Status Code on Success: 201 Created HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error

delete

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/appfwprofile_fieldformat_binding/name_value<String>

Query-parameters:

args

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/appfwprofile_fieldformat_binding/name_value<String>?**args=fieldformat:<String_value>,formactionurl_ff:<String_value>**

HTTP Method:DELETE

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue>

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error

get

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/appfwprofile_fieldformat_binding/name_value<String>

Query-parameters:

filter

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/appfwprofile_fieldformat_binding/name_value<String>?**filter=property-name1:property-value1,property-name2:property-value2**

Use this query-parameter to get the filtered set of appfwprofile_fieldformat_binding resources configured on Citrix ADC. Filtering can be done on any of the properties of the resource.

pagination

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/appfwprofile_fieldformat_binding/name_value<String>?**pagesize=#no;pageno=#no**

Use this query-parameter to get the appfwprofile_fieldformat_binding resources in chunks.

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "appfwprofile_fieldformat_binding": [ {

"state":<String_value>,

"fieldformatmaxlength":<Double_value>,

"isregex_ff":<String_value>,

"fieldtype":<String_value>,

"formactionurl_ff":<String_value>,

"name":<String_value>,

"fieldformatminlength":<Double_value>,

"comment":<String_value>,

"fieldformat":<String_value>

}]}
```

get (all)

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/appfwprofile_fieldformat_binding

Query-parameters:

bulkbindings

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/appfwprofile_fieldformat_binding?**bulkbindings=yes**

NITRO allows you to fetch bindings in bulk.

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "appfwprofile_fieldformat_binding": [ {

"state":<String_value>,

"fieldformatmaxlength":<Double_value>,

"isregex_ff":<String_value>,

"fieldtype":<String_value>,

"formactionurl_ff":<String_value>,

"name":<String_value>,

"fieldformatminlength":<Double_value>,

"comment":<String_value>,
```

```
"fieldformat":<String_value>

}]}
```

count

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/appfwprofile_fieldformat_binding/name_value<String>?**count=yes**

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{"appfwprofile_fieldformat_binding": [ { "__count": "#no"} ] }
```

appfwprofile_logexpression_binding

Binding object showing the logexpression that can be bound to appfwprofile.

Properties

(click to see Operations)

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
state	<String>	Read-write	Enabled.

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
			Possible values = ENABLED, DISABLED
name	<String>	Read-write	Name of the profile to which to bind an exemption or rule. Minimum length = 1
as_logexpression	<String>	Read-write	LogExpression to log when violation happened on appfw profile. Maximum length = 1500
logexpression	<String>	Read-write	Name of LogExpression object.
comment	<String>	Read-write	Any comments about the purpose of profile, or other useful information about the profile.
__count	<Double>	Read-write	count parameter

Operations

(click to see Properties)

ADD:| DELETE:| GET| GET (ALL)| COUNT

Some options that you can use for each operations:

- **Getting warnings in response:**NITRO allows you to get warnings in an operation by specifying the "warning" query parameter as "yes". For example, to get warnings while connecting to the Citrix ADC appliance, the URL is as follows:

  http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/login?warning=yes

If any, the warnings are displayed in the response payload with the HTTP code "209 X-NITRO-WARNING".

- **Authenticated access for individual NITRO operations:**NITRO allows you to logon to the Citrix ADC appliance to perform individual operations. You can use this option instead of creating a NITRO session (using the login object) and then using that session to perform all operations,

  To do this, you must specify the username and password in the request header of the NITRO request as follows:

  X-NITRO-USER:<username>

  X-NITRO-PASS:<password>

  Note: In such cases, make sure that the request header DOES not include the following:

  Cookie:NITRO_AUTH_TOKEN=<tokenvalue>

`*Note:`*

Mandatory parameters are marked in redand placeholder content is marked in <green>.

add

URL:http://<Citrix-ADC-IP-address(NSIP)/nitro/v1/config/appfwprofile_logexpression_binding

HTTP Method:PUT

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Content-Type:application/json

Request Payload:

```
{

"appfwprofile_logexpression_binding":{

<b>"name":<String_value>,

</b>"comment":<String_value>,
```

```
"state":<String_value>,

"logexpression":<String_value>,

"as_logexpression":<String_value>

}}
```

Response:

HTTP Status Code on Success: 201 Created HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error

delete

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/appfwprofile_logexpression_binding/name_value<String>

Query-parameters:

args

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/appfwprofile_logexpression_binding/name_value<String>?**args=logexpression:<String_value>**

HTTP Method:DELETE

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue>

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error

get

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/appfwprofile_logexpression_binding/name_value<String>

Query-parameters:

filter

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/appfwprofile_logexpression_binding/name_value<String>?**filter=property-name1:property-value1,property-name2:property-value2**

Use this query-parameter to get the filtered set of appfwprofile_logexpression_binding resources configured on Citrix ADC. Filtering can be done on any of the properties of the resource.

pagination

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/appfwprofile_logexpression_binding/name_value<String>?**pagesize=#no;pageno=#no**

Use this query-parameter to get the appfwprofile_logexpression_binding resources in chunks.

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "appfwprofile_logexpression_binding": [ {

"state":<String_value>,

"name":<String_value>,

"as_logexpression":<String_value>,

"logexpression":<String_value>,
```

```
"comment":<String_value>

}]}
```

get (all)

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/appfwprofile_logexpression_binding

Query-parameters:

bulkbindings

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/appfwprofile_logexpression_binding?**bulkbindings=yes**

NITRO allows you to fetch bindings in bulk.

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "appfwprofile_logexpression_binding": [ {

"state":<String_value>,

"name":<String_value>,

"as_logexpression":<String_value>,
```

```
"logexpression":<String_value>,

"comment":<String_value>

}]}
```

count

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/appfwprofile_logexpression_binding/name_value<String>?**count=yes**

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{"appfwprofile_logexpression_binding": [ { "__count": "#no"} ] }
```

appfwprofile_safeobject_binding

Binding object showing the safeobject that can be bound to appfwprofile.

Properties

(click to see Operations)

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
maxmatchlength	<Double>	Read-write	Maximum match length for a Safe Object expression.
state	<String>	Read-write	Enabled. Possible values = ENABLED, DISABLED
name	<String>	Read-write	Name of the profile to which to bind an exemption or rule. Minimum length = 1
safeobject	<String>	Read-write	Name of the Safe Object.
as_expression	<String>	Read-write	A regular expression that defines the Safe Object.
comment	<String>	Read-write	Any comments about the purpose of profile, or other useful information about the profile.
action	<String[]>	Read-write	Safe Object action types. (BLOCK | LOG | STATS | NONE). Possible values = none, block, log, remove, stats, xout
__count	<Double>	Read-write	count parameter

Operations

(click to see Properties)

ADD:| DELETE:| GET| GET (ALL)| COUNT

Some options that you can use for each operations:

- **Getting warnings in response:**NITRO allows you to get warnings in an operation by specifying the "warning" query parameter as "yes". For example, to get warnings while connecting to the Citrix ADC appliance, the URL is as follows:

  http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/login?warning=yes

  If any, the warnings are displayed in the response payload with the HTTP code "209 X-NITRO-WARNING".

- **Authenticated access for individual NITRO operations:**NITRO allows you to logon to the Citrix ADC appliance to perform individual operations. You can use this option instead of creating a NITRO session (using the login object) and then using that session to perform all operations,

  To do this, you must specify the username and password in the request header of the NITRO request as follows:

  X-NITRO-USER:<username>

  X-NITRO-PASS:<password>

  Note: In such cases, make sure that the request header DOES not include the following:

  Cookie:NITRO_AUTH_TOKEN=<tokenvalue>

`*Note:`*

Mandatory parameters are marked in redand placeholder content is marked in <green>.

add

URL:http://<Citrix-ADC-IP-address(NSIP)/nitro/v1/config/appfwprofile_safeobject_binding

HTTP Method:PUT

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Content-Type:application/json

Request Payload:

```
{

"appfwprofile_safeobject_binding":{

<b>"name":<String_value>,

</b>"safeobject":<String_value>,

"as_expression":<String_value>,

"maxmatchlength":<Double_value>,

"action":<String[]_value>,

"comment":<String_value>,

"state":<String_value>

}}
```

Response:

HTTP Status Code on Success: 201 Created HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error

delete

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/appfwprofile_safeobject_binding/name_value<String>

Query-parameters:

args

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/appfwprofile_safeobject_binding/name_value<String>?**args=safeobject:<String_value>**

HTTP Method:DELETE

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue>

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error

get

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/appfwprofile_safeobject_binding/name_value<String>

Query-parameters:

filter

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/appfwprofile_safeobject_binding/name_value<String>?**filter=property-name1:property-value1,property-name2:property-value2**

Use this query-parameter to get the filtered set of appfwprofile_safeobject_binding resources configured on Citrix ADC. Filtering can be done on any of the properties of the resource.

pagination

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/appfwprofile_safeobject_binding/name_value<String>?**pagesize=#no;pageno=#no**

Use this query-parameter to get the appfwprofile_safeobject_binding resources in chunks.

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "appfwprofile_safeobject_binding": [ {

"maxmatchlength":<Double_value>,

"state":<String_value>,

"name":<String_value>,

"safeobject":<String_value>,

"as_expression":<String_value>,

"comment":<String_value>,

"action":<String[]_value>

}]}
```

get (all)

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/appfwprofile_safeobject_binding

Query-parameters:

bulkbindings

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/appfwprofile_safeobject_binding?**bulkbindings=yes**

NITRO allows you to fetch bindings in bulk.

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "appfwprofile_safeobject_binding": [ {

"maxmatchlength":<Double_value>,

"state":<String_value>,

"name":<String_value>,

"safeobject":<String_value>,

"as_expression":<String_value>,

"comment":<String_value>,

"action":<String[]_value>

}]}
```

count

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/appfwprofile_safeobject_binding/name_value<String>?**count=yes**

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{"appfwprofile_safeobject_binding": [ { "__count": "#no"} ] }
```

appfwprofile_sqlinjection_binding

Binding object showing the sqlinjection that can be bound to appfwprofile.

Properties

(click to see Operations)

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
as_value_expr_sql	<String>	Read-write	The web form value expression.
state	<String>	Read-write	Enabled. Possible values = ENABLED, DISABLED
formactionurl_sql	<String>	Read-write	The web form action URL.
name	<String>	Read-write	Name of the profile to which to bind an exemption or rule. Minimum length = 1
isregex_sql	<String>	Read-write	Is the web form field name a regular expression?.

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
			Possible values = REGEX, NOTREGEX
isvalueregex_sql	<String>	Read-write	Is the web form field value a regular expression?. Possible values = REGEX, NOTREGEX
as_scan_location_sql	<String>	Read-write	Location of SQL injection exception - form field, header or cookie. Possible values = FORMFIELD, HEADER, COOKIE
sqlinjection	<String>	Read-write	The web form field name.
as_value_type_sql	<String>	Read-write	The web form value type. Possible values = Keyword, SpecialString, Wildchar
comment	<String>	Read-write	Any comments about the purpose of profile, or other useful information about the profile.
__count	<Double>	Read-write	count parameter

Operations

(click to see Properties)

ADD:| DELETE:| GET| GET (ALL)| COUNT

Some options that you can use for each operations:

- **Getting warnings in response:**NITRO allows you to get warnings in an operation by specifying the "warning" query parameter as "yes". For example, to get warnings while connecting to the Citrix ADC appliance, the URL is as follows:

  http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/login?warning=yes

  If any, the warnings are displayed in the response payload with the HTTP code "209 X-NITRO-WARNING".

- **Authenticated access for individual NITRO operations:**NITRO allows you to logon to the Citrix ADC appliance to perform individual operations. You can use this option instead of creating a NITRO session (using the login object) and then using that session to perform all operations,

  To do this, you must specify the username and password in the request header of the NITRO request as follows:

  X-NITRO-USER:<username>

  X-NITRO-PASS:<password>

  Note: In such cases, make sure that the request header DOES not include the following:

  Cookie:NITRO_AUTH_TOKEN=<tokenvalue>

`*Note:`*

Mandatory parameters are marked in redand placeholder content is marked in <green>.

add

URL:http://<Citrix-ADC-IP-address(NSIP)/nitro/v1/config/appfwprofile_sqlinjection_binding

HTTP Method:PUT

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Content-Type:application/json

Request Payload:

```
{

"appfwprofile_sqlinjection_binding":{

<b>"name":<String_value>,

</b>"sqlinjection":<String_value>,

"formactionurl_sql":<String_value>,

"isregex_sql":<String_value>,

"as_scan_location_sql":<String_value>,

"as_value_type_sql":<String_value>,

"as_value_expr_sql":<String_value>,

"isvalueregex_sql":<String_value>,

"comment":<String_value>,

"state":<String_value>

}}
```

Response:

HTTP Status Code on Success: 201 Created HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error

delete

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/appfwprofile_sqlinjection_binding/name_value<String>

Query-parameters:

args

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/appfwprofile_sqlinjection_binding/name_value<String>?**args=sqlinjection:<String_value>,formactionurl_sql:<String_value>,as_scan_location_sql:<String_valu**

HTTP Method:DELETE

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue>

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error

get

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/appfwprofile_sqlinjection_binding/name_value<String>

Query-parameters:

filter

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/appfwprofile_sqlinjection_binding/name_value<String>?**filter=property-name1:property-value1,property-name2:property-value2**

Use this query-parameter to get the filtered set of appfwprofile_sqlinjection_binding resources configured on Citrix ADC. Filtering can be done on any of the properties of the resource.

pagination

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/appfwprofile_sqlinjection_binding/name_value<String>?**pagesize=#no;pageno=#no**

Use this query-parameter to get the appfwprofile_sqlinjection_binding resources in chunks.

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "appfwprofile_sqlinjection_binding": [ {

"as_value_expr_sql":<String_value>,

"state":<String_value>,

"formactionurl_sql":<String_value>,

"name":<String_value>,

"isregex_sql":<String_value>,

"isvalueregex_sql":<String_value>,

"as_scan_location_sql":<String_value>,

"sqlinjection":<String_value>,

"as_value_type_sql":<String_value>,

"comment":<String_value>

}]}
```

get (all)

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/appfwprofile_sqlinjection_binding

Query-parameters:

bulkbindings

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/appfwprofile_sqlinjection_binding?**bulkbindings=yes**

NITRO allows you to fetch bindings in bulk.

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "appfwprofile_sqlinjection_binding": [ {

"as_value_expr_sql":<String_value>,

"state":<String_value>,

"formactionurl_sql":<String_value>,

"name":<String_value>,

"isregex_sql":<String_value>,

"isvalueregex_sql":<String_value>,

"as_scan_location_sql":<String_value>,

"sqlinjection":<String_value>,

"as_value_type_sql":<String_value>,

"comment":<String_value>
```

```
}]}
```

count

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/appfwprofile_sqlinjection_binding/name_value<String>?**count=yes**

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{"appfwprofile_sqlinjection_binding": [ { "__count": "#no"} ] }
```

appfwprofile_starturl_binding

Binding object showing the starturl that can be bound to appfwprofile.

Properties

(click to see Operations)

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
state	<String>	Read-write	Enabled.

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
			Possible values = ENABLED, DISABLED
name	<String>	Read-write	Name of the profile to which to bind an exemption or rule. Minimum length = 1
starturl	<String>	Read-write	A regular expression that designates a URL on the Start URL list.
comment	<String>	Read-write	Any comments about the purpose of profile, or other useful information about the profile.
__count	<Double>	Read-write	count parameter

Operations

(click to see Properties)

ADD:| DELETE:| GET| GET (ALL)| COUNT

Some options that you can use for each operations:

- **Getting warnings in response:**NITRO allows you to get warnings in an operation by specifying the "warning" query parameter as "yes". For example, to get warnings while connecting to the Citrix ADC appliance, the URL is as follows:

  http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/login?warning=yes

  If any, the warnings are displayed in the response payload with the HTTP code "209 X-NITRO-WARNING".
- **Authenticated access for individual NITRO operations:**NITRO allows you to logon to the Citrix ADC appliance to perform individual operations. You can use this option instead of

creating a NITRO session (using the login object) and then using that session to perform all operations,

To do this, you must specify the username and password in the request header of the NITRO request as follows:

X-NITRO-USER:<username>

X-NITRO-PASS:<password>

Note: In such cases, make sure that the request header DOES not include the following:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue>

`*Note:`*

Mandatory parameters are marked in redand placeholder content is marked in <green>.

add

URL:http://<Citrix-ADC-IP-address(NSIP)/nitro/v1/config/appfwprofile_starturl_binding

HTTP Method:PUT

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Content-Type:application/json

Request Payload:

```
{

"appfwprofile_starturl_binding":{

<b>"name":<String_value>,

</b>"starturl":<String_value>,

"comment":<String_value>,

"state":<String_value>

}}
```

Response:

HTTP Status Code on Success: 201 Created HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error

delete

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/appfwprofile_starturl_binding/name_value<String>

Query-parameters:

args

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/appfwprofile_starturl_binding/name_value<String>?**args=starturl:<String_value>**

HTTP Method:DELETE

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue>

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error

get

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/appfwprofile_starturl_binding/name_value<String>

Query-parameters:

filter

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/appfwprofile_starturl_binding/name_value<String>?**filter=property-name1:property-value1,property-name2:property-value2**

Use this query-parameter to get the filtered set of appfwprofile_starturl_binding resources configured on Citrix ADC. Filtering can be done on any of the properties of the resource.

pagination

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/appfwprofile_starturl_binding/name_value<String>?**pagesize=#no;pageno=#no**

Use this query-parameter to get the appfwprofile_starturl_binding resources in chunks.

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "appfwprofile_starturl_binding": [ {

"state":<String_value>,

"name":<String_value>,

"starturl":<String_value>,

"comment":<String_value>

}]}
```

get (all)

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/appfwprofile_starturl_binding

Query-parameters:

bulkbindings

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/appfwprofile_starturl_binding?**bulkbindings=yes**

NITRO allows you to fetch bindings in bulk.

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "appfwprofile_starturl_binding": [ {

"state":<String_value>,

"name":<String_value>,

"starturl":<String_value>,

"comment":<String_value>

}]}
```

count

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/appfwprofile_starturl_binding/name_value<String>?**count=yes**

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{"appfwprofile_starturl_binding": [ { "__count": "#no"} ] }
```

appfwprofile_trustedlearningclients_binding

Binding object showing the trustedlearningclients that can be bound to appfwprofile.

Properties

(click to see Operations)

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
state	<String>	Read-write	Enabled. Possible values = ENABLED, DISABLED
trustedlearningclients	<String>	Read-write	Specify trusted host/ network IP.
name	<String>	Read-write	Name of the profile to which to bind an exemption or rule. Minimum length = 1

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
comment	<String>	Read-write	Any comments about the purpose of profile, or other useful information about the profile.
__count	<Double>	Read-write	count parameter

Operations

(click to see Properties)

ADD:| DELETE:| GET| GET (ALL)| COUNT

Some options that you can use for each operations:

- **Getting warnings in response:**NITRO allows you to get warnings in an operation by specifying the "warning" query parameter as "yes". For example, to get warnings while connecting to the Citrix ADC appliance, the URL is as follows:

  http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/login?warning=yes

  If any, the warnings are displayed in the response payload with the HTTP code "209 X-NITRO-WARNING".

- **Authenticated access for individual NITRO operations:**NITRO allows you to logon to the Citrix ADC appliance to perform individual operations. You can use this option instead of creating a NITRO session (using the login object) and then using that session to perform all operations,

  To do this, you must specify the username and password in the request header of the NITRO request as follows:

  X-NITRO-USER:<username>

  X-NITRO-PASS:<password>

  Note: In such cases, make sure that the request header DOES not include the following:

  Cookie:NITRO_AUTH_TOKEN=<tokenvalue>

`*Note:`*

Mandatory parameters are marked in redand placeholder content is marked in <green>.

add

URL:http://<Citrix-ADC-IP-address(NSIP)/nitro/v1/config/appfwprofile_trustedlearningclients_binding

HTTP Method:PUT

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Content-Type:application/json

Request Payload:

```
{

"appfwprofile_trustedlearningclients_binding":{

<b>"name":<String_value>,

</b>"trustedlearningclients":<String_value>,

"comment":<String_value>,

"state":<String_value>

}}
```

Response:

HTTP Status Code on Success: 201 Created HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error

delete

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/appfwprofile_trustedlearningclients_binding/name_value<String>

Query-parameters:

args

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/appfwprofile_trustedlearningclients_binding/name_value<String>?**args=trustedlearningclients:<String_value>**

HTTP Method:DELETE

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue>

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error

get

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/appfwprofile_trustedlearningclients_binding/name_value<String>

Query-parameters:

filter

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/appfwprofile_trustedlearningclients_binding/name_value<String>?**filter=property-name1:property-value1,property-name2:property-value2**

Use this query-parameter to get the filtered set of appfwprofile_trustedlearningclients_binding resources configured on Citrix ADC. Filtering can be done on any of the properties of the resource.

pagination

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/appfwprofile_trustedlearningclients_binding/name_value<String>?**pagesize=#no;pageno=#no**

Use this query-parameter to get the appfwprofile_trustedlearningclients_binding resources in chunks.

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "appfwprofile_trustedlearningclients_binding": [ {

"state":<String_value>,

"trustedlearningclients":<String_value>,

"name":<String_value>,

"comment":<String_value>

}]}
```

get (all)

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/appfwprofile_trustedlearningclients_binding

Query-parameters:

bulkbindings

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/appfwprofile_trustedlearningclients_binding?**bulkbindings=yes**

NITRO allows you to fetch bindings in bulk.

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "appfwprofile_trustedlearningclients_binding": [ {

"state":<String_value>,

"trustedlearningclients":<String_value>,

"name":<String_value>,

"comment":<String_value>

}]}
```

count

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/appfwprofile_trustedlearningclients_binding/name_value<String>?**count=yes**

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{"appfwprofile_trustedlearningclients_binding": [ { "__count": "#no"} ] }
```

appfwprofile_xmlattachmenturl_binding

Binding object showing the xmlattachmenturl that can be bound to appfwprofile.

Properties

(click to see Operations)

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
xmlattachmenturl	<String>	Read-write	XML attachment URL regular expression length.
name	<String>	Read-write	Name of the profile to which to bind an exemption or rule. Minimum length = 1
xmlmaxattachmentsize	<Double>	Read-write	Specify maximum attachment size.
xmlmaxattachmentsizecheck	<String>	Read-write	State if XML Max attachment size Check is ON or OFF. Protects against XML requests with large attachment data. Possible values = ON, OFF
state	<String>	Read-write	Enabled.

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
			Possible values = ENABLED, DISABLED
xmlattachmentcontenttypecheck	<String>	Read-write	State if XML attachment content-type check is ON or OFF. Protects against XML requests with illegal attachments. Possible values = ON, OFF
comment	<String>	Read-write	Any comments about the purpose of profile, or other useful information about the profile.
xmlattachmentcontenttype	<String>	Read-write	Specify content-type regular expression.
__count	<Double>	Read-write	count parameter

Operations

(click to see Properties)

ADD:| DELETE:| GET| GET (ALL)| COUNT

Some options that you can use for each operations:

- **Getting warnings in response:**NITRO allows you to get warnings in an operation by specifying the "warning" query parameter as "yes". For example, to get warnings while connecting to the Citrix ADC appliance, the URL is as follows:

  http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/login?warning=yes

  If any, the warnings are displayed in the response payload with the HTTP code "209 X-NITRO-WARNING".

- **Authenticated access for individual NITRO operations:**NITRO allows you to logon to the Citrix ADC appliance to perform individual operations. You can use this option instead of creating a NITRO session (using the login object) and then using that session to perform all operations,

  To do this, you must specify the username and password in the request header of the NITRO request as follows:

  X-NITRO-USER:<username>

  X-NITRO-PASS:<password>

  Note: In such cases, make sure that the request header DOES not include the following:

  Cookie:NITRO_AUTH_TOKEN=<tokenvalue>

`*Note:`*

Mandatory parameters are marked in redand placeholder content is marked in <green>.

add

URL:http://<Citrix-ADC-IP-address(NSIP)/nitro/v1/config/appfwprofile_xmlattachmenturl_binding

HTTP Method:PUT

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Content-Type:application/json

Request Payload:

```
{

"appfwprofile_xmlattachmenturl_binding":{

<b>"name":<String_value>,

</b>"comment":<String_value>,

"state":<String_value>,

"xmlattachmenturl":<String_value>,
```

```
"xmlmaxattachmentsizecheck":<String_value>,

"xmlmaxattachmentsize":<Double_value>,

"xmlattachmentcontenttypecheck":<String_value>,

"xmlattachmentcontenttype":<String_value>

}}
```

Response:

HTTP Status Code on Success: 201 Created HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error

delete

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/appfwprofile_xmlattachmenturl_binding/name_value<String>

Query-parameters:

args

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/appfwprofile_xmlattachmenturl_binding/name_value<String>?**args=xmlattachmenturl:<String_value>**

HTTP Method:DELETE

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue>

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error

get

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/appfwprofile_xmlattachmenturl_binding/name_value<String>

Query-parameters:

filter

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/appfwprofile_xmlattachmenturl_binding/name_value<String>?**filter=property-name1:property-value1,property-name2:property-value2**

Use this query-parameter to get the filtered set of appfwprofile_xmlattachmenturl_binding resources configured on Citrix ADC. Filtering can be done on any of the properties of the resource.

pagination

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/appfwprofile_xmlattachmenturl_binding/name_value<String>?**pagesize=#no;pageno=#no**

Use this query-parameter to get the appfwprofile_xmlattachmenturl_binding resources in chunks.

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "appfwprofile_xmlattachmenturl_binding": [ {

"xmlattachmenturl":<String_value>,

"name":<String_value>,
```

```
"xmlmaxattachmentsize":<Double_value>,

"xmlmaxattachmentsizecheck":<String_value>,

"state":<String_value>,

"xmlattachmentcontenttypecheck":<String_value>,

"comment":<String_value>,

"xmlattachmentcontenttype":<String_value>

}]}
```

get (all)

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/appfwprofile_xmlattachmenturl_binding

Query-parameters:

bulkbindings

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/appfwprofile_xmlattachmenturl_binding?**bulkbindings=yes**

NITRO allows you to fetch bindings in bulk.

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "appfwprofile_xmlattachmenturl_binding": [ {

"xmlattachmenturl":<String_value>,

"name":<String_value>,

"xmlmaxattachmentsize":<Double_value>,

"xmlmaxattachmentsizecheck":<String_value>,

"state":<String_value>,

"xmlattachmentcontenttypecheck":<String_value>,

"comment":<String_value>,

"xmlattachmentcontenttype":<String_value>

}]}
```

count

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/appfwprofile_xmlattachmenturl_binding/name_value<String>?**count=yes**

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{"appfwprofile_xmlattachmenturl_binding": [ { "__count": "#no"} ] }
```

appfwprofile_xmldosurl_binding

Binding object showing the xmldosurl that can be bound to appfwprofile.

Properties

(click to see Operations)

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
xmlmaxelementdepthcheck	<String>	Read-write	State if XML Max element depth check is ON or OFF. Possible values = ON, OFF
xmlmaxfilesize	<Double>	Read-write	Specify the maximum size of XML messages. Protects against overflow attacks.
xmlmaxnamespaceurilength	<Double>	Read-write	Specify the longest URI of any XML namespace. Protects against overflow attacks.
xmldosurl	<String>	Read-write	XML DoS URL regular expression length.
state	<String>	Read-write	Enabled. Possible values = ENABLED, DISABLED

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
xmlsoaparraycheck	<String>	Read-write	State if XML SOAP Array check is ON or OFF. Possible values = ON, OFF
xmlmaxelementnamelengthcheck	<String>	Read-write	State if XML Max element name length check is ON or OFF. Possible values = ON, OFF
xmlmaxelementscheck	<String>	Read-write	State if XML Max elements check is ON or OFF. Possible values = ON, OFF
xmlmaxentityexpansions	<Double>	Read-write	Specify maximum allowed number of entity expansions. Protects aganist Entity Expansion Attack.
xmlmaxattributes	<Double>	Read-write	Specify maximum number of attributes per XML element. Protects against overflow attacks.
xmlmaxfilesizecheck	<String>	Read-write	State if XML Max file size check is ON or OFF. Possible values = ON, OFF
xmlmaxchardatalength	<Double>	Read-write	Specify the maximum size of CDATA. Protects

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
			against overflow attacks and large quantities of unparsed data within XML messages.
xmlmaxnamespaces check	<String>	Read-write	State if XML Max namespaces check is ON or OFF. Possible values = ON, OFF
xmlmaxnamespaces	<Double>	Read-write	Specify maximum number of active namespaces. Protects against overflow attacks.
xmlmaxattributenam elengthcheck	<String>	Read-write	State if XML Max attribute name length check is ON or OFF. Possible values = ON, OFF
xmlblockdtd	<String>	Read-write	State if XML DTD is ON or OFF. Protects against recursive Document Type Declaration (DTD) entity expansion attacks. Also, SOAP messages cannot have DTDs in messages. . Possible values = ON, OFF
xmlmaxattributevalue length	<Double>	Read-write	Specify the longest value of any XML attribute. Protects

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
			against overflow attacks.
xmlmaxelementdepth	<Double>	Read-write	Maximum nesting (depth) of XML elements. This check protects against documents that have excessive hierarchy depths.
xmlmaxelementnamelength	<Double>	Read-write	Specify the longest name of any element (including the expanded namespace) to protect against overflow attacks.
name	<String>	Read-write	Name of the profile to which to bind an exemption or rule. Minimum length = 1
xmlblockpi	<String>	Read-write	State if XML Block PI is ON or OFF. Protects resources from denial of service attacks as SOAP messages cannot have processing instructions (PI) in messages. Possible values = ON, OFF
xmlmaxelementchildrencheck	<String>	Read-write	State if XML Max element children check is ON or OFF.

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
			Possible values = ON, OFF
xmlmaxelements	<Double>	Read-write	Specify the maximum number of XML elements allowed. Protects against overflow attacks.
xmlmaxentityexpansionscheck	<String>	Read-write	State if XML Max Entity Expansions Check is ON or OFF. Possible values = ON, OFF
xmlmaxnamespaceurilengthcheck	<String>	Read-write	State if XML Max namespace URI length check is ON or OFF. Possible values = ON, OFF
xmlmaxentityexpansiondepthcheck	<String>	Read-write	State if XML Max Entity Expansions Depth Check is ON or OFF. Possible values = ON, OFF
xmlmaxattributevaluelengthcheck	<String>	Read-write	State if XML Max atribute value length is ON or OFF. Possible values = ON, OFF
xmlmaxsoaparraysize	<Double>	Read-write	XML Max Total SOAP Array Size. Protects against

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
			SOAP Array Abuse attack.
xmlmaxentityexpansiondepth	<Double>	Read-write	Specify maximum entity expansion depth. Protects aganist Entity Expansion Attack.
xmlmaxnodescheck	<String>	Read-write	State if XML Max nodes check is ON or OFF. Possible values = ON, OFF
xmlmaxattributenamelength	<Double>	Read-write	Specify the longest name of any XML attribute. Protects against overflow attacks.
xmlmaxchardatalengthcheck	<String>	Read-write	State if XML Max CDATA length check is ON or OFF. Possible values = ON, OFF
xmlminfilesizecheck	<String>	Read-write	State if XML Min file size check is ON or OFF. Possible values = ON, OFF
xmlmaxelementchildren	<Double>	Read-write	Specify the maximum number of children allowed per XML element. Protects against overflow attacks.

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
xmlminfilesize	<Double>	Read-write	Enforces minimum message size.
xmlmaxnodes	<Double>	Read-write	Specify the maximum number of XML nodes. Protects against overflow attacks.
comment	<String>	Read-write	Any comments about the purpose of profile, or other useful information about the profile.
xmlmaxattributescheck	<String>	Read-write	State if XML Max attributes check is ON or OFF. Possible values = ON, OFF
xmlmaxsoaparrayrank	<Double>	Read-write	XML Max Individual SOAP Array Rank. This is the dimension of the SOAP array.
xmlblockexternalentities	<String>	Read-write	State if XML Block External Entities Check is ON or OFF. Protects against XML External Entity (XXE) attacks that force applications to parse untrusted external entities (sources) in XML documents. Possible values = ON, OFF

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
__count	<Double>	Read-write	count parameter

Operations

(click to see Properties)

ADD:| DELETE:| GET| GET (ALL)| COUNT

Some options that you can use for each operations:

- **Getting warnings in response:**NITRO allows you to get warnings in an operation by specifying the "warning" query parameter as "yes". For example, to get warnings while connecting to the Citrix ADC appliance, the URL is as follows:

  http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/login?warning=yes

  If any, the warnings are displayed in the response payload with the HTTP code "209 X-NITRO-WARNING".
- **Authenticated access for individual NITRO operations:**NITRO allows you to logon to the Citrix ADC appliance to perform individual operations. You can use this option instead of creating a NITRO session (using the login object) and then using that session to perform all operations,

  To do this, you must specify the username and password in the request header of the NITRO request as follows:

  X-NITRO-USER:<username>

  X-NITRO-PASS:<password>

  Note: In such cases, make sure that the request header DOES not include the following:

  Cookie:NITRO_AUTH_TOKEN=<tokenvalue>

`*Note:`*

Mandatory parameters are marked in redand placeholder content is marked in <green>.

add

URL:http://<Citrix-ADC-IP-address(NSIP)/nitro/v1/config/appfwprofile_xmldosurl_binding

HTTP Method:PUT

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Content-Type:application/json

Request Payload:

```
{

"appfwprofile_xmldosurl_binding":{

<b>"name":<String_value>,

</b>"comment":<String_value>,

"state":<String_value>,

"xmldosurl":<String_value>,

"xmlmaxelementdepthcheck":<String_value>,

"xmlmaxelementdepth":<Double_value>,

"xmlmaxelementnamelengthcheck":<String_value>,

"xmlmaxelementnamelength":<Double_value>,

"xmlmaxelementscheck":<String_value>,

"xmlmaxelements":<Double_value>,

"xmlmaxelementchildrencheck":<String_value>,

"xmlmaxelementchildren":<Double_value>,

"xmlmaxattributescheck":<String_value>,

"xmlmaxattributes":<Double_value>,

"xmlmaxattributenamelengthcheck":<String_value>,

"xmlmaxattributenamelength":<Double_value>,
```

```
"xmlmaxattributevaluelengthcheck":<String_value>,

"xmlmaxattributevaluelength":<Double_value>,

"xmlmaxchardatalengthcheck":<String_value>,

"xmlmaxchardatalength":<Double_value>,

"xmlmaxfilesizecheck":<String_value>,

"xmlmaxfilesize":<Double_value>,

"xmlminfilesizecheck":<String_value>,

"xmlminfilesize":<Double_value>,

"xmlblockpi":<String_value>,

"xmlblockdtd":<String_value>,

"xmlblockexternalentities":<String_value>,

"xmlmaxentityexpansionscheck":<String_value>,

"xmlmaxentityexpansions":<Double_value>,

"xmlmaxentityexpansiondepthcheck":<String_value>,

"xmlmaxentityexpansiondepth":<Double_value>,

"xmlmaxnamespacescheck":<String_value>,

"xmlmaxnamespaces":<Double_value>,

"xmlmaxnamespaceurilengthcheck":<String_value>,

"xmlmaxnamespaceurilength":<Double_value>,

"xmlsoaparraycheck":<String_value>,

"xmlmaxsoaparraysize":<Double_value>,

"xmlmaxsoaparrayrank":<Double_value>
```

```
}}
```

Response:

HTTP Status Code on Success: 201 Created HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error

delete

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/appfwprofile_xmldosurl_binding/name_value<String>

Query-parameters:

args

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/appfwprofile_xmldosurl_binding/name_value<String>?**args=xmldosurl:<String_value>**

HTTP Method:DELETE

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue>

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error

get

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/appfwprofile_xmldosurl_binding/name_value<String>

Query-parameters:

filter

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/appfwprofile_xmldosurl_binding/name_value<String>?**filter=property-name1:property-value1,property-name2:property-value2**

Use this query-parameter to get the filtered set of appfwprofile_xmldosurl_binding resources configured on Citrix ADC. Filtering can be done on any of the properties of the resource.

pagination

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/appfwprofile_xmldosurl_binding/name_value<String>?**pagesize=#no;pageno=#no**

Use this query-parameter to get the appfwprofile_xmldosurl_binding resources in chunks.

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "appfwprofile_xmldosurl_binding": [ {

"xmlmaxelementdepthcheck":<String_value>,

"xmlmaxfilesize":<Double_value>,

"xmlmaxnamespaceurilength":<Double_value>,

"xmldosurl":<String_value>,

"state":<String_value>,

"xmlsoaparraycheck":<String_value>,

"xmlmaxelementnamelengthcheck":<String_value>,
```

```
"xmlmaxelementscheck":<String_value>,

"xmlmaxentityexpansions":<Double_value>,

"xmlmaxattributes":<Double_value>,

"xmlmaxfilesizecheck":<String_value>,

"xmlmaxchardatalength":<Double_value>,

"xmlmaxnamespacescheck":<String_value>,

"xmlmaxnamespaces":<Double_value>,

"xmlmaxattributenamelengthcheck":<String_value>,

"xmlblockdtd":<String_value>,

"xmlmaxattributevaluelength":<Double_value>,

"xmlmaxelementdepth":<Double_value>,

"xmlmaxelementnamelength":<Double_value>,

"name":<String_value>,

"xmlblockpi":<String_value>,

"xmlmaxelementchildrencheck":<String_value>,

"xmlmaxelements":<Double_value>,

"xmlmaxentityexpansionscheck":<String_value>,

"xmlmaxnamespaceurilengthcheck":<String_value>,

"xmlmaxentityexpansiondepthcheck":<String_value>,

"xmlmaxattributevaluelengthcheck":<String_value>,

"xmlmaxsoaparraysize":<Double_value>,

"xmlmaxentityexpansiondepth":<Double_value>,
```

```
"xmlmaxnodescheck":<String_value>,

"xmlmaxattributenamelength":<Double_value>,

"xmlmaxchardatalengthcheck":<String_value>,

"xmlminfilesizecheck":<String_value>,

"xmlmaxelementchildren":<Double_value>,

"xmlminfilesize":<Double_value>,

"xmlmaxnodes":<Double_value>,

"comment":<String_value>,

"xmlmaxattributescheck":<String_value>,

"xmlmaxsoaparrayrank":<Double_value>,

"xmlblockexternalentities":<String_value>

}]}
```

get (all)

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/appfwprofile_xmldosurl_binding

Query-parameters:

bulkbindings

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/appfwprofile_xmldosurl_binding?**bulkbindings=yes**

NITRO allows you to fetch bindings in bulk.

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "appfwprofile_xmldosurl_binding": [ {

"xmlmaxelementdepthcheck":<String_value>,

"xmlmaxfilesize":<Double_value>,

"xmlmaxnamespaceurilength":<Double_value>,

"xmldosurl":<String_value>,

"state":<String_value>,

"xmlsoaparraycheck":<String_value>,

"xmlmaxelementnamelengthcheck":<String_value>,

"xmlmaxelementscheck":<String_value>,

"xmlmaxentityexpansions":<Double_value>,

"xmlmaxattributes":<Double_value>,

"xmlmaxfilesizecheck":<String_value>,

"xmlmaxchardatalength":<Double_value>,

"xmlmaxnamespacescheck":<String_value>,

"xmlmaxnamespaces":<Double_value>,

"xmlmaxattributenamelengthcheck":<String_value>,
```

```
"xmlblockdtd":<String_value>,

"xmlmaxattributevaluelength":<Double_value>,

"xmlmaxelementdepth":<Double_value>,

"xmlmaxelementnamelength":<Double_value>,

"name":<String_value>,

"xmlblockpi":<String_value>,

"xmlmaxelementchildrencheck":<String_value>,

"xmlmaxelements":<Double_value>,

"xmlmaxentityexpansionscheck":<String_value>,

"xmlmaxnamespaceurilengthcheck":<String_value>,

"xmlmaxentityexpansiondepthcheck":<String_value>,

"xmlmaxattributevaluelengthcheck":<String_value>,

"xmlmaxsoaparraysize":<Double_value>,

"xmlmaxentityexpansiondepth":<Double_value>,

"xmlmaxnodescheck":<String_value>,

"xmlmaxattributenamelength":<Double_value>,

"xmlmaxchardatalengthcheck":<String_value>,

"xmlminfilesizecheck":<String_value>,

"xmlmaxelementchildren":<Double_value>,

"xmlminfilesize":<Double_value>,

"xmlmaxnodes":<Double_value>,

"comment":<String_value>,
```

```
"xmlmaxattributescheck":<String_value>,

"xmlmaxsoaparrayrank":<Double_value>,

"xmlblockexternalentities":<String_value>

}]}
```

count

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/appfwprofile_xmldosurl_binding/name_value<String>?**count=yes**

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{"appfwprofile_xmldosurl_binding": [ { "__count": "#no"} ] }
```

appfwprofile_xmlsqlinjection_binding

Binding object showing the xmlsqlinjection that can be bound to appfwprofile.

Properties

(click to see Operations)

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
as_scan_location_xmlsql	<String>	Read-write	Location of SQL injection exception - XML Element or Attribute. Possible values = ELEMENT, ATTRIBUTE
state	<String>	Read-write	Enabled. Possible values = ENABLED, DISABLED
name	<String>	Read-write	Name of the profile to which to bind an exemption or rule. Minimum length = 1
xmlsqlinjection	<String>	Read-write	Exempt the specified URL from the XML SQL injection check. An XML SQL injection exemption (relaxation) consists of the following items: * Name. Name to exempt, as a string or a PCRE-format regular expression. * ISREGEX flag. REGEX if URL is a regular expression, NOTREGEX if URL is a fixed string. * Location. ELEMENT if the injection is located in an XML element, ATTRIBUTE if

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
			located in an XML attribute.
isregex_xmlsql	<String>	Read-write	Is the XML SQL Injection exempted field name a regular expression?. Possible values = REGEX, NOTREGEX
comment	<String>	Read-write	Any comments about the purpose of profile, or other useful information about the profile.
__count	<Double>	Read-write	count parameter

Operations

(click to see Properties)

ADD:| DELETE:| GET| GET (ALL)| COUNT

Some options that you can use for each operations:

- **Getting warnings in response:**NITRO allows you to get warnings in an operation by specifying the "warning" query parameter as "yes". For example, to get warnings while connecting to the Citrix ADC appliance, the URL is as follows:

  http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/login?warning=yes

  If any, the warnings are displayed in the response payload with the HTTP code "209 X-NITRO-WARNING".
- **Authenticated access for individual NITRO operations:**NITRO allows you to logon to the Citrix ADC appliance to perform individual operations. You can use this option instead of creating a NITRO session (using the login object) and then using that session to perform all operations,

To do this, you must specify the username and password in the request header of the NITRO request as follows:

X-NITRO-USER:<username>

X-NITRO-PASS:<password>

Note: In such cases, make sure that the request header DOES not include the following:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue>

`*Note:`*

Mandatory parameters are marked in redand placeholder content is marked in <green>.

add

URL:http://<Citrix-ADC-IP-address(NSIP)/nitro/v1/config/appfwprofile_xmlsqlinjection_binding

HTTP Method:PUT

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Content-Type:application/json

Request Payload:

```
{

"appfwprofile_xmlsqlinjection_binding":{

<b>"name":<String_value>,

</b>"comment":<String_value>,

"state":<String_value>,

"xmlsqlinjection":<String_value>,

"isregex_xmlsql":<String_value>,

"as_scan_location_xmlsql":<String_value>
```

```
}}
```

Response:

HTTP Status Code on Success: 201 Created HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error

delete

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/appfwprofile_xmlsqlinjection_binding/name_value<String>

Query-parameters:

args

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/appfwprofile_xmlsqlinjection_binding/name_value<String>?
args=xmlsqlinjection:<String_value>,as_scan_location_xmlsql:<String_value>

HTTP Method:DELETE

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue>

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error

get

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/appfwprofile_xmlsqlinjection_binding/name_value<String>

Query-parameters:

filter

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/appfwprofile_xmlsqlinjection_binding/name_value<String>?**filter=property-name1:property-value1,property-name2:property-value2**

Use this query-parameter to get the filtered set of appfwprofile_xmlsqlinjection_binding resources configured on Citrix ADC. Filtering can be done on any of the properties of the resource.

pagination

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/appfwprofile_xmlsqlinjection_binding/name_value<String>?**pagesize=#no;pageno=#no**

Use this query-parameter to get the appfwprofile_xmlsqlinjection_binding resources in chunks.

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "appfwprofile_xmlsqlinjection_binding": [ {

"as_scan_location_xmlsql":<String_value>,

"state":<String_value>,

"name":<String_value>,

"xmlsqlinjection":<String_value>,

"isregex_xmlsql":<String_value>,

"comment":<String_value>
```

```
}]}
```

get (all)

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/appfwprofile_xmlsqlinjection_binding

Query-parameters:

bulkbindings

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/appfwprofile_xmlsqlinjection_binding?**bulkbindings=yes**

NITRO allows you to fetch bindings in bulk.

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "appfwprofile_xmlsqlinjection_binding": [ {

"as_scan_location_xmlsql":<String_value>,

"state":<String_value>,

"name":<String_value>,

"xmlsqlinjection":<String_value>,
```

```
"isregex_xmlsql":<String_value>,

"comment":<String_value>

}]}
```

count

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/appfwprofile_xmlsqlinjection_binding/name_value<String>?**count=yes**

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{"appfwprofile_xmlsqlinjection_binding": [ { "__count": "#no"} ] }
```

appfwprofile_xmlvalidationurl_binding

Binding object showing the xmlvalidationurl that can be bound to appfwprofile.

Properties

(click to see Operations)

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
state	<String>	Read-write	Enabled. Possible values = ENABLED, DISABLED
xmlwsdl	<String>	Read-write	WSDL object for soap request validation.
xmlendpointcheck	<String>	Read-write	Modifies the behaviour of the Request URL validation w.r.t. the Service URL. If set to ABSOLUTE, the entire request URL is validated with the entire URL mentioned in Service of the associated WSDL. eg: Service URL: http://example.org/ExampleService, Request URL: http//example.com/ExampleService would FAIL the validation. If set to RELAIVE, only the non-hostname part of the request URL is validated against the non-hostname part of the Service URL. eg: Service URL: http://example.org/ExampleService, Request URL: http//example.com/

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
			ExampleService would PASS the validation. Possible values = ABSOLUTE, RELATIVE
name	<String>	Read-write	Name of the profile to which to bind an exemption or rule. Minimum length = 1
xmlvalidateresponse	<String>	Read-write	Validate response message. Possible values = ON, OFF
xmlvalidationurl	<String>	Read-write	XML Validation URL regular expression.
xmlresponseschema	<String>	Read-write	XML Schema object for response validation.
xmlvalidatesoapenvelope	<String>	Read-write	Validate SOAP Evelope only. Possible values = ON, OFF
xmlrequestschema	<String>	Read-write	XML Schema object for request validation .
xmladditionalsoapheaders	<String>	Read-write	Allow addtional soap headers. Possible values = ON, OFF
comment	<String>	Read-write	Any comments about the purpose of

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
			profile, or other useful information about the profile.
__count	<Double>	Read-write	count parameter

Operations

(click to see Properties)

ADD:| DELETE:| GET| GET (ALL)| COUNT

Some options that you can use for each operations:

- **Getting warnings in response:**NITRO allows you to get warnings in an operation by specifying the "warning" query parameter as "yes". For example, to get warnings while connecting to the Citrix ADC appliance, the URL is as follows:

  http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/login?warning=yes

  If any, the warnings are displayed in the response payload with the HTTP code "209 X-NITRO-WARNING".

- **Authenticated access for individual NITRO operations:**NITRO allows you to logon to the Citrix ADC appliance to perform individual operations. You can use this option instead of creating a NITRO session (using the login object) and then using that session to perform all operations,

  To do this, you must specify the username and password in the request header of the NITRO request as follows:

  X-NITRO-USER:<username>

  X-NITRO-PASS:<password>

  Note: In such cases, make sure that the request header DOES not include the following:

  Cookie:NITRO_AUTH_TOKEN=<tokenvalue>

`*Note:`*

Mandatory parameters are marked in redand placeholder content is marked in <green>.

add

URL:http://<Citrix-ADC-IP-address(NSIP)/nitro/v1/config/appfwprofile_xmlvalidationurl_binding

HTTP Method:PUT

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Content-Type:application/json

Request Payload:

```
{

"appfwprofile_xmlvalidationurl_binding":{

<b>"name":<String_value>,

</b>"comment":<String_value>,

"state":<String_value>,

"xmlvalidationurl":<String_value>,

"xmlrequestschema":<String_value>,

"xmlresponseschema":<String_value>,

"xmlwsdl":<String_value>,

"xmladditionalsoapheaders":<String_value>,

"xmlendpointcheck":<String_value>,

"xmlvalidatesoapenvelope":<String_value>,

"xmlvalidateresponse":<String_value>

}}
```

Response:

HTTP Status Code on Success: 201 Created HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error

delete

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/appfwprofile_xmlvalidationurl_binding/name_value<String>

Query-parameters:

args

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/appfwprofile_xmlvalidationurl_binding/name_value<String>?**args=xmlvalidationurl:<String_value>**

HTTP Method:DELETE

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue>

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error

get

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/appfwprofile_xmlvalidationurl_binding/name_value<String>

Query-parameters:

filter

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/appfwprofile_xmlvalidationurl_binding/name_value<String>?**filter=property-name1:property-value1,property-name2:property-value2**

Use this query-parameter to get the filtered set of appfwprofile_xmlvalidationurl_binding resources configured on Citrix ADC. Filtering can be done on any of the properties of the resource.

pagination

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/appfwprofile_xmlvalidationurl_binding/name_value<String>?**pagesize=#no;pageno=#no**

Use this query-parameter to get the appfwprofile_xmlvalidationurl_binding resources in chunks.

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "appfwprofile_xmlvalidationurl_binding": [ {

"state":<String_value>,

"xmlwsdl":<String_value>,

"xmlendpointcheck":<String_value>,

"name":<String_value>,

"xmlvalidateresponse":<String_value>,

"xmlvalidationurl":<String_value>,

"xmlresponseschema":<String_value>,

"xmlvalidatesoapenvelope":<String_value>,

"xmlrequestschema":<String_value>,

"xmladditionalsoapheaders":<String_value>,
```

```
"comment":<String_value>

}]}
```

get (all)

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/appfwprofile_xmlvalidationurl_binding

Query-parameters:

bulkbindings

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/appfwprofile_xmlvalidationurl_binding?**bulkbindings=yes**

NITRO allows you to fetch bindings in bulk.

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "appfwprofile_xmlvalidationurl_binding": [ {

"state":<String_value>,

"xmlwsdl":<String_value>,

"xmlendpointcheck":<String_value>,
```

```
"name":<String_value>,

"xmlvalidateresponse":<String_value>,

"xmlvalidationurl":<String_value>,

"xmlresponseschema":<String_value>,

"xmlvalidatesoapenvelope":<String_value>,

"xmlrequestschema":<String_value>,

"xmladditionalsoapheaders":<String_value>,

"comment":<String_value>

}]}
```

count

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/appfwprofile_xmlvalidationurl_binding/name_value<String>?**count=yes**

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{"appfwprofile_xmlvalidationurl_binding": [ { "__count": "#no"} ] }
```

appfwprofile_xmlwsiurl_binding

Binding object showing the xmlwsiurl that can be bound to appfwprofile.

Properties

(click to see Operations)

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
xmlwsichecks	<String>	Read-write	Specify a comma separated list of relevant WS-I rule IDs. (R1140, R1141).
name	<String>	Read-write	Name of the profile to which to bind an exemption or rule. Minimum length = 1
xmlwsiurl	<String>	Read-write	XML WS-I URL regular expression length.
state	<String>	Read-write	Enabled. Possible values = ENABLED, DISABLED
comment	<String>	Read-write	Any comments about the purpose of profile, or other useful information about the profile.
__count	<Double>	Read-write	count parameter

Operations

(click to see Properties)

ADD:| DELETE:| GET| GET (ALL)| COUNT

Some options that you can use for each operations:

- **Getting warnings in response:**NITRO allows you to get warnings in an operation by specifying the "warning" query parameter as "yes". For example, to get warnings while connecting to the Citrix ADC appliance, the URL is as follows:

  http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/login?warning=yes

  If any, the warnings are displayed in the response payload with the HTTP code "209 X-NITRO-WARNING".

- **Authenticated access for individual NITRO operations:**NITRO allows you to logon to the Citrix ADC appliance to perform individual operations. You can use this option instead of creating a NITRO session (using the login object) and then using that session to perform all operations,

  To do this, you must specify the username and password in the request header of the NITRO request as follows:

  X-NITRO-USER:<username>

  X-NITRO-PASS:<password>

  Note: In such cases, make sure that the request header DOES not include the following:

  Cookie:NITRO_AUTH_TOKEN=<tokenvalue>

`*Note:`*

Mandatory parameters are marked in redand placeholder content is marked in <green>.

add

URL:http://<Citrix-ADC-IP-address(NSIP)/nitro/v1/config/appfwprofile_xmlwsiurl_binding

HTTP Method:PUT

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Content-Type:application/json

Request Payload:

```
{

"appfwprofile_xmlwsiurl_binding":{

<b>"name":<String_value>,

</b>"comment":<String_value>,

"state":<String_value>,

"xmlwsiurl":<String_value>,

"xmlwsichecks":<String_value>

}}
```

Response:

HTTP Status Code on Success: 201 Created HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error

delete

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/appfwprofile_xmlwsiurl_binding/name_value<String>

Query-parameters:

args

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/appfwprofile_xmlwsiurl_binding/name_value<String>?**args=xmlwsiurl:<String_value>**

HTTP Method:DELETE

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue>

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error

get

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/appfwprofile_xmlwsiurl_binding/name_value<String>

Query-parameters:

filter

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/appfwprofile_xmlwsiurl_binding/name_value<String>?**filter=property-name1:property-value1,property-name2:property-value2**

Use this query-parameter to get the filtered set of appfwprofile_xmlwsiurl_binding resources configured on Citrix ADC. Filtering can be done on any of the properties of the resource.

pagination

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/appfwprofile_xmlwsiurl_binding/name_value<String>?**pagesize=#no;pageno=#no**

Use this query-parameter to get the appfwprofile_xmlwsiurl_binding resources in chunks.

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "appfwprofile_xmlwsiurl_binding": [ {
```

```
"xmlwsichecks":<String_value>,

"name":<String_value>,

"xmlwsiurl":<String_value>,

"state":<String_value>,

"comment":<String_value>

}]}
```

get (all)

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/appfwprofile_xmlwsiurl_binding

Query-parameters:

bulkbindings

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/appfwprofile_xmlwsiurl_binding?**bulkbindings=yes**

NITRO allows you to fetch bindings in bulk.

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "appfwprofile_xmlwsiurl_binding": [ {

"xmlwsichecks":<String_value>,

"name":<String_value>,

"xmlwsiurl":<String_value>,

"state":<String_value>,

"comment":<String_value>

}]}
```

count

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/appfwprofile_xmlwsiurl_binding/name_value<String>?**count=yes**

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{"appfwprofile_xmlwsiurl_binding": [ { "__count": "#no"} ] }
```

appfwprofile_xmlxss_binding

Binding object showing the xmlxss that can be bound to appfwprofile.

Properties

(click to see Operations)

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
state	<String>	Read-write	Enabled. Possible values = ENABLED, DISABLED
name	<String>	Read-write	Name of the profile to which to bind an exemption or rule. Minimum length = 1
as_scan_location_xmlxss	<String>	Read-write	Location of XSS injection exception - XML Element or Attribute. Possible values = ELEMENT, ATTRIBUTE
isregex_xmlxss	<String>	Read-write	Is the XML XSS exempted field name a regular expression?. Possible values = REGEX, NOTREGEX
comment	<String>	Read-write	Any comments about the purpose of profile, or other useful information about the profile.

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
xmlxss	<String>	Read-write	Exempt the specified URL from the XML cross-site scripting (XSS) check. An XML cross-site scripting exemption (relaxation) consists of the following items: * URL. URL to exempt, as a string or a PCRE-format regular expression. * ISREGEX flag. REGEX if URL is a regular expression, NOTREGEX if URL is a fixed string. * Location. ELEMENT if the attachment is located in an XML element, ATTRIBUTE if located in an XML attribute.
__count	<Double>	Read-write	count parameter

Operations

(click to see Properties)

ADD:| DELETE:| GET| GET (ALL)| COUNT

Some options that you can use for each operations:

- **Getting warnings in response:**NITRO allows you to get warnings in an operation by specifying the "warning" query parameter as "yes". For example, to get warnings while connecting to the Citrix ADC appliance, the URL is as follows:

  http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/login?warning=yes

If any, the warnings are displayed in the response payload with the HTTP code "209 X-NITRO-WARNING".

- **Authenticated access for individual NITRO operations:**NITRO allows you to logon to the Citrix ADC appliance to perform individual operations. You can use this option instead of creating a NITRO session (using the login object) and then using that session to perform all operations,

  To do this, you must specify the username and password in the request header of the NITRO request as follows:

  X-NITRO-USER:<username>

  X-NITRO-PASS:<password>

  Note: In such cases, make sure that the request header DOES not include the following:

  Cookie:NITRO_AUTH_TOKEN=<tokenvalue>

`*Note:`*

Mandatory parameters are marked in redand placeholder content is marked in <green>.

add

URL:http://<Citrix-ADC-IP-address(NSIP)/nitro/v1/config/appfwprofile_xmlxss_binding

HTTP Method:PUT

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Content-Type:application/json

Request Payload:

```
{

"appfwprofile_xmlxss_binding":{

<b>"name":<String_value>,

</b>"comment":<String_value>,
```

```
"state":<String_value>,

"xmlxss":<String_value>,

"isregex_xmlxss":<String_value>,

"as_scan_location_xmlxss":<String_value>

}}
```

Response:

HTTP Status Code on Success: 201 Created HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error

delete

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/appfwprofile_xmlxss_binding/name_value<String>

Query-parameters:

args

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/appfwprofile_xmlxss_binding/name_value<String>?**args=xmlxss:<String_value>,as_scan_location_xmlxss:<String_value>**

HTTP Method:DELETE

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue>

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error

get

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/appfwprofile_xmlxss_binding/name_value<String>

Query-parameters:

filter

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/appfwprofile_xmlxss_binding/name_value<String>?**filter=property-name1:property-value1,property-name2:property-value2**

Use this query-parameter to get the filtered set of appfwprofile_xmlxss_binding resources configured on Citrix ADC. Filtering can be done on any of the properties of the resource.

pagination

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/appfwprofile_xmlxss_binding/name_value<String>?**pagesize=#no;pageno=#no**

Use this query-parameter to get the appfwprofile_xmlxss_binding resources in chunks.

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "appfwprofile_xmlxss_binding": [ {

"state":<String_value>,

"name":<String_value>,
```

```
"as_scan_location_xmlxss":<String_value>,

"isregex_xmlxss":<String_value>,

"comment":<String_value>,

"xmlxss":<String_value>

}]}
```

get (all)

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/appfwprofile_xmlxss_binding

Query-parameters:

bulkbindings

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/appfwprofile_xmlxss_binding?**bulkbindings=yes**

NITRO allows you to fetch bindings in bulk.

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "appfwprofile_xmlxss_binding": [ {
```

```
"state":<String_value>,

"name":<String_value>,

"as_scan_location_xmlxss":<String_value>,

"isregex_xmlxss":<String_value>,

"comment":<String_value>,

"xmlxss":<String_value>

}]}
```

count

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/appfwprofile_xmlxss_binding/name_value<String>?**count=yes**

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{"appfwprofile_xmlxss_binding": [ { "__count": "#no"} ] }
```

appfwsettings

Configuration for AS settings resource.

Properties

(click to see Operations)

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
defaultprofile	<String>	Read-write	Profile to use when a connection does not match any policy. Default setting is APPFW_BYPASS, which sends unmatched connections back to the Citrix ADC without attempting to filter them further. Default value: APPFW_BYPASS Minimum length = 1
undefaction	<String>	Read-write	Profile to use when an application firewall policy evaluates to undefined (UNDEF). An UNDEF event indicates an internal error condition. The APPFW_BLOCK built-in profile is the default setting. You can specify a different built-in or user-created profile

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
			as the UNDEF profile. Default value: APPFW_BLOCK Minimum length = 1
sessiontimeout	<Double>	Read-write	Timeout, in seconds, after which a user session is terminated. Before continuing to use the protected web site, the user must establish a new session by opening a designated start URL. Default value: 900 Minimum value = 1 Maximum value = 65535
learnratelimit	<Double>	Read-write	Maximum number of connections per second that the application firewall learning engine examines to generate new relaxations for learning-enabled security checks. The application firewall drops any connections above this limit from the list of connections used

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
			by the learning engine. Default value: 400 Minimum value = 1 Maximum value = 1000
sessionlifetime	<Double>	Read-write	Maximum amount of time (in seconds) that the application firewall allows a user session to remain active, regardless of user activity. After this time, the user session is terminated. Before continuing to use the protected web site, the user must establish a new session by opening a designated start URL. Default value: 0 Minimum value = 0 Maximum value = 2147483647
sessioncookiename	<String>	Read-write	Name of the session cookie that the application firewall uses to track user sessions. Must begin with a letter or number, and can consist of

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
			from 1 to 31 letters, numbers, and the hyphen (-) and underscore (_) symbols. The following requirement applies only to the Citrix ADC CLI: If the name includes one or more spaces, enclose the name in double or single quotation marks (for example, "my cookie name" or 'my cookie name'). Minimum length = 1
clientiploggingheader	<String>	Read-write	Name of an HTTP header that contains the IP address that the client used to connect to the protected web site or service.
importsizelimit	<Double>	Read-write	Cumulative total maximum number of bytes in web forms imported to a protected web site. If a user attempts to upload files with a total byte count higher than the

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
			specified limit, the application firewall blocks the request. Default value: 134217728 Minimum value = 1 Maximum value = 268435456
signatureautoupdate	<String>	Read-write	Flag used to enable/disable auto update signatures. Default value: OFF Possible values = ON, OFF
signatureurl	<String>	Read-write	URL to download the mapping file from server. Default value: https://s3.amazonaws.com/NSAppFwSignatures/SignaturesMapping.xml
cookiepostencryptprefix	<String>	Read-write	String that is prepended to all encrypted cookie values. Minimum length = 1
logmalformedreq	<String>	Read-write	Log requests that are so malformed that application firewall

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
			parsing doesn't occur. Default value: ON Possible values = ON, OFF
geolocationlogging	<String>	Read-write	Enable Geo-Location Logging in CEF format logs. Default value: OFF Possible values = ON, OFF
ceflogging	<String>	Read-write	Enable CEF format logs. Default value: OFF Possible values = ON, OFF
entitydecoding	<String>	Read-write	Transform multibyte (double- or half-width) characters to single width characters. Default value: OFF Possible values = ON, OFF
useconfigurablesecretkey	<String>	Read-write	Use configurable secret key in AppFw operations. Default value: OFF Possible values = ON, OFF

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
sessionlimit	<Double>	Read-write	Maximum number of sessions that the application firewall allows to be active, regardless of user activity. After the max_limit reaches, No more user session will be created . Default value: 100000 Minimum value = 0 Maximum value = 500000
malformedreqaction	<String[]>	Read-write	flag to define action on malformed requests that application firewall cannot parse. Possible values = none, block, log, stats

Operations

(click to see Properties)

UPDATE| UNSET| GET (ALL)

Some options that you can use for each operations:

- **Getting warnings in response:**NITRO allows you to get warnings in an operation by specifying the "warning" query parameter as "yes". For example, to get warnings while connecting to the Citrix ADC appliance, the URL is as follows:

  http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/login?warning=yes

If any, the warnings are displayed in the response payload with the HTTP code "209 X-NITRO-WARNING".

- **Authenticated access for individual NITRO operations:**NITRO allows you to logon to the Citrix ADC appliance to perform individual operations. You can use this option instead of creating a NITRO session (using the login object) and then using that session to perform all operations,

  To do this, you must specify the username and password in the request header of the NITRO request as follows:

  X-NITRO-USER:<username>

  X-NITRO-PASS:<password>

  Note: In such cases, make sure that the request header DOES not include the following:

  Cookie:NITRO_AUTH_TOKEN=<tokenvalue>

`*Note:`*

Mandatory parameters are marked in redand placeholder content is marked in <green>.

update

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/appfwsettings

HTTP Method:PUT

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Content-Type:application/json

Request Payload:

```
{"appfwsettings":{

"defaultprofile":<String_value>,

"undefaction":<String_value>,

"sessiontimeout":<Double_value>,
```

```
"learnratelimit":<Double_value>,

"sessionlifetime":<Double_value>,

"sessioncookiename":<String_value>,

"clientiploggingheader":<String_value>,

"importsizelimit":<Double_value>,

"signatureautoupdate":<String_value>,

"signatureurl":<String_value>,

"cookiepostencryptprefix":<String_value>,

"logmalformedreq":<String_value>,

"geolocationlogging":<String_value>,

"ceflogging":<String_value>,

"entitydecoding":<String_value>,

"useconfigurablesecretkey":<String_value>,

"sessionlimit":<Double_value>,

"malformedreqaction":<String[]_value>

}}
```

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error

unset

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/appfwsettings?**action=unset**

HTTP Method:POST

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Content-Type:application/json

Request Payload:

```
{"appfwsettings":{

"defaultprofile":true,

"undefaction":true,

"sessiontimeout":true,

"learnratelimit":true,

"sessionlifetime":true,

"sessioncookiename":true,

"clientiploggingheader":true,

"importsizelimit":true,

"signatureautoupdate":true,

"signatureurl":true,

"cookiepostencryptprefix":true,

"logmalformedreq":true,

"geolocationlogging":true,

"ceflogging":true,

"entitydecoding":true,

"useconfigurablesecretkey":true,
```

```
"sessionlimit":true,

"malformedreqaction":true

}}
```

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error

get (all)

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/appfwsettings

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "appfwsettings": [ {

"defaultprofile":<String_value>,

"undefaction":<String_value>,

"sessiontimeout":<Double_value>,
```

```
"learnratelimit":<Double_value>,

"sessionlifetime":<Double_value>,

"sessioncookiename":<String_value>,

"clientiploggingheader":<String_value>,

"importsizelimit":<Double_value>,

"signatureautoupdate":<String_value>,

"signatureurl":<String_value>,

"cookiepostencryptprefix":<String_value>,

"logmalformedreq":<String_value>,

"geolocationlogging":<String_value>,

"ceflogging":<String_value>,

"entitydecoding":<String_value>,

"useconfigurablesecretkey":<String_value>,

"sessionlimit":<Double_value>,

"malformedreqaction":<String[]_value>

}]}
```

appfwsignatures

Configuration for application firewall signatures XML configuration resource.

Properties

(click to see Operations)

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
name	<String>	Read-write	Name of the signature object. Minimum length = 1 Maximum length = 31
src	<String>	Read-write	URL (protocol, host, path, and file name) for the location at which to store the imported signatures object. NOTE: The import fails if the object to be imported is on an HTTPS server that requires client certificate authentication for access. Minimum length = 1 Maximum length = 2047
xslt	<String>	Read-write	XSLT file source. Maximum length = 2047
comment	<String>	Read-write	Any comments to preserve information about the signatures object. Maximum length = 128

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
overwrite	<Boolean>	Read-write	Overwrite any existing signatures object of the same name.
merge	<Boolean>	Read-write	Merges the existing Signature with new signature rules.
preservedefactions	<Boolean>	Read-write	preserves def actions of signature rules.
sha1	<String>	Read-write	File path for sha1 file to validate signature file. Minimum length = 1 Maximum length = 2047
mergedefault	<Boolean>	Read-write	Merges signature file with default signature file.
response	<String>	Read-only	.

Operations

(click to see Properties)

DELETE| IMPORT| CHANGE| GET (ALL)| GET

Some options that you can use for each operations:

- **Getting warnings in response:**NITRO allows you to get warnings in an operation by specifying the "warning" query parameter as "yes". For example, to get warnings while connecting to the Citrix ADC appliance, the URL is as follows:

  http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/login?warning=yes

If any, the warnings are displayed in the response payload with the HTTP code "209 X-NITRO-WARNING".

- **Authenticated access for individual NITRO operations:**NITRO allows you to logon to the Citrix ADC appliance to perform individual operations. You can use this option instead of creating a NITRO session (using the login object) and then using that session to perform all operations,

  To do this, you must specify the username and password in the request header of the NITRO request as follows:

  X-NITRO-USER:<username>

  X-NITRO-PASS:<password>

  Note: In such cases, make sure that the request header DOES not include the following:

  Cookie:NITRO_AUTH_TOKEN=<tokenvalue>

`*Note:`*

Mandatory parameters are marked in redand placeholder content is marked in <green>.

delete

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/appfwsignatures/name_value<String>

HTTP Method:DELETE

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue>

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error

Import

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/appfwsignatures?**action=Import**

HTTP Method:POST

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Content-Type:application/json

Request Payload:

```
{"appfwsignatures":{

<b>"src":<String_value>,

</b><b>"name":<String_value>,

</b>"xslt":<String_value>,

"comment":<String_value>,

"overwrite":<Boolean_value>,

"merge":<Boolean_value>,

"preservedefactions":<Boolean_value>,

"sha1":<String_value>

}}
```

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error

change

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/appfwsignatures?**action=update**

HTTP Method:POST

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Content-Type:application/json

Request Payload:

```
{"appfwsignatures":{

<b>"name":<String_value>,

</b>"mergedefault":<Boolean_value>

}}
```

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error

get (all)

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/appfwsignatures

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "appfwsignatures": [ {

"name":<String_value>,

"response":<String_value>,
```

```
"src":<String_value>

}]}
```

get

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/appfwsignatures/name_value<String>

Query-parameters:

attrs

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/appfwsignatures/name_value<String>?**attrs=property-name1,property-name2**

Use this query parameter to specify the resource details that you want to retrieve.

view

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/appfwsignatures/name_value<String>?**view=summary**

Note: By default, the retrieved results are displayed in detail view (?view=detail).

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "appfwsignatures": [ {
```

```
"name":<String_value>,

"response":<String_value>,

"src":<String_value>

}]}
```

appfwtransactionrecords

Configuration for Application firewall transaction record resource.

Properties

(click to see Operations)

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
nodeid	<Double>	Read-write	Unique number that identifies the cluster node. Minimum value = 0 Maximum value = 31
httptransactionid	<Double>	Read-only	The http transaction identifier.
packetengineid	<Integer>	Read-only	The packet engine identifier.
appfwsessionid	<String>	Read-only	The session identifier set by the Application Firewall to track the user session.
profilename	<String>	Read-only	Application Firewall profile name.
url	<String>	Read-only	Request URL.

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
clientip	<String>	Read-only	The IP address of client.
destip	<String>	Read-only	The IP address of destination.
starttime	<String>	Read-only	Conveys time at which request processing started.
endtime	<String>	Read-only	Conveys time at which request processing end.
requestcontentlength	<Integer>	Read-only	The content length of request.
requestyields	<Double>	Read-only	The number of times yielded during request processing to send heart beat packets.
requestmaxprocessingtime	<Double>	Read-only	The maximum processing time across yields during request processing.
responsecontentlength	<Integer>	Read-only	The content length of response.
responseyields	<Double>	Read-only	The number of times yielded during response processing to send heart beat packets.
responsemaxprocessingtime	<Double>	Read-only	The maximum processing time across yields during

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
			response processing.
__count	<Double>	Read-only	count parameter

Operations

(click to see Properties)

GET (ALL)| COUNT

Some options that you can use for each operations:

- **Getting warnings in response:**NITRO allows you to get warnings in an operation by specifying the "warning" query parameter as "yes". For example, to get warnings while connecting to the Citrix ADC appliance, the URL is as follows:

  http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/login?warning=yes

  If any, the warnings are displayed in the response payload with the HTTP code "209 X-NITRO-WARNING".
- **Authenticated access for individual NITRO operations:**NITRO allows you to logon to the Citrix ADC appliance to perform individual operations. You can use this option instead of creating a NITRO session (using the login object) and then using that session to perform all operations,

  To do this, you must specify the username and password in the request header of the NITRO request as follows:

  X-NITRO-USER:<username>

  X-NITRO-PASS:<password>

  Note: In such cases, make sure that the request header DOES not include the following:

  Cookie:NITRO_AUTH_TOKEN=<tokenvalue>

`*Note:`*

Mandatory parameters are marked in redand placeholder content is marked in <green>.

get (all)

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/appfwtransactionrecords

Query-parameters:

args

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/appfwtransactionrecords?**args=nodeid:<Double_value>**

Use this query-parameter to get appfwtransactionrecords resources based on additional properties.

attrs

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/appfwtransactionrecords?**attrs=property-name1,property-name2**

Use this query parameter to specify the resource details that you want to retrieve.

filter

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/appfwtransactionrecords?**filter=property-name1:property-val1,property-name2:property-val2**

Use this query-parameter to get the filtered set of appfwtransactionrecords resources configured on Citrix ADC. Filtering can be done on any of the properties of the resource.

view

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/appfwtransactionrecords?**view=summary**

Note: By default, the retrieved results are displayed in detail view (?view=detail).

pagination

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/appfwtransactionrecords?**pagesize=#no;pageno=#no**

Use this query-parameter to get the appfwtransactionrecords resources in chunks.

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "appfwtransactionrecords": [ {

nodeid:<Double_value>"httptransactionid":<Double_value>,

"packetengineid":<Integer_value>,

"appfwsessionid":<String_value>,

"profilename":<String_value>,

"url":<String_value>,

"clientip":<String_value>,

"destip":<String_value>,

"starttime":<String_value>,

"endtime":<String_value>,

"requestcontentlength":<Integer_value>,

"requestyields":<Double_value>,

"requestmaxprocessingtime":<Double_value>,

"responsecontentlength":<Integer_value>,

"responseyields":<Double_value>,
```

```
"responsemaxprocessingtime":<Double_value>

}]}
```

count

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/appfwtransactionrecords?**count=yes**

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "appfwtransactionrecords": [ { "__count": "#no"} ] }
```

appfwwsdl

Configuration for WSDL file resource.

Properties

(click to see Operations)

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
name	<String>	Read-write	Name of the WSDL file to remove.

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
			Minimum length = 1 Maximum length = 31
src	<String>	Read-write	URL (protocol, host, path, and name) of the WSDL file to be imported is stored. NOTE: The import fails if the object to be imported is on an HTTPS server that requires client certificate authentication for access. Minimum length = 1 Maximum length = 2047
comment	<String>	Read-write	Any comments to preserve information about the WSDL. Maximum length = 128
overwrite	<Boolean>	Read-write	Overwrite any existing WSDL of the same name.
response	<String>	Read-only	.

Operations

(click to see Properties)

DELETE| IMPORT| GET (ALL)| GET

Some options that you can use for each operations:

- **Getting warnings in response:**NITRO allows you to get warnings in an operation by specifying the "warning" query parameter as "yes". For example, to get warnings while connecting to the Citrix ADC appliance, the URL is as follows:

  http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/login?warning=yes

  If any, the warnings are displayed in the response payload with the HTTP code "209 X-NITRO-WARNING".

- **Authenticated access for individual NITRO operations:**NITRO allows you to logon to the Citrix ADC appliance to perform individual operations. You can use this option instead of creating a NITRO session (using the login object) and then using that session to perform all operations,

  To do this, you must specify the username and password in the request header of the NITRO request as follows:

  X-NITRO-USER:<username>

  X-NITRO-PASS:<password>

  Note: In such cases, make sure that the request header DOES not include the following:

  Cookie:NITRO_AUTH_TOKEN=<tokenvalue>

`*Note:`*

Mandatory parameters are marked in redand placeholder content is marked in <green>.

delete

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/appfwwsdl/name_value<String>

HTTP Method:DELETE

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue>

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error

Import

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/appfwwsdl?**action=Import**

HTTP Method:POST

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Content-Type:application/json

Request Payload:

```
{"appfwwsdl":{

<b>"src":<String_value>,

</b><b>"name":<String_value>,

</b>"comment":<String_value>,

"overwrite":<Boolean_value>

}}
```

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error

get (all)

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/appfwwsdl

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "appfwwsdl": [ {

"name":<String_value>,

"response":<String_value>

}]}
```

get

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/appfwwsdl/name_value<String>

Query-parameters:

attrs

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/appfwwsdl/name_value<String>?**attrs=property-name1,property-name2**

Use this query parameter to specify the resource details that you want to retrieve.

view

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/appfwwsdl/name_value<String>?**view=summary**

Note: By default, the retrieved results are displayed in detail view (?view=detail).

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "appfwwsdl": [ {

"name":<String_value>,

"response":<String_value>

}]}
```

appfwxmlcontenttype

Configuration for XML Content type resource.

Properties

(click to see Operations)

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
xmlcontenttypevalue	<String>	Read-write	Content type to be classified as XML. Minimum length = 1
isregex	<String>	Read-write	Is field name a regular expression?. Default value: NOTREGEX

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
			Possible values = REGEX, NOTREGEX
__count	<Double>	Read-only	count parameter

Operations

(click to see Properties)

ADD| DELETE| GET (ALL)| GET| COUNT

Some options that you can use for each operations:

- **Getting warnings in response:**NITRO allows you to get warnings in an operation by specifying the "warning" query parameter as "yes". For example, to get warnings while connecting to the Citrix ADC appliance, the URL is as follows:

  http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/login?warning=yes

  If any, the warnings are displayed in the response payload with the HTTP code "209 X-NITRO-WARNING".

- **Authenticated access for individual NITRO operations:**NITRO allows you to logon to the Citrix ADC appliance to perform individual operations. You can use this option instead of creating a NITRO session (using the login object) and then using that session to perform all operations,

  To do this, you must specify the username and password in the request header of the NITRO request as follows:

  X-NITRO-USER:<username>

  X-NITRO-PASS:<password>

  Note: In such cases, make sure that the request header DOES not include the following:

  Cookie:NITRO_AUTH_TOKEN=<tokenvalue>

`*Note:`*

Mandatory parameters are marked in redand placeholder content is marked in <green>.

add

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/appfwxmlcontenttype

HTTP Method:POST

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Content-Type:application/json

Request Payload:

```
{"appfwxmlcontenttype":{

<b>"xmlcontenttypevalue":<String_value>,

</b>"isregex":<String_value>

}}
```

Response:

HTTP Status Code on Success: 201 Created HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error

delete

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/appfwxmlcontenttype/xmlcontenttypevalue_value<String>

HTTP Method:DELETE

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue>

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error

get (all)

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/appfwxmlcontenttype

Query-parameters:

attrs

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/appfwxmlcontenttype?**attrs=property-name1,property-name2**

Use this query parameter to specify the resource details that you want to retrieve.

filter

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/appfwxmlcontenttype?**filter=property-name1:property-val1,property-name2:property-val2**

Use this query-parameter to get the filtered set of appfwxmlcontenttype resources configured on Citrix ADC. Filtering can be done on any of the properties of the resource.

view

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/appfwxmlcontenttype?**view=summary**

Note: By default, the retrieved results are displayed in detail view (?view=detail).

pagination

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/appfwxmlcontenttype?**pagesize=#no;pageno=#no**

Use this query-parameter to get the appfwxmlcontenttype resources in chunks.

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "appfwxmlcontenttype": [ {

"xmlcontenttypevalue":<String_value>,

"isregex":<String_value>

}]}
```

get

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/appfwxmlcontenttype/xmlcontenttypevalue_value<String>

Query-parameters:

attrs

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/appfwxmlcontenttype/xmlcontenttypevalue_value<String>?**attrs=property-name1,property-name2**

Use this query parameter to specify the resource details that you want to retrieve.

view

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/appfwxmlcontenttype/xmlcontenttypevalue_value<String>?**view=summary**

Note: By default, the retrieved results are displayed in detail view (?view=detail).

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "appfwxmlcontenttype": [ {

"xmlcontenttypevalue":<String_value>,

"isregex":<String_value>

}]}
```

count

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/appfwxmlcontenttype?**count=yes**

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "appfwxmlcontenttype": [ { "__count": "#no"} ] }
```

appfwxmlerrorpage

Configuration for xml error page resource.

Properties

(click to see Operations)

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
name	<String>	Read-write	Indicates name of the imported xml error page to be removed. Minimum length = 1 Maximum length = 31
src	<String>	Read-write	URL (protocol, host, path, and name) for the location at which to store the imported XML error object. NOTE: The import fails if the object to be imported is on an HTTPS server that requires client certificate authentication for access. Minimum length = 1 Maximum length = 2047
comment	<String>	Read-write	Any comments to preserve information

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
			about the XML error object. Maximum length = 128
overwrite	<Boolean>	Read-write	Overwrite any existing XML error object of the same name.
response	<String>	Read-only	.

Operations

(click to see Properties)

DELETE| IMPORT| CHANGE| GET (ALL)| GET

Some options that you can use for each operations:

- **Getting warnings in response:**NITRO allows you to get warnings in an operation by specifying the "warning" query parameter as "yes". For example, to get warnings while connecting to the Citrix ADC appliance, the URL is as follows:

  http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/login?warning=yes

  If any, the warnings are displayed in the response payload with the HTTP code "209 X-NITRO-WARNING".

- **Authenticated access for individual NITRO operations:**NITRO allows you to logon to the Citrix ADC appliance to perform individual operations. You can use this option instead of creating a NITRO session (using the login object) and then using that session to perform all operations,

  To do this, you must specify the username and password in the request header of the NITRO request as follows:

  X-NITRO-USER:<username>

  X-NITRO-PASS:<password>

  Note: In such cases, make sure that the request header DOES not include the following:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue>

`*Note:` *

Mandatory parameters are marked in redand placeholder content is marked in <green>.

delete

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/appfwxmlerrorpage/name_value<String>

HTTP Method:DELETE

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue>

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error

Import

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/appfwxmlerrorpage?**action=Import**

HTTP Method:POST

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Content-Type:application/json

Request Payload:

```
{"appfwxmlerrorpage":{

<b>"src":<String_value>,

</b><b>"name":<String_value>,

</b>"comment":<String_value>,

"overwrite":<Boolean_value>
```

```
}}
```

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error

change

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/appfwxmlerrorpage?**action=update**

HTTP Method:POST

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Content-Type:application/json

Request Payload:

```
{"appfwxmlerrorpage":{

<b>"name":<String_value>

</b>}}
```

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error

get (all)

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/appfwxmlerrorpage

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "appfwxmlerrorpage": [ {

"name":<String_value>,

"response":<String_value>,

"src":<String_value>

}]}
```

get

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/appfwxmlerrorpage/name_value<String>

Query-parameters:

attrs

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/appfwxmlerrorpage/name_value<String>?**attrs=property-name1,property-name2**

Use this query parameter to specify the resource details that you want to retrieve.

view

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/appfwxmlerrorpage/name_value<String>?**view=summary**

Note: By default, the retrieved results are displayed in detail view (?view=detail).

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "appfwxmlerrorpage": [ {

"name":<String_value>,

"response":<String_value>,

"src":<String_value>

}]}
```

appfwxmlschema

Configuration for XML schema resource.

Properties

(click to see Operations)

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
name	<String>	Read-write	Name of the XML Schema object to remove.

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
			Minimum length = 1 Maximum length = 31
src	<String>	Read-write	URL (protocol, host, path, and file name) for the location at which to store the imported XML Schema. NOTE: The import fails if the object to be imported is on an HTTPS server that requires client certificate authentication for access. Minimum length = 1 Maximum length = 2047
comment	<String>	Read-write	Any comments to preserve information about the XML Schema object. Maximum length = 128
overwrite	<Boolean>	Read-write	Overwrite any existing XML Schema object of the same name.
response	<String>	Read-only	.

Operations

(click to see Properties)

DELETE| IMPORT| GET (ALL)| GET

Some options that you can use for each operations:

- **Getting warnings in response:**NITRO allows you to get warnings in an operation by specifying the "warning" query parameter as "yes". For example, to get warnings while connecting to the Citrix ADC appliance, the URL is as follows:

  http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/login?warning=yes

  If any, the warnings are displayed in the response payload with the HTTP code "209 X-NITRO-WARNING".

- **Authenticated access for individual NITRO operations:**NITRO allows you to logon to the Citrix ADC appliance to perform individual operations. You can use this option instead of creating a NITRO session (using the login object) and then using that session to perform all operations,

  To do this, you must specify the username and password in the request header of the NITRO request as follows:

  X-NITRO-USER:<username>

  X-NITRO-PASS:<password>

  Note: In such cases, make sure that the request header DOES not include the following:

  Cookie:NITRO_AUTH_TOKEN=<tokenvalue>

`*Note:`*

Mandatory parameters are marked in redand placeholder content is marked in <green>.

delete

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/appfwxmlschema/name_value<String>

HTTP Method:DELETE

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue>

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error

Import

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/appfwxmlschema?**action=Import**

HTTP Method:POST

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Content-Type:application/json

Request Payload:

```
{"appfwxmlschema":{

<b>"src":<String_value>,

</b><b>"name":<String_value>,

</b>"comment":<String_value>,

"overwrite":<Boolean_value>

}}
```

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error

get (all)

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/appfwxmlschema

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "appfwxmlschema": [ {

"name":<String_value>,

"response":<String_value>

}]}
```

get

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/appfwxmlschema/name_value<String>

Query-parameters:

attrs

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/appfwxmlschema/name_value<String>?**attrs=property-name1,property-name2**

Use this query parameter to specify the resource details that you want to retrieve.

view

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/appfwxmlschema/name_value<String>?**view=summary**

Note: By default, the retrieved results are displayed in detail view (?view=detail).

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "appfwxmlschema": [ {

"name":<String_value>,

"response":<String_value>

}]}
```

Appqoe

Appqoe

Application Level Quality of Experience configuration

NAME	DESCRIPTION
appqoeaction	Configuration for AppQoS action
appqoecustomresp	Configuration for AppQoE custom response page
appqoeparameter	Configuration for QOS parameter

NAME	DESCRIPTION
appqoepolicy	Configuration for AppQoS policy
appqoepolicy_binding	Binding object showing the resources that can be bound to appqoepolicy
appqoepolicy_lbvserver_binding	Binding object showing the lbvserver that can be bound to appqoepolicy

appqoeaction

Configuration for AppQoS action resource.

Properties

(click to see Operations)

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
name	<String>	Read-write	Name for the AppQoE action. Must begin with a letter, number, or the underscore symbol (_). Other characters allowed, after the first character, are the hyphen (-), period (.) hash (#), space ( ), at (@), equals (=), and colon (:) characters. This is a mandatory argument.
priority	<String>	Read-write	Priority for queuing the request. If server resources are not available for a request that matches the configured rule, this option specifies

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
			a priority for queuing the request until the server resources are available again. If priority is not configured then Lowest priority will be used to queue the request. Possible values = HIGH, MEDIUM, LOW, LOWEST
respondwith	<String>	Read-write	Responder action to be taken when the threshold is reached. Available settings function as follows: ACS - Serve content from an alternative content service Threshold : maxConn or delay NS - Serve from the Citrix ADC (built-in response) Threshold : maxConn or delay. Possible values = ACS, NS
customfile	<String>	Read-write	name of the HTML page object to use as the response. Minimum length = 1

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
altcontentsvcname	<String>	Read-write	Name of the alternative content service to be used in the ACS. Minimum length = 1 Maximum length = 127
altcontentpath	<String>	Read-write	Path to the alternative content service to be used in the ACS. Minimum length = 4 Maximum length = 127
polqdepth	<Double>	Read-write	Policy queue depth threshold value. When the policy queue size (number of requests queued for the policy binding this action is attached to) increases to the specified polqDepth value, subsequent requests are dropped to the lowest priority level. Minimum value = 0 Maximum value = 4294967294
priqdepth	<Double>	Read-write	Queue depth threshold value per priorirty level. If the queue size (number

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
			of requests in the queue of that particular priorirty) on the virtual server to which this policy is bound, increases to the specified qDepth value, subsequent requests are dropped to the lowest priority level. Minimum value = 0 Maximum value = 4294967294
maxconn	<Double>	Read-write	Maximum number of concurrent connections that can be open for requests that matches with rule. Minimum value = 1 Maximum value = 4294967294
delay	<Double>	Read-write	Delay threshold, in microseconds, for requests that match the policy's rule. If the delay statistics gathered for the matching request exceed the specified delay, configured action triggered for that request, if there is no action then requests are

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
			dropped to the lowest priority level. Minimum value = 1 Maximum value = 599999999
dostrigexpression	<String>	Read-write	Optional expression to add second level check to trigger DoS actions. Specifically used for Analytics based DoS response generation.
dosaction	<String>	Read-write	DoS Action to take when vserver will be considered under DoS attack and corresponding rule matches. Mandatory if AppQoE actions are to be used for DoS attack prevention. Possible values = SimpleResponse, HICResponse
tcpprofile	<String>	Read-write	Bind TCP Profile based on L2/L3/L7 parameters. Minimum length = 1 Maximum length = 127
hits	<Double>	Read-only	.
__count	<Double>	Read-only	count parameter

Operations

(click to see Properties)

ADD| DELETE| UPDATE| UNSET| GET (ALL)| GET| COUNT

Some options that you can use for each operations:

- **Getting warnings in response:**NITRO allows you to get warnings in an operation by specifying the "warning" query parameter as "yes". For example, to get warnings while connecting to the Citrix ADC appliance, the URL is as follows:

  http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/login?warning=yes

  If any, the warnings are displayed in the response payload with the HTTP code "209 X-NITRO-WARNING".

- **Authenticated access for individual NITRO operations:**NITRO allows you to logon to the Citrix ADC appliance to perform individual operations. You can use this option instead of creating a NITRO session (using the login object) and then using that session to perform all operations,

  To do this, you must specify the username and password in the request header of the NITRO request as follows:

  X-NITRO-USER:<username>

  X-NITRO-PASS:<password>

  Note: In such cases, make sure that the request header DOES not include the following:

  Cookie:NITRO_AUTH_TOKEN=<tokenvalue>

`*Note:`*

Mandatory parameters are marked in redand placeholder content is marked in <green>.

add

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/appqoeaction

HTTP Method:POST

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Content-Type:application/json

Request Payload:

```
{"appqoeaction":{

<b>"name":<String_value>,

</b>"priority":<String_value>,

"respondwith":<String_value>,

"customfile":<String_value>,

"altcontentsvcname":<String_value>,

"altcontentpath":<String_value>,

"polqdepth":<Double_value>,

"priqdepth":<Double_value>,

"maxconn":<Double_value>,

"delay":<Double_value>,

"dostrigexpression":<String_value>,

"dosaction":<String_value>,

"tcpprofile":<String_value>

}}
```

Response:

HTTP Status Code on Success: 201 Created HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error

delete

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/appqoeaction/name_value<String>

HTTP Method:DELETE

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue>

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error

update

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/appqoeaction

HTTP Method:PUT

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Content-Type:application/json

Request Payload:

```
{"appqoeaction":{

<b>"name":<String_value>,

</b>"priority":<String_value>,

"altcontentsvcname":<String_value>,

"altcontentpath":<String_value>,

"polqdepth":<Double_value>,

"priqdepth":<Double_value>,

"maxconn":<Double_value>,
```

```
"delay":<Double_value>,

"dostrigexpression":<String_value>,

"dosaction":<String_value>,

"tcpprofile":<String_value>

}}
```

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error

unset

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/appqoeaction?**action=unset**

HTTP Method:POST

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Content-Type:application/json

Request Payload:

```
{"appqoeaction":{

<b>"name":<String_value>,

</b>"priority":true,

"altcontentsvcname":true,

"altcontentpath":true,

"polqdepth":true,
```

```
"priqdepth":true,

"maxconn":true,

"delay":true,

"dosaction":true,

"tcpprofile":true

}}
```

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error

get (all)

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/appqoeaction

Query-parameters:

attrs

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/appqoeaction?**attrs=property-name1,property-name2**

Use this query parameter to specify the resource details that you want to retrieve.

filter

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/appqoeaction?**filter=property-name1:property-val1,property-name2:property-val2**

Use this query-parameter to get the filtered set of appqoeaction resources configured on Citrix ADC. Filtering can be done on any of the properties of the resource.

view

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/appqoeaction?**view=summary**

Note: By default, the retrieved results are displayed in detail view (?view=detail).

pagination

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/appqoeaction?**pagesize=#no;pageno=#no**

Use this query-parameter to get the appqoeaction resources in chunks.

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "appqoeaction": [ {

"name":<String_value>,

"hits":<Double_value>,

"priority":<String_value>,

"respondwith":<String_value>,

"polqdepth":<Double_value>,

"priqdepth":<Double_value>,

"altcontentsvcname":<String_value>,

"altcontentpath":<String_value>,
```

```
"maxconn":<Double_value>,

"delay":<Double_value>,

"customfile":<String_value>,

"dostrigexpression":<String_value>,

"dosaction":<String_value>,

"tcpprofile":<String_value>

}]}
```

get

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/appqoeaction/name_value<String>

Query-parameters:

attrs

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/appqoeaction/name_value<String>?**attrs=property-name1,property-name2**

Use this query parameter to specify the resource details that you want to retrieve.

view

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/appqoeaction/name_value<String>?**view=summary**

Note: By default, the retrieved results are displayed in detail view (?view=detail).

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "appqoeaction": [ {

"name":<String_value>,

"hits":<Double_value>,

"priority":<String_value>,

"respondwith":<String_value>,

"polqdepth":<Double_value>,

"priqdepth":<Double_value>,

"altcontentsvcname":<String_value>,

"altcontentpath":<String_value>,

"maxconn":<Double_value>,

"delay":<Double_value>,

"customfile":<String_value>,

"dostrigexpression":<String_value>,

"dosaction":<String_value>,

"tcpprofile":<String_value>

}]}
```

count

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/appqoeaction?**count=yes**

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "appqoeaction": [ { "__count": "#no"} ] }
```

appqoecustomresp

Configuration for AppQoE custom response page resource.

Properties

(click to see Operations)

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
src	<String>	Read-write	. Minimum length = 1 Maximum length = 2047
name	<String>	Read-write	Indicates name of the custom response

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
			HTML page to import/update. Minimum length = 1 Maximum length = 31
__count	<Double>	Read-only	count parameter

Operations

(click to see Properties)

IMPORT| DELETE| CHANGE| GET (ALL)| COUNT

Some options that you can use for each operations:

- **Getting warnings in response:**NITRO allows you to get warnings in an operation by specifying the "warning" query parameter as "yes". For example, to get warnings while connecting to the Citrix ADC appliance, the URL is as follows:

  http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/login?warning=yes

  If any, the warnings are displayed in the response payload with the HTTP code "209 X-NITRO-WARNING".

- **Authenticated access for individual NITRO operations:**NITRO allows you to logon to the Citrix ADC appliance to perform individual operations. You can use this option instead of creating a NITRO session (using the login object) and then using that session to perform all operations,

  To do this, you must specify the username and password in the request header of the NITRO request as follows:

  X-NITRO-USER:<username>

  X-NITRO-PASS:<password>

  Note: In such cases, make sure that the request header DOES not include the following:

  Cookie:NITRO_AUTH_TOKEN=<tokenvalue>

`*Note:`*

Mandatory parameters are marked in redand placeholder content is marked in <green>.

Import

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/appqoecustomresp?**action=Import**

HTTP Method:POST

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Content-Type:application/json

Request Payload:

```
{"appqoecustomresp":{

"src":<String_value>,

<b>"name":<String_value>

</b>}}
```

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error

delete

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/appqoecustomresp/name_value<String>

HTTP Method:DELETE

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue>

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error

change

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/appqoecustomresp?**action=update**

HTTP Method:POST

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Content-Type:application/json

Request Payload:

```
{"appqoecustomresp":{

<b>"name":<String_value>

</b>}}
```

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error

get (all)

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/appqoecustomresp

Query-parameters:

attrs

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/appqoecustomresp?**attrs=property-name1,property-name2**

Use this query parameter to specify the resource details that you want to retrieve.

filter

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/appqoecustomresp?**filter=property-name1:property-val1,property-name2:property-val2**

Use this query-parameter to get the filtered set of appqoecustomresp resources configured on Citrix ADC. Filtering can be done on any of the properties of the resource.

view

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/appqoecustomresp?**view=summary**

Note: By default, the retrieved results are displayed in detail view (?view=detail).

pagination

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/appqoecustomresp?**pagesize=#no;pageno=#no**

Use this query-parameter to get the appqoecustomresp resources in chunks.

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "appqoecustomresp": [ {

"name":<String_value>,

"src":<String_value>

}]}
```

count

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/appqoecustomresp?**count=yes**

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "appqoecustomresp": [ { "__count": "#no"} ] }
```

appqoeparameter

Configuration for QOS parameter resource.

Properties

(click to see Operations)

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
sessionlife	<Double>	Read-write	Time, in seconds, between the first time and the next time the AppQoE alternative content window is displayed. The alternative

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
			content window is displayed only once during a session for the same browser accessing a configured URL, so this parameter determines the length of a session. Default value: 300 Minimum value = 1 Maximum value = 4294967294
avgwaitingclient	<Double>	Read-write	average number of client connections, that can sit in service waiting queue. Default value: 1000000 Minimum value = 0 Maximum value = 4294967294
maxaltrespbandwidth	<Double>	Read-write	maximum bandwidth which will determine whether to send alternate content response. Default value: 100 Minimum value = 1 Maximum value = 4294967294

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
dosattackthresh	<Double>	Read-write	average number of client connection that can queue up on vserver level without triggering DoS mitigation module. Default value: 2000 Minimum value = 0 Maximum value = 4294967294

Operations

(click to see Properties)

UPDATE| UNSET| GET (ALL)

Some options that you can use for each operations:

- **Getting warnings in response:**NITRO allows you to get warnings in an operation by specifying the "warning" query parameter as "yes". For example, to get warnings while connecting to the Citrix ADC appliance, the URL is as follows:

  http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/login?warning=yes

  If any, the warnings are displayed in the response payload with the HTTP code "209 X-NITRO-WARNING".

- **Authenticated access for individual NITRO operations:**NITRO allows you to logon to the Citrix ADC appliance to perform individual operations. You can use this option instead of creating a NITRO session (using the login object) and then using that session to perform all operations,

  To do this, you must specify the username and password in the request header of the NITRO request as follows:

  X-NITRO-USER:<username>

  X-NITRO-PASS:<password>

Note: In such cases, make sure that the request header DOES not include the following:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue>

`*Note:` *

Mandatory parameters are marked in redand placeholder content is marked in <green>.

update

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/appqoeparameter

HTTP Method:PUT

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Content-Type:application/json

Request Payload:

```
{"appqoeparameter":{

"sessionlife":<Double_value>,

"avgwaitingclient":<Double_value>,

"maxaltrespbandwidth":<Double_value>,

"dosattackthresh":<Double_value>

}}
```

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error

unset

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/appqoeparameter?**action=unset**

HTTP Method:POST

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Content-Type:application/json

Request Payload:

```
{"appqoeparameter":{

"sessionlife":true,

"avgwaitingclient":true,

"maxaltrespbandwidth":true,

"dosattackthresh":true

}}
```

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error

get (all)

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/appqoeparameter

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "appqoeparameter": [ {

"sessionlife":<Double_value>,

"avgwaitingclient":<Double_value>,

"maxaltrespbandwidth":<Double_value>,

"dosattackthresh":<Double_value>

}]}
```

appqoepolicy

Configuration for AppQoS policy resource.

Properties

(click to see Operations)

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
name	<String>	Read-write	. Minimum length = 1
rule	<String>	Read-write	Expression or name of a named expression, against which the request is evaluated. The policy is applied if the rule evaluates to true.
action	<String>	Read-write	Configured AppQoE action to trigger. Minimum length = 1

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
hits	<Double>	Read-only	Number of hits.
__count	<Double>	Read-only	count parameter

Operations

(click to see Properties)

ADD| DELETE| UPDATE| GET (ALL)| GET| COUNT

Some options that you can use for each operations:

- **Getting warnings in response:**NITRO allows you to get warnings in an operation by specifying the "warning" query parameter as "yes". For example, to get warnings while connecting to the Citrix ADC appliance, the URL is as follows:

  http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/login?warning=yes

  If any, the warnings are displayed in the response payload with the HTTP code "209 X-NITRO-WARNING".

- **Authenticated access for individual NITRO operations:**NITRO allows you to logon to the Citrix ADC appliance to perform individual operations. You can use this option instead of creating a NITRO session (using the login object) and then using that session to perform all operations,

  To do this, you must specify the username and password in the request header of the NITRO request as follows:

  X-NITRO-USER:<username>

  X-NITRO-PASS:<password>

  Note: In such cases, make sure that the request header DOES not include the following:

  Cookie:NITRO_AUTH_TOKEN=<tokenvalue>

`*Note:`*

Mandatory parameters are marked in redand placeholder content is marked in <green>.

add

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/appqoepolicy

HTTP Method:POST

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Content-Type:application/json

Request Payload:

```
{"appqoepolicy":{

<b>"name":<String_value>,

</b><b>"rule":<String_value>,

</b><b>"action":<String_value>

</b>}}
```

Response:

HTTP Status Code on Success: 201 Created HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error

delete

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/appqoepolicy/name_value<String>

HTTP Method:DELETE

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue>

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error

update

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/appqoepolicy

HTTP Method:PUT

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Content-Type:application/json

Request Payload:

```
{"appqoepolicy":{

<b>"name":<String_value>,

</b>"rule":<String_value>,

"action":<String_value>

}}
```

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error

get (all)

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/appqoepolicy

Query-parameters:

attrs

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/appqoepolicy?**attrs=property-name1,property-name2**

Use this query parameter to specify the resource details that you want to retrieve.

filter

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/appqoepolicy?**filter=property-name1:property-val1,property-name2:property-val2**

Use this query-parameter to get the filtered set of appqoepolicy resources configured on Citrix ADC. Filtering can be done on any of the properties of the resource.

view

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/appqoepolicy?**view=summary**

Note: By default, the retrieved results are displayed in detail view (?view=detail).

pagination

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/appqoepolicy?**pagesize=#no;pageno=#no**

Use this query-parameter to get the appqoepolicy resources in chunks.

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "appqoepolicy": [ {

"name":<String_value>,

"rule":<String_value>,

"action":<String_value>,

"hits":<Double_value>
```

```
}]}
```

get

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/appqoepolicy/name_value<String>

Query-parameters:

attrs

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/appqoepolicy/name_value<String>?**attrs=property-name1,property-name2**

Use this query parameter to specify the resource details that you want to retrieve.

view

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/appqoepolicy/name_value<String>?**view=summary**

Note: By default, the retrieved results are displayed in detail view (?view=detail).

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "appqoepolicy": [ {

"name":<String_value>,
```

```
"rule":<String_value>,

"action":<String_value>,

"hits":<Double_value>

}]}
```

count

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/appqoepolicy?**count=yes**

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "appqoepolicy": [ { "__count": "#no"} ] }
```

appqoepolicy_binding

Binding object which returns the resources bound to appqoepolicy.

Properties

(click to see Operations)

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
name	<String>	Read-write	. Minimum length = 1
appqoepolicy_lbvserver_binding	<appqoepolicy_lbvserver_binding[]>	Read-only	lbvserver that can be bound to appqoepolicy.

Operations

(click to see Properties)

GET| GET (ALL)

Some options that you can use for each operations:

- **Getting warnings in response:**NITRO allows you to get warnings in an operation by specifying the "warning" query parameter as "yes". For example, to get warnings while connecting to the Citrix ADC appliance, the URL is as follows:

  http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/login?warning=yes

  If any, the warnings are displayed in the response payload with the HTTP code "209 X-NITRO-WARNING".

- **Authenticated access for individual NITRO operations:**NITRO allows you to logon to the Citrix ADC appliance to perform individual operations. You can use this option instead of creating a NITRO session (using the login object) and then using that session to perform all operations,

  To do this, you must specify the username and password in the request header of the NITRO request as follows:

  X-NITRO-USER:<username>

  X-NITRO-PASS:<password>

  Note: In such cases, make sure that the request header DOES not include the following:

  Cookie:NITRO_AUTH_TOKEN=<tokenvalue>

`*Note:`*

Mandatory parameters are marked in redand placeholder content is marked in <green>.

get

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/appqoepolicy_binding/name_value<String>

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "appqoepolicy_binding": [ {

"name":<String_value>,

"appqoepolicy_lbvserver_binding":<appqoepolicy_lbvserver_binding[]_value>

}]}
```

get (all)

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/appqoepolicy_binding

Query-parameters:

bulkbindings

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/appqoepolicy_binding?**bulkbindings=yes**

NITRO allows you to fetch bindings in bulk.

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "appqoepolicy_binding": [ {

"name":<String_value>,

"appqoepolicy_lbvserver_binding":<appqoepolicy_lbvserver_binding[]_value>

}]}
```

appqoepolicy_lbvserver_binding

Binding object showing the lbvserver that can be bound to appqoepolicy.

Properties

(click to see Operations)

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
name	<String>	Read-write	. Minimum length = 1
boundto	<String>	Read-write	The name of the entity to which the policy is bound.

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
bindpriority	<Double>	Read-only	Specifies the binding of the policy. use only in display.
gotopriorityexpression	<String>	Read-only	Expression specifying the priority of the next policy which will get evaluated if the current policy rule evaluates to TRUE.
activepolicy	<Double>	Read-only	.
__count	<Double>	Read-only	count parameter

Operations

(click to see Properties)

GET| GET (ALL)| COUNT

Some options that you can use for each operations:

- **Getting warnings in response:**NITRO allows you to get warnings in an operation by specifying the "warning" query parameter as "yes". For example, to get warnings while connecting to the Citrix ADC appliance, the URL is as follows:

  http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/login?warning=yes

  If any, the warnings are displayed in the response payload with the HTTP code "209 X-NITRO-WARNING".

- **Authenticated access for individual NITRO operations:**NITRO allows you to logon to the Citrix ADC appliance to perform individual operations. You can use this option instead of creating a NITRO session (using the login object) and then using that session to perform all operations,

  To do this, you must specify the username and password in the request header of the NITRO request as follows:

  X-NITRO-USER:<username>

X-NITRO-PASS:<password>

Note: In such cases, make sure that the request header DOES not include the following:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue>

`*Note:`*

Mandatory parameters are marked in redand placeholder content is marked in <green>.

get

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/appqoepolicy_lbvserver_binding/name_value<String>

Query-parameters:

filter

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/appqoepolicy_lbvserver_binding/name_value<String>?**filter=property-name1:property-value1,property-name2:property-value2**

Use this query-parameter to get the filtered set of appqoepolicy_lbvserver_binding resources configured on Citrix ADC. Filtering can be done on any of the properties of the resource.

pagination

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/appqoepolicy_lbvserver_binding/name_value<String>?**pagesize=#no;pageno=#no**

Use this query-parameter to get the appqoepolicy_lbvserver_binding resources in chunks.

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "appqoepolicy_lbvserver_binding": [ {

"name":<String_value>,

"boundto":<String_value>,

"bindpriority":<Double_value>,

"gotopriorityexpression":<String_value>,

"activepolicy":<Double_value>

}]}
```

get (all)

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/appqoepolicy_lbvserver_binding

Query-parameters:

bulkbindings

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/appqoepolicy_lbvserver_binding?**bulkbindings=yes**

NITRO allows you to fetch bindings in bulk.

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "appqoepolicy_lbvserver_binding": [ {

"name":<String_value>,

"boundto":<String_value>,

"bindpriority":<Double_value>,

"gotopriorityexpression":<String_value>,

"activepolicy":<Double_value>

}]}
```

count

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/appqoepolicy_lbvserver_binding/name_value<String>?**count=yes**

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{"appqoepolicy_lbvserver_binding": [ { "__count": "#no"} ] }
```


Audit

Audit

Audit configuration.

NAME	DESCRIPTION
auditmessageaction	Configuration for message action
auditmessages	Configuration for audit message
auditnslogaction	Configuration for ns log action
auditnslogglobal_auditnslogpolicy_binding	Binding object showing the auditnslogpolicy that can be bound to auditnslogglobal
auditnslogglobal_binding	Binding object showing the resources that can be bound to auditnslogglobal
auditnslogparams	Configuration for ns log parameters
auditnslogpolicy	Configuration for ns log policy
auditnslogpolicy_aaagroup_binding	Binding object showing the aaagroup that can be bound to auditnslogpolicy
auditnslogpolicy_aaauser_binding	Binding object showing the aaauser that can be bound to auditnslogpolicy
auditnslogpolicy_appfwglobal_binding	Binding object showing the appfwglobal that can be bound to auditnslogpolicy
auditnslogpolicy_auditnslogglobal_binding	Binding object showing the auditnslogglobal that can be bound to auditnslogpolicy

NAME	DESCRIPTION
auditnslogpolicy_authenticationvserver_binding	Binding object showing the authenticationvserver that can be bound to auditnslogpolicy
auditnslogpolicy_binding	Binding object showing the resources that can be bound to auditnslogpolicy
auditnslogpolicy_csvserver_binding	Binding object showing the csvserver that can be bound to auditnslogpolicy
auditnslogpolicy_lbvserver_binding	Binding object showing the lbvserver that can be bound to auditnslogpolicy
auditnslogpolicy_systemglobal_binding	Binding object showing the systemglobal that can be bound to auditnslogpolicy
auditnslogpolicy_tmglobal_binding	Binding object showing the tmglobal that can be bound to auditnslogpolicy
auditnslogpolicy_vpnglobal_binding	Binding object showing the vpnglobal that can be bound to auditnslogpolicy
auditnslogpolicy_vpnvserver_binding	Binding object showing the vpnvserver that can be bound to auditnslogpolicy
auditsyslogaction	Configuration for system log action
auditsyslogglobal_auditsyslogpolicy_binding	Binding object showing the auditsyslogpolicy that can be bound to auditsyslogglobal
auditsyslogglobal_binding	Binding object showing the resources that can be bound to auditsyslogglobal
auditsyslogparams	Configuration for system log parameters
auditsyslogpolicy	Configuration for system log policy
auditsyslogpolicy_aaagroup_binding	Binding object showing the aaagroup that can be bound to auditsyslogpolicy
auditsyslogpolicy_aaauser_binding	Binding object showing the aaauser that can be bound to auditsyslogpolicy

NAME	DESCRIPTION
auditsyslogpolicy_auditsysloglobal_binding	Binding object showing the auditsysloglobal that can be bound to auditsyslogpolicy
auditsyslogpolicy_authenticationvserver_binding	Binding object showing the authenticationvserver that can be bound to auditsyslogpolicy
auditsyslogpolicy_binding	Binding object showing the resources that can be bound to auditsyslogpolicy
auditsyslogpolicy_csvserver_binding	Binding object showing the csvserver that can be bound to auditsyslogpolicy
auditsyslogpolicy_lbvserver_binding	Binding object showing the lbvserver that can be bound to auditsyslogpolicy
auditsyslogpolicy_rnatglobal_binding	Binding object showing the rnatglobal that can be bound to auditsyslogpolicy
auditsyslogpolicy_systemglobal_binding	Binding object showing the systemglobal that can be bound to auditsyslogpolicy
auditsyslogpolicy_tmglobal_binding	Binding object showing the tmglobal that can be bound to auditsyslogpolicy
auditsyslogpolicy_vpnglobal_binding	Binding object showing the vpnglobal that can be bound to auditsyslogpolicy
auditsyslogpolicy_vpnvserver_binding	Binding object showing the vpnvserver that can be bound to auditsyslogpolicy

auditmessageaction

Configuration for message action resource.

Properties

(click to see Operations)

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
name	<String>	Read-write	Name of the audit message action. Must begin with a letter, number, or the underscore character (_), and must contain only letters, numbers, and the hyphen (-), period (.) pound (#), space ( ), at (@), equals (=), colon (:), and underscore characters. Cannot be changed after the message action is added. The following requirement applies only to the Citrix ADC CLI: If the name includes one or more spaces, enclose the name in double or single quotation marks (for example, "my message action" or 'my message action'). Minimum length = 1
loglevel	<String>	Read-write	Audit log level, which specifies the severity level of the log message being generated..

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
			The following loglevels are valid: * EMERGENCY - Events that indicate an immediate crisis on the server. * ALERT - Events that might require action. * CRITICAL - Events that indicate an imminent server crisis. * ERROR - Events that indicate some type of error. * WARNING - Events that require action in the near future. * NOTICE - Events that the administrator should know about. * INFORMATIONAL - All but low-level events. * DEBUG - All events, in extreme detail.

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
			Possible values = EMERGENCY, ALERT, CRITICAL, ERROR, WARNING, NOTICE, INFORMATIONAL, DEBUG
stringbuilderexpr	<String>	Read-write	Default-syntax expression that defines the format and content of the log message.
logtonewnslog	<String>	Read-write	Send the message to the new nslog. Possible values = YES, NO
bypasssafetycheck	<String>	Read-write	Bypass the safety check and allow unsafe expressions. Default value: NO Possible values = YES, NO
loglevel1	<String>	Read-only	. Possible values = ALL, EMERGENCY, ALERT, CRITICAL, ERROR, WARNING, NOTICE, INFORMATIONAL, DEBUG, NONE

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
hits	<Double>	Read-only	The number of times the action has been taken.
undefhits	<Double>	Read-only	The number of times the action resulted in UNDEF.
referencecount	<Double>	Read-only	The number of references to the action.
__count	<Double>	Read-only	count parameter

Operations

(click to see Properties)

ADD| DELETE| UPDATE| UNSET| GET (ALL)| GET| COUNT

Some options that you can use for each operations:

- **Getting warnings in response:**NITRO allows you to get warnings in an operation by specifying the "warning" query parameter as "yes". For example, to get warnings while connecting to the Citrix ADC appliance, the URL is as follows:

  http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/login?warning=yes

  If any, the warnings are displayed in the response payload with the HTTP code "209 X-NITRO-WARNING".

- **Authenticated access for individual NITRO operations:**NITRO allows you to logon to the Citrix ADC appliance to perform individual operations. You can use this option instead of creating a NITRO session (using the login object) and then using that session to perform all operations,

  To do this, you must specify the username and password in the request header of the NITRO request as follows:

  X-NITRO-USER:<username>

  X-NITRO-PASS:<password>

Note: In such cases, make sure that the request header DOES not include the following:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue>

`*Note:` *

Mandatory parameters are marked in redand placeholder content is marked in <green>.

add

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/auditmessageaction

HTTP Method:POST

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Content-Type:application/json

Request Payload:

```
{"auditmessageaction":{

<b>"name":<String_value>,

</b><b>"loglevel":<String_value>,

</b><b>"stringbuilderexpr":<String_value>,

</b>"logtonewnslog":<String_value>,

"bypasssafetycheck":<String_value>

}}
```

Response:

HTTP Status Code on Success: 201 Created HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error

delete

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/auditmessageaction/name_value<String>

HTTP Method:DELETE

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue>

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error

update

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/auditmessageaction

HTTP Method:PUT

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Content-Type:application/json

Request Payload:

```
{"auditmessageaction":{

<b>"name":<String_value>,

</b>"loglevel":<String_value>,

"stringbuilderexpr":<String_value>,

"logtonewnslog":<String_value>,

"bypasssafetycheck":<String_value>

}}
```

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error

unset

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/auditmessageaction?**action=unset**

HTTP Method:POST

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Content-Type:application/json

Request Payload:

```
{"auditmessageaction":{

<b>"name":<String_value>,

</b>"logtonewnslog":true,

"bypasssafetycheck":true

}}
```

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error

get (all)

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/auditmessageaction

Query-parameters:

attrs

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/auditmessageaction?**attrs=property-name1,property-name2**

Use this query parameter to specify the resource details that you want to retrieve.

filter

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/auditmessageaction?**filter=property-name1:property-val1,property-name2:property-val2**

Use this query-parameter to get the filtered set of auditmessageaction resources configured on Citrix ADC. Filtering can be done on any of the properties of the resource.

view

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/auditmessageaction?**view=summary**

Note: By default, the retrieved results are displayed in detail view (?view=detail).

pagination

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/auditmessageaction?**pagesize=#no;pageno=#no**

Use this query-parameter to get the auditmessageaction resources in chunks.

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "auditmessageaction": [ {

"name":<String_value>,

"loglevel":<String_value>,

"loglevel1":<String_value>,

"stringbuilderexpr":<String_value>,

"logtonewnslog":<String_value>,

"bypasssafetycheck":<String_value>,

"hits":<Double_value>,

"undefhits":<Double_value>,

"referencecount":<Double_value>

}]}
```

get

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/auditmessageaction/name_value<String>

Query-parameters:

attrs

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/auditmessageaction/name_value<String>?**attrs=property-name1,property-name2**

Use this query parameter to specify the resource details that you want to retrieve.

view

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/auditmessageaction/name_value<String>?**view=summary**

Note: By default, the retrieved results are displayed in detail view (?view=detail).

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "auditmessageaction": [ {

"name":<String_value>,

"loglevel":<String_value>,

"loglevel1":<String_value>,

"stringbuilderexpr":<String_value>,

"logtonewnslog":<String_value>,

"bypasssafetycheck":<String_value>,

"hits":<Double_value>,

"undefhits":<Double_value>,

"referencecount":<Double_value>

}]}
```

count

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/auditmessageaction?**count=yes**

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "auditmessageaction": [ { "__count": "#no"} ] }
```

auditmessages

Configuration for audit message resource.

Properties

(click to see Operations)

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
loglevel	<String[]>	Read-write	Audit log level filter, which specifies the types of events to display. The following loglevels are valid: * ALL - All events.

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
			* EMERGENCY - Events that indicate an immediate crisis on the server. * ALERT - Events that might require action. * CRITICAL - Events that indicate an imminent server crisis. * ERROR - Events that indicate some type of error. * WARNING - Events that require action in the near future. * NOTICE - Events that the administrator should know about. * INFORMATIONAL - All but low-level events. * DEBUG - All events, in extreme detail. Possible values = ALL, EMERGENCY, ALERT, CRITICAL,

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
			ERROR, WARNING, NOTICE, INFORMATIONAL, DEBUG
numofmesgs	<Double>	Read-write	Number of log messages to be displayed. Default value: 20 Minimum value = 1 Maximum value = 256
value	<String>	Read-only	The Audit message.
__count	<Double>	Read-only	count parameter

Operations

(click to see Properties)

GET (ALL)| COUNT

Some options that you can use for each operations:

- **Getting warnings in response:**NITRO allows you to get warnings in an operation by specifying the "warning" query parameter as "yes". For example, to get warnings while connecting to the Citrix ADC appliance, the URL is as follows:

  http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/login?warning=yes

  If any, the warnings are displayed in the response payload with the HTTP code "209 X-NITRO-WARNING".

- **Authenticated access for individual NITRO operations:**NITRO allows you to logon to the Citrix ADC appliance to perform individual operations. You can use this option instead of creating a NITRO session (using the login object) and then using that session to perform all operations,

To do this, you must specify the username and password in the request header of the NITRO request as follows:

X-NITRO-USER:<username>

X-NITRO-PASS:<password>

Note: In such cases, make sure that the request header DOES not include the following:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue>

`*Note:`*

Mandatory parameters are marked in redand placeholder content is marked in <green>.

get (all)

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/auditmessages

Query-parameters:

args

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/auditmessages?**args=loglevel:<String_value>,numofmesgs:<Double_value>**

Use this query-parameter to get auditmessages resources based on additional properties.

attrs

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/auditmessages?**attrs=property-name1,property-name2**

Use this query parameter to specify the resource details that you want to retrieve.

filter

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/auditmessages?**filter=property-name1:property-val1,property-name2:property-val2**

Use this query-parameter to get the filtered set of auditmessages resources configured on Citrix ADC. Filtering can be done on any of the properties of the resource.

view

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/auditmessages?**view=summary**

Note: By default, the retrieved results are displayed in detail view (?view=detail).

pagination

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/auditmessages?**pagesize=#no;pageno=#no**

Use this query-parameter to get the auditmessages resources in chunks.

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "auditmessages": [ {

loglevel:<String[]_value>,numofmesgs:<Double_value>"value":<String_value>

}]}
```

count

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/auditmessages?**count=yes**

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "auditmessages": [ { "__count": "#no"} ] }
```

auditnslogaction

Configuration for ns log action resource.

Properties

(click to see Operations)

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
name	<String>	Read-write	Name of the nslog action. Must begin with a letter, number, or the underscore character (_), and must contain only letters, numbers, and the hyphen (-), period (.) pound (#), space ( ), at (@), equals (=), colon (:), and underscore characters. Cannot be changed after the nslog action is added. The following requirement applies

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
			only to the Citrix ADC CLI: If the name includes one or more spaces, enclose the name in double or single quotation marks (for example, "my nslog action" or 'my nslog action'). Minimum length = 1
serverip	<String>	Read-write	IP address of the nslog server. Minimum length = 1
serverdomainname	<String>	Read-write	Auditserver name as a FQDN. Mutually exclusive with serverIP. Minimum length = 1 Maximum length = 255
domainresolveretry	<Integer>	Read-write	Time, in seconds, for which the Citrix ADC waits before sending another DNS query to resolve the host name of the audit server if the last query failed. Default value: 5 Minimum value = 5

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
			Maximum value = 20939
serverport	<Integer>	Read-write	Port on which the nslog server accepts connections. Minimum value = 1
loglevel	<String[]>	Read-write	Audit log level, which specifies the types of events to log. Available settings function as follows: * ALL - All events. * EMERGENCY - Events that indicate an immediate crisis on the server. * ALERT - Events that might require action. * CRITICAL - Events that indicate an imminent server crisis. * ERROR - Events that indicate some type of error. * WARNING - Events that require action in the near future.

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
			* NOTICE - Events that the administrator should know about. * INFORMATIONAL - All but low-level events. * DEBUG - All events, in extreme detail. * NONE - No events. Possible values = ALL, EMERGENCY, ALERT, CRITICAL, ERROR, WARNING, NOTICE, INFORMATIONAL, DEBUG, NONE
dateformat	<String>	Read-write	Format of dates in the logs. Supported formats are: * MMDDYYYY - U.S. style month/date/year format. * DDMMYYYY - European style date/month/year format.

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
			* YYYYMMDD - ISO style year/ month/date format. Possible values = MMDDYYYY, DDMMYYYY, YYYYMMDD
logfacility	<String>	Read-write	Facility value, as defined in RFC 3164, assigned to the log message. Log facility values are numbers 0 to 7 (LOCAL0 through LOCAL7). Each number indicates where a specific message originated from, such as the Citrix ADC itself, the VPN, or external. Possible values = LOCAL0, LOCAL1, LOCAL2, LOCAL3, LOCAL4, LOCAL5, LOCAL6, LOCAL7
tcp	<String>	Read-write	Log TCP messages. Possible values = NONE, ALL
acl	<String>	Read-write	Log access control list (ACL) messages.

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
			Possible values = ENABLED, DISABLED
timezone	<String>	Read-write	Time zone used for date and timestamps in the logs. Available settings function as follows: * GMT_TIME. Coordinated Universal Time. * LOCAL_TIME. The server's timezone setting. Possible values = GMT_TIME, LOCAL_TIME
userdefinedauditlog	<String>	Read-write	Log user-configurable log messages to nslog. Setting this parameter to NO causes auditing to ignore all user-configured message actions. Setting this parameter to YES causes auditing to log user-configured message actions that meet the other logging criteria.

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
			Possible values = YES, NO
appflowexport	<String>	Read-write	Export log messages to AppFlow collectors. Appflow collectors are entities to which log messages can be sent so that some action can be performed on them. Possible values = ENABLED, DISABLED
lsn	<String>	Read-write	Log the LSN messages. Possible values = ENABLED, DISABLED
alg	<String>	Read-write	Log the ALG messages. Possible values = ENABLED, DISABLED
subscriberlog	<String>	Read-write	Log subscriber session event information. Possible values = ENABLED, DISABLED
sslinterception	<String>	Read-write	Log SSL Interception event information.

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
			Possible values = ENABLED, DISABLED
urlfiltering	<String>	Read-write	Log URL filtering event information. Possible values = ENABLED, DISABLED
contentinspectionlog	<String>	Read-write	Log Content Inspection event information. Possible values = ENABLED, DISABLED
domainresolvenow	<Boolean>	Read-write	Immediately send a DNS query to resolve the server's domain name.
ip	<String>	Read-only	The resolved IP address of the auditserver.
__count	<Double>	Read-only	count parameter

Operations

(click to see Properties)

ADD| DELETE| UPDATE| UNSET| GET (ALL)| GET| COUNT

Some options that you can use for each operations:

- **Getting warnings in response:**NITRO allows you to get warnings in an operation by specifying the "warning" query parameter as "yes". For example, to get warnings while connecting to the Citrix ADC appliance, the URL is as follows:

  http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/login?warning=yes

If any, the warnings are displayed in the response payload with the HTTP code "209 X-NITRO-WARNING".

- **Authenticated access for individual NITRO operations:**NITRO allows you to logon to the Citrix ADC appliance to perform individual operations. You can use this option instead of creating a NITRO session (using the login object) and then using that session to perform all operations,

  To do this, you must specify the username and password in the request header of the NITRO request as follows:

  X-NITRO-USER:<username>

  X-NITRO-PASS:<password>

  Note: In such cases, make sure that the request header DOES not include the following:

  Cookie:NITRO_AUTH_TOKEN=<tokenvalue>

`*Note:`*

Mandatory parameters are marked in redand placeholder content is marked in <green>.

add

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/auditnslogaction

HTTP Method:POST

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Content-Type:application/json

Request Payload:

```
{"auditnslogaction":{

<b>"name":<String_value>,

</b>"serverip":<String_value>,

"serverdomainname":<String_value>,
```

```
"domainresolveretry":<Integer_value>,

"serverport":<Integer_value>,

<b>"loglevel":<String[]_value>,

</b>"dateformat":<String_value>,

"logfacility":<String_value>,

"tcp":<String_value>,

"acl":<String_value>,

"timezone":<String_value>,

"userdefinedauditlog":<String_value>,

"appflowexport":<String_value>,

"lsn":<String_value>,

"alg":<String_value>,

"subscriberlog":<String_value>,

"sslinterception":<String_value>,

"urlfiltering":<String_value>,

"contentinspectionlog":<String_value>

}}
```

Response:

HTTP Status Code on Success: 201 Created HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error

delete

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/auditnslogaction/name_value<String>

HTTP Method:DELETE

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue>

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error

update

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/auditnslogaction

HTTP Method:PUT

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Content-Type:application/json

Request Payload:

```
{"auditnslogaction":{

<b>"name":<String_value>,

</b>"serverip":<String_value>,

"serverdomainname":<String_value>,

"domainresolveretry":<Integer_value>,

"domainresolvenow":<Boolean_value>,

"serverport":<Integer_value>,

"loglevel":<String[]_value>,
```

```
"dateformat":<String_value>,

"logfacility":<String_value>,

"tcp":<String_value>,

"acl":<String_value>,

"timezone":<String_value>,

"userdefinedauditlog":<String_value>,

"appflowexport":<String_value>,

"lsn":<String_value>,

"alg":<String_value>,

"subscriberlog":<String_value>,

"sslinterception":<String_value>,

"urlfiltering":<String_value>,

"contentinspectionlog":<String_value>

}}
```

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error

unset

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/auditnslogaction?**action=unset**

HTTP Method:POST

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Content-Type:application/json

Request Payload:

```
{"auditnslogaction":{

<b>"name":<String_value>,

</b>"serverport":true,

"loglevel":true,

"dateformat":true,

"logfacility":true,

"tcp":true,

"acl":true,

"timezone":true,

"userdefinedauditlog":true,

"appflowexport":true,

"lsn":true,

"alg":true,

"subscriberlog":true,

"sslinterception":true,

"contentinspectionlog":true,

"urlfiltering":true

}}
```

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error

get (all)

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/auditnslogaction

Query-parameters:

attrs

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/auditnslogaction?**attrs=property-name1,property-name2**

Use this query parameter to specify the resource details that you want to retrieve.

filter

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/auditnslogaction?**filter=property-name1:property-val1,property-name2:property-val2**

Use this query-parameter to get the filtered set of auditnslogaction resources configured on Citrix ADC. Filtering can be done on any of the properties of the resource.

view

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/auditnslogaction?**view=summary**

Note: By default, the retrieved results are displayed in detail view (?view=detail).

pagination

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/auditnslogaction?**pagesize=#no;pageno=#no**

Use this query-parameter to get the auditnslogaction resources in chunks.

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "auditnslogaction": [ {

"name":<String_value>,

"serverip":<String_value>,

"serverdomainname":<String_value>,

"domainresolveretry":<Integer_value>,

"domainresolvenow":<Boolean_value>,

"ip":<String_value>,

"serverport":<Integer_value>,

"loglevel":<String[]_value>,

"dateformat":<String_value>,

"logfacility":<String_value>,

"tcp":<String_value>,

"acl":<String_value>,

"timezone":<String_value>,

"userdefinedauditlog":<String_value>,

"appflowexport":<String_value>,
```

```
"lsn":<String_value>,

"alg":<String_value>,

"subscriberlog":<String_value>,

"sslinterception":<String_value>,

"urlfiltering":<String_value>,

"contentinspectionlog":<String_value>

}]}
```

get

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/auditnslogaction/name_value<String>

Query-parameters:

attrs

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/auditnslogaction/name_value<String>?**attrs=property-name1,property-name2**

Use this query parameter to specify the resource details that you want to retrieve.

view

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/auditnslogaction/name_value<String>?**view=summary**

Note: By default, the retrieved results are displayed in detail view (?view=detail).

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "auditnslogaction": [ {

"name":<String_value>,

"serverip":<String_value>,

"serverdomainname":<String_value>,

"domainresolveretry":<Integer_value>,

"domainresolvenow":<Boolean_value>,

"ip":<String_value>,

"serverport":<Integer_value>,

"loglevel":<String[]_value>,

"dateformat":<String_value>,

"logfacility":<String_value>,

"tcp":<String_value>,

"acl":<String_value>,

"timezone":<String_value>,

"userdefinedauditlog":<String_value>,

"appflowexport":<String_value>,

"lsn":<String_value>,
```

```
"alg":<String_value>,

"subscriberlog":<String_value>,

"sslinterception":<String_value>,

"urlfiltering":<String_value>,

"contentinspectionlog":<String_value>

}]}
```

count

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/auditnslogaction?**count=yes**

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "auditnslogaction": [ { "__count": "#no"} ] }
```

auditnslogglobal_auditnslogpolicy_binding

Binding object showing the auditnslogpolicy that can be bound to auditnslogglobal.

Properties

(click to see Operations)

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
priority	<Double>	Read-write	Specifies the priority of the policy. Minimum value = 1 Maximum value = 2147483647
globalbindtype	<String>	Read-write	. Default value: SYSTEM_GLOBAL Possible values = SYSTEM_GLOBAL, VPN_GLOBAL, RNAT_GLOBAL
policyname	<String>	Read-write	Name of the audit nslog policy.
numpol	<Double>	Read-only	number of polices bound to label.
__count	<Double>	Read-only	count parameter

Operations

(click to see Properties)

ADD:| DELETE:| GET| GET (ALL)| COUNT

Some options that you can use for each operations:

- **Getting warnings in response:**NITRO allows you to get warnings in an operation by specifying the "warning" query parameter as "yes". For example, to get warnings while connecting to the Citrix ADC appliance, the URL is as follows:

  http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/login?warning=yes

If any, the warnings are displayed in the response payload with the HTTP code "209 X-NITRO-WARNING".

- **Authenticated access for individual NITRO operations:**NITRO allows you to logon to the Citrix ADC appliance to perform individual operations. You can use this option instead of creating a NITRO session (using the login object) and then using that session to perform all operations,

  To do this, you must specify the username and password in the request header of the NITRO request as follows:

  X-NITRO-USER:<username>

  X-NITRO-PASS:<password>

  Note: In such cases, make sure that the request header DOES not include the following:

  Cookie:NITRO_AUTH_TOKEN=<tokenvalue>

`*Note:`*

Mandatory parameters are marked in redand placeholder content is marked in <green>.

add

URL:http://<Citrix-ADC-IP-address(NSIP)/nitro/v1/config/auditnslogglobal_auditnslogpolicy_binding

HTTP Method:PUT

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Content-Type:application/json

Request Payload:

```
{

"auditnslogglobal_auditnslogpolicy_binding":{

<b>"policyname":<String_value>,

</b><b>"priority":<Double_value>,
```

```
</b>"globalbindtype":<String_value>

}}
```

Response:

HTTP Status Code on Success: 201 Created HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error

delete

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/auditnslogglobal_auditnslogpolicy_binding

Query-parameters:

args

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/auditnslogglobal_auditnslogpolicy_binding?**args=policyname:<String_value>,globalbindtype:<String_value>**

HTTP Method:DELETE

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue>

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error

get

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/auditnslogglobal_auditnslogpolicy_binding

Query-parameters:

filter

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/auditnslogglobal_auditnslogpolicy_binding?**filter=property-name1:property-value1,property-name2:property-value2**

Use this query-parameter to get the filtered set of auditnslogglobal_auditnslogpolicy_binding resources configured on Citrix ADC. Filtering can be done on any of the properties of the resource.

pagination

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/auditnslogglobal_auditnslogpolicy_binding?**pagesize=#no;pageno=#no**

Use this query-parameter to get the auditnslogglobal_auditnslogpolicy_binding resources in chunks.

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "auditnslogglobal_auditnslogpolicy_binding": [ {

"priority":<Double_value>,

"globalbindtype":<String_value>,

"policyname":<String_value>,

"numpol":<Double_value>

}]}
```

get (all)

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/auditnslogglobal_auditnslogpolicy_binding

Query-parameters:

bulkbindings

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/auditnslogglobal_auditnslogpolicy_binding?**bulkbindings=yes**

NITRO allows you to fetch bindings in bulk.

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "auditnslogglobal_auditnslogpolicy_binding": [ {

"priority":<Double_value>,

"globalbindtype":<String_value>,

"policyname":<String_value>,

"numpol":<Double_value>

}]}
```

count

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/auditnslogglobal_auditnslogpolicy_binding?**count=yes**

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "auditnslogglobal_auditnslogpolicy_binding": [ { "__count": "#no"} ] }
```

auditnslogglobal_binding

Binding object which returns the resources bound to auditnslogglobal.

Properties

(click to see Operations)

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
auditnslogglobal_auditnslogpolicy_binding	<auditnslogglobal_auditnslogpolicy_binding[]>	Read-only	auditnslogpolicy that can be bound to auditnslogglobal.

Operations

(click to see Properties)

GET| GET (ALL)

Some options that you can use for each operations:

- **Getting warnings in response:**NITRO allows you to get warnings in an operation by specifying the "warning" query parameter as "yes". For example, to get warnings while connecting to the Citrix ADC appliance, the URL is as follows:

  http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/login?warning=yes

  If any, the warnings are displayed in the response payload with the HTTP code "209 X-NITRO-WARNING".

- **Authenticated access for individual NITRO operations:**NITRO allows you to logon to the Citrix ADC appliance to perform individual operations. You can use this option instead of creating a NITRO session (using the login object) and then using that session to perform all operations,

  To do this, you must specify the username and password in the request header of the NITRO request as follows:

  X-NITRO-USER:<username>

  X-NITRO-PASS:<password>

  Note: In such cases, make sure that the request header DOES not include the following:

  Cookie:NITRO_AUTH_TOKEN=<tokenvalue>

`*Note:`*

Mandatory parameters are marked in redand placeholder content is marked in <green>.

get

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/auditnslogglobal_binding

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "auditnslogglobal_binding": [ {

"auditnslogglobal_auditnslogpolicy_binding":<auditnslogglobal_auditnslogpolicy_binding[]_value>

}]}
```

get (all)

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/auditnslogglobal_binding

Query-parameters:

bulkbindings

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/auditnslogglobal_binding?**bulkbindings=yes**

NITRO allows you to fetch bindings in bulk.

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "auditnslogglobal_binding": [ {

"auditnslogglobal_auditnslogpolicy_binding":<auditnslogglobal_auditnslogpolicy_binding[]_value>

}]}
```

auditnslogparams

Configuration for ns log parameters resource.

Properties

(click to see Operations)

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
serverip	<String>	Read-write	IP address of the nslog server. Minimum length = 1
serverport	<Integer>	Read-write	Port on which the nslog server accepts connections. Minimum value = 1
dateformat	<String>	Read-write	Format of dates in the logs. Supported formats are: * MMDDYYYY - U.S. style month/date/year format.

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
			* DDMMYYYY - European style date/month/year format. * YYYYMMDD - ISO style year/month/date format. Possible values = MMDDYYYY, DDMMYYYY, YYYYMMDD
loglevel	<String[]>	Read-write	Types of information to be logged. Available settings function as follows: * ALL - All events. * EMERGENCY - Events that indicate an immediate crisis on the server. * ALERT - Events that might require action. * CRITICAL - Events that indicate an imminent server crisis. * ERROR - Events that indicate some type of error.

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
			* WARNING - Events that require action in the near future. * NOTICE - Events that the administrator should know about. * INFORMATIONAL - All but low-level events. * DEBUG - All events, in extreme detail. * NONE - No events. Possible values = ALL, EMERGENCY, ALERT, CRITICAL, ERROR, WARNING, NOTICE, INFORMATIONAL, DEBUG, NONE
logfacility	<String>	Read-write	Facility value, as defined in RFC 3164, assigned to the log message. Log facility values are numbers 0 to 7 (LOCAL0 through LOCAL7). Each number indicates

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
			where a specific message originated from, such as the Citrix ADC itself, the VPN, or external. Possible values = LOCAL0, LOCAL1, LOCAL2, LOCAL3, LOCAL4, LOCAL5, LOCAL6, LOCAL7
tcp	<String>	Read-write	Configure auditing to log TCP messages. Possible values = NONE, ALL
acl	<String>	Read-write	Configure auditing to log access control list (ACL) messages. Possible values = ENABLED, DISABLED
timezone	<String>	Read-write	Time zone used for date and timestamps in the logs. Supported settings are: * GMT_TIME - Coordinated Universal Time. * LOCAL_TIME - Use the server's timezone setting.

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
			Possible values = GMT_TIME, LOCAL_TIME
userdefinedauditlog	<String>	Read-write	Log user-configurable log messages to nslog. Setting this parameter to NO causes auditing to ignore all user-configured message actions. Setting this parameter to YES causes auditing to log user-configured message actions that meet the other logging criteria. Possible values = YES, NO
appflowexport	<String>	Read-write	Export log messages to AppFlow collectors. Appflow collectors are entities to which log messages can be sent so that some action can be performed on them. Possible values = ENABLED, DISABLED

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
lsn	<String>	Read-write	Log the LSN messages. Possible values = ENABLED, DISABLED
alg	<String>	Read-write	Log the ALG messages. Possible values = ENABLED, DISABLED
subscriberlog	<String>	Read-write	Log subscriber session event information. Possible values = ENABLED, DISABLED
sslinterception	<String>	Read-write	Log SSL Interception event information. Possible values = ENABLED, DISABLED
urlfiltering	<String>	Read-write	Log URL filtering event information. Possible values = ENABLED, DISABLED
contentinspectionlog	<String>	Read-write	Log Content Inspection event information. Possible values = ENABLED, DISABLED

Operations

(click to see Properties)

UPDATE| UNSET| GET (ALL)

Some options that you can use for each operations:

- **Getting warnings in response:**NITRO allows you to get warnings in an operation by specifying the "warning" query parameter as "yes". For example, to get warnings while connecting to the Citrix ADC appliance, the URL is as follows:

  http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/login?warning=yes

  If any, the warnings are displayed in the response payload with the HTTP code "209 X-NITRO-WARNING".

- **Authenticated access for individual NITRO operations:**NITRO allows you to logon to the Citrix ADC appliance to perform individual operations. You can use this option instead of creating a NITRO session (using the login object) and then using that session to perform all operations,

  To do this, you must specify the username and password in the request header of the NITRO request as follows:

  X-NITRO-USER:<username>

  X-NITRO-PASS:<password>

  Note: In such cases, make sure that the request header DOES not include the following:

  Cookie:NITRO_AUTH_TOKEN=<tokenvalue>

`*Note:`*

Mandatory parameters are marked in redand placeholder content is marked in <green>.

update

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/auditnslogparams

HTTP Method:PUT

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Content-Type:application/json

Request Payload:

```
{"auditnslogparams":{

"serverip":<String_value>,

"serverport":<Integer_value>,

"dateformat":<String_value>,

"loglevel":<String[]_value>,

"logfacility":<String_value>,

"tcp":<String_value>,

"acl":<String_value>,

"timezone":<String_value>,

"userdefinedauditlog":<String_value>,

"appflowexport":<String_value>,

"lsn":<String_value>,

"alg":<String_value>,

"subscriberlog":<String_value>,

"sslinterception":<String_value>,

"urlfiltering":<String_value>,

"contentinspectionlog":<String_value>

}}
```

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error

unset

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/auditnslogparams?**action=unset**

HTTP Method:POST

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Content-Type:application/json

Request Payload:

```
{"auditnslogparams":{

"serverip":true,

"serverport":true,

"loglevel":true,

"dateformat":true,

"logfacility":true,

"tcp":true,

"acl":true,

"timezone":true,

"userdefinedauditlog":true,

"appflowexport":true,

"lsn":true,

"alg":true,
```

```
"subscriberlog":true,

"sslinterception":true,

"urlfiltering":true,

"contentinspectionlog":true

}}
```

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error

get (all)

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/auditnslogparams

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "auditnslogparams": [ {

"name":<String_value>,
```

```
"serverip":<String_value>,

"serverport":<Integer_value>,

"dateformat":<String_value>,

"loglevel":<String[]_value>,

"logfacility":<String_value>,

"tcp":<String_value>,

"acl":<String_value>,

"timezone":<String_value>,

"userdefinedauditlog":<String_value>,

"appflowexport":<String_value>,

"lsn":<String_value>,

"alg":<String_value>,

"subscriberlog":<String_value>,

"sslinterception":<String_value>,

"urlfiltering":<String_value>,

"contentinspectionlog":<String_value>

}]}
```

auditnslogpolicy

Configuration for ns log policy resource.

Properties

(click to see Operations)

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
name	<String>	Read-write	Name for the policy. Must begin with a letter, number, or the underscore character (_), and must consist only of letters, numbers, and the hyphen (-), period (.) pound (#), space ( ), at sign (@), equals (=), colon (:), and underscore characters. Cannot be changed after the nslog policy is added. The following requirement applies only to the Citrix ADC CLI: If the name includes one or more spaces, enclose the name in double or single quotation marks (for example, "my nslog policy" or 'my nslog policy'). Minimum length = 1
rule	<String>	Read-write	Name of the Citrix ADC named rule, or an expression, that defines the

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
			messages to be logged to the nslog server. Minimum length = 1
action	<String>	Read-write	Nslog server action that is performed when this policy matches. NOTE: An nslog server action must be associated with an nslog audit policy. Minimum length = 1
expressiontype	<String>	Read-only	Type of policy (Classic/Advanced). Possible values = Classic Policy, Advanced Policy
__count	<Double>	Read-only	count parameter

Operations

(click to see Properties)

ADD| DELETE| UPDATE| GET (ALL)| GET| COUNT

Some options that you can use for each operations:

- **Getting warnings in response:**NITRO allows you to get warnings in an operation by specifying the "warning" query parameter as "yes". For example, to get warnings while connecting to the Citrix ADC appliance, the URL is as follows:

  http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/login?warning=yes

  If any, the warnings are displayed in the response payload with the HTTP code "209 X-NITRO-WARNING".

- **Authenticated access for individual NITRO operations:**NITRO allows you to logon to the Citrix ADC appliance to perform individual operations. You can use this option instead of creating a NITRO session (using the login object) and then using that session to perform all operations,

  To do this, you must specify the username and password in the request header of the NITRO request as follows:

  X-NITRO-USER:<username>

  X-NITRO-PASS:<password>

  Note: In such cases, make sure that the request header DOES not include the following:

  Cookie:NITRO_AUTH_TOKEN=<tokenvalue>

`*Note:`*

Mandatory parameters are marked in redand placeholder content is marked in <green>.

add

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/auditnslogpolicy

HTTP Method:POST

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Content-Type:application/json

Request Payload:

```
{"auditnslogpolicy":{

<b>"name":<String_value>,

</b><b>"rule":<String_value>,

</b><b>"action":<String_value>

</b>}}
```

Response:

HTTP Status Code on Success: 201 Created HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error

delete

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/auditnslogpolicy/name_value<String>

HTTP Method:DELETE

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue>

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error

update

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/auditnslogpolicy

HTTP Method:PUT

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Content-Type:application/json

Request Payload:

```
{"auditnslogpolicy":{

<b>"name":<String_value>,

</b>"rule":<String_value>,

"action":<String_value>

}}
```

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error

get (all)

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/auditnslogpolicy

Query-parameters:

attrs

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/auditnslogpolicy?**attrs=property-name1,property-name2**

Use this query parameter to specify the resource details that you want to retrieve.

filter

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/auditnslogpolicy?**filter=property-name1:property-val1,property-name2:property-val2**

Use this query-parameter to get the filtered set of auditnslogpolicy resources configured on Citrix ADC. Filtering can be done on any of the properties of the resource.

view

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/auditnslogpolicy?**view=summary**

Note: By default, the retrieved results are displayed in detail view (?view=detail).

pagination

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/auditnslogpolicy?**pagesize=#no;pageno=#no**

Use this query-parameter to get the auditnslogpolicy resources in chunks.

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "auditnslogpolicy": [ {

"name":<String_value>,

"rule":<String_value>,

"action":<String_value>,

"expressiontype":<String_value>

}]}
```

get

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/auditnslogpolicy/name_value<String>

Query-parameters:

attrs

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/auditnslogpolicy/name_value<String>?**attrs=property-name1,property-name2**

Use this query parameter to specify the resource details that you want to retrieve.

view

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/auditnslogpolicy/name_value<String>?**view=summary**

Note: By default, the retrieved results are displayed in detail view (?view=detail).

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "auditnslogpolicy": [ {

"name":<String_value>,

"rule":<String_value>,

"action":<String_value>,

"expressiontype":<String_value>

}]}
```

count

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/auditnslogpolicy?**count=yes**

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "auditnslogpolicy": [ { "__count": "#no"} ] }
```

auditnslogpolicy_aaagroup_binding

Binding object showing the aaagroup that can be bound to auditnslogpolicy.

Properties

(click to see Operations)

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
name	<String>	Read-write	Name of the policy. Minimum length = 1
boundto	<String>	Read-write	The entity name to which policy is bound.
priority	<Double>	Read-only	.
activepolicy	<Double>	Read-only	.
__count	<Double>	Read-only	count parameter

Operations

(click to see Properties)

GET| GET (ALL)| COUNT

Some options that you can use for each operations:

- **Getting warnings in response:**NITRO allows you to get warnings in an operation by specifying the "warning" query parameter as "yes". For example, to get warnings while connecting to the Citrix ADC appliance, the URL is as follows:

  http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/login?warning=yes

  If any, the warnings are displayed in the response payload with the HTTP code "209 X-NITRO-WARNING".

- **Authenticated access for individual NITRO operations:**NITRO allows you to logon to the Citrix ADC appliance to perform individual operations. You can use this option instead of creating a NITRO session (using the login object) and then using that session to perform all operations,

  To do this, you must specify the username and password in the request header of the NITRO request as follows:

  X-NITRO-USER:<username>

  X-NITRO-PASS:<password>

  Note: In such cases, make sure that the request header DOES not include the following:

  Cookie:NITRO_AUTH_TOKEN=<tokenvalue>

`*Note:`*

Mandatory parameters are marked in redand placeholder content is marked in <green>.

get

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/auditnslogpolicy_aaagroup_binding/name_value<String>

Query-parameters:

filter

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/auditnslogpolicy_aaagroup_binding/name_value<String>?**filter=property-name1:property-value1,property-name2:property-value2**

Use this query-parameter to get the filtered set of auditnslogpolicy_aaagroup_binding resources configured on Citrix ADC. Filtering can be done on any of the properties of the resource.

pagination

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/auditnslogpolicy_aaagroup_binding/name_value<String>?**pagesize=#no;pageno=#no**

Use this query-parameter to get the auditnslogpolicy_aaagroup_binding resources in chunks.

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "auditnslogpolicy_aaagroup_binding": [ {

"name":<String_value>,

"boundto":<String_value>,

"priority":<Double_value>,

"activepolicy":<Double_value>

}]}
```

get (all)

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/auditnslogpolicy_aaagroup_binding

Query-parameters:

bulkbindings

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/auditnslogpolicy_aaagroup_binding?**bulkbindings=yes**

NITRO allows you to fetch bindings in bulk.

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "auditnslogpolicy_aaagroup_binding": [ {

"name":<String_value>,

"boundto":<String_value>,

"priority":<Double_value>,

"activepolicy":<Double_value>

}]}
```

count

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/auditnslogpolicy_aaagroup_binding/name_value<String>?**count=yes**

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{"auditnslogpolicy_aaagroup_binding": [ { "__count": "#no"} ] }
```

auditnslogpolicy_aaauser_binding

Binding object showing the aaauser that can be bound to auditnslogpolicy.

Properties

(click to see Operations)

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
name	<String>	Read-write	Name of the policy. Minimum length = 1
boundto	<String>	Read-write	The entity name to which policy is bound.
priority	<Double>	Read-only	.
activepolicy	<Double>	Read-only	.
__count	<Double>	Read-only	count parameter

Operations

(click to see Properties)

GET| GET (ALL)| COUNT

Some options that you can use for each operations:

- **Getting warnings in response:**NITRO allows you to get warnings in an operation by specifying the "warning" query parameter as "yes". For example, to get warnings while connecting to the Citrix ADC appliance, the URL is as follows:

  http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/login?warning=yes

  If any, the warnings are displayed in the response payload with the HTTP code "209 X-NITRO-WARNING".

- **Authenticated access for individual NITRO operations:**NITRO allows you to logon to the Citrix ADC appliance to perform individual operations. You can use this option instead of creating a NITRO session (using the login object) and then using that session to perform all operations,

  To do this, you must specify the username and password in the request header of the NITRO request as follows:

  X-NITRO-USER:<username>

  X-NITRO-PASS:<password>

  Note: In such cases, make sure that the request header DOES not include the following:

  Cookie:NITRO_AUTH_TOKEN=<tokenvalue>

`*Note:`*

Mandatory parameters are marked in redand placeholder content is marked in <green>.

get

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/auditnslogpolicy_aaauser_binding/name_value<String>

Query-parameters:

filter

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/auditnslogpolicy_aaauser_binding/name_value<String>?**filter=property-name1:property-value1,property-name2:property-value2**

Use this query-parameter to get the filtered set of auditnslogpolicy_aaauser_binding resources configured on Citrix ADC. Filtering can be done on any of the properties of the resource.

pagination

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/auditnslogpolicy_aaauser_binding/name_value<String>?**pagesize=#no;pageno=#no**

Use this query-parameter to get the auditnslogpolicy_aaauser_binding resources in chunks.

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "auditnslogpolicy_aaauser_binding": [ {

"name":<String_value>,

"boundto":<String_value>,

"priority":<Double_value>,

"activepolicy":<Double_value>

}]}
```

get (all)

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/auditnslogpolicy_aaauser_binding

Query-parameters:

bulkbindings

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/auditnslogpolicy_aaauser_binding?**bulkbindings=yes**

NITRO allows you to fetch bindings in bulk.

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "auditnslogpolicy_aaauser_binding": [ {

"name":<String_value>,

"boundto":<String_value>,

"priority":<Double_value>,

"activepolicy":<Double_value>

}]}
```

count

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/auditnslogpolicy_aaauser_binding/name_value<String>?**count=yes**

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{"auditnslogpolicy_aaauser_binding": [ { "__count": "#no"} ] }
```

auditnslogpolicy_appfwglobal_binding

Binding object showing the appfwglobal that can be bound to auditnslogpolicy.

Properties

(click to see Operations)

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
name	<String>	Read-write	Name of the policy. Minimum length = 1
boundto	<String>	Read-write	The entity name to which policy is bound.
priority	<Double>	Read-only	.
activepolicy	<Double>	Read-only	.
__count	<Double>	Read-only	count parameter

Operations

(click to see Properties)

GET| GET (ALL)| COUNT

Some options that you can use for each operations:

- **Getting warnings in response:**NITRO allows you to get warnings in an operation by specifying the "warning" query parameter as "yes". For example, to get warnings while connecting to the Citrix ADC appliance, the URL is as follows:

  http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/login?warning=yes

  If any, the warnings are displayed in the response payload with the HTTP code "209 X-NITRO-WARNING".

- **Authenticated access for individual NITRO operations:**NITRO allows you to logon to the Citrix ADC appliance to perform individual operations. You can use this option instead of creating a NITRO session (using the login object) and then using that session to perform all operations,

  To do this, you must specify the username and password in the request header of the NITRO request as follows:

  X-NITRO-USER:<username>

  X-NITRO-PASS:<password>

  Note: In such cases, make sure that the request header DOES not include the following:

  Cookie:NITRO_AUTH_TOKEN=<tokenvalue>

`*Note:`*

Mandatory parameters are marked in redand placeholder content is marked in <green>.

get

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/auditnslogpolicy_appfwglobal_binding/name_value<String>

Query-parameters:

filter

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/auditnslogpolicy_appfwglobal_binding/name_value<String>?**filter=property-name1:property-value1,property-name2:property-value2**

Use this query-parameter to get the filtered set of auditnslogpolicy_appfwglobal_binding resources configured on Citrix ADC. Filtering can be done on any of the properties of the resource.

pagination

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/auditnslogpolicy_appfwglobal_binding/name_value<String>?**pagesize=#no;pageno=#no**

Use this query-parameter to get the auditnslogpolicy_appfwglobal_binding resources in chunks.

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "auditnslogpolicy_appfwglobal_binding": [ {

"name":<String_value>,

"boundto":<String_value>,

"priority":<Double_value>,

"activepolicy":<Double_value>

}]}
```

get (all)

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/auditnslogpolicy_appfwglobal_binding

Query-parameters:

bulkbindings

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/auditnslogpolicy_appfwglobal_binding?**bulkbindings=yes**

NITRO allows you to fetch bindings in bulk.

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "auditnslogpolicy_appfwglobal_binding": [ {

"name":<String_value>,

"boundto":<String_value>,

"priority":<Double_value>,

"activepolicy":<Double_value>

}]}
```

count

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/auditnslogpolicy_appfwglobal_binding/name_value<String>?**count=yes**

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{"auditnslogpolicy_appfwglobal_binding": [ { "__count": "#no"} ] }
```

auditnslogpolicy_auditnslogglobal_binding

Binding object showing the auditnslogglobal that can be bound to auditnslogpolicy.

Properties

(click to see Operations)

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
name	<String>	Read-write	Name of the policy. Minimum length = 1
boundto	<String>	Read-write	The entity name to which policy is bound.
priority	<Double>	Read-only	.

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
activepolicy	<Double>	Read-only	.
__count	<Double>	Read-only	count parameter

Operations

(click to see Properties)

GET| GET (ALL)| COUNT

Some options that you can use for each operations:

- **Getting warnings in response:**NITRO allows you to get warnings in an operation by specifying the "warning" query parameter as "yes". For example, to get warnings while connecting to the Citrix ADC appliance, the URL is as follows:

  http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/login?warning=yes

  If any, the warnings are displayed in the response payload with the HTTP code "209 X-NITRO-WARNING".
- **Authenticated access for individual NITRO operations:**NITRO allows you to logon to the Citrix ADC appliance to perform individual operations. You can use this option instead of creating a NITRO session (using the login object) and then using that session to perform all operations,

  To do this, you must specify the username and password in the request header of the NITRO request as follows:

  X-NITRO-USER:<username>

  X-NITRO-PASS:<password>

  Note: In such cases, make sure that the request header DOES not include the following:

  Cookie:NITRO_AUTH_TOKEN=<tokenvalue>

`*Note:`*

Mandatory parameters are marked in redand placeholder content is marked in <green>.

get

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/auditnslogpolicy_auditnslogglobal_binding/name_value<String>

Query-parameters:

filter

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/auditnslogpolicy_auditnslogglobal_binding/name_value<String>?**filter=property-name1:property-value1,property-name2:property-value2**

Use this query-parameter to get the filtered set of auditnslogpolicy_auditnslogglobal_binding resources configured on Citrix ADC. Filtering can be done on any of the properties of the resource.

pagination

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/auditnslogpolicy_auditnslogglobal_binding/name_value<String>?**pagesize=#no;pageno=#no**

Use this query-parameter to get the auditnslogpolicy_auditnslogglobal_binding resources in chunks.

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "auditnslogpolicy_auditnslogglobal_binding": [ {

"name":<String_value>,
```

```
"boundto":<String_value>,

"priority":<Double_value>,

"activepolicy":<Double_value>

}]}
```

get (all)

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/auditnslogpolicy_auditnslogglobal_binding

Query-parameters:

bulkbindings

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/auditnslogpolicy_auditnslogglobal_binding?**bulkbindings=yes**

NITRO allows you to fetch bindings in bulk.

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "auditnslogpolicy_auditnslogglobal_binding": [ {

"name":<String_value>,
```

```
"boundto":<String_value>,

"priority":<Double_value>,

"activepolicy":<Double_value>

}]}
```

count

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/auditnslogpolicy_auditnslogglobal_binding/name_value<String>?**count=yes**

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{"auditnslogpolicy_auditnslogglobal_binding": [ { "__count": "#no"} ] }
```

auditnslogpolicy_authenticationvserver_binding

Binding object showing the authenticationvserver that can be bound to auditnslogpolicy.

Properties

(click to see Operations)

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
name	<String>	Read-write	Name of the policy. Minimum length = 1
boundto	<String>	Read-write	The entity name to which policy is bound.
priority	<Double>	Read-only	.
activepolicy	<Double>	Read-only	.
__count	<Double>	Read-only	count parameter

Operations

(click to see Properties)

GET| GET (ALL)| COUNT

Some options that you can use for each operations:

- **Getting warnings in response:**NITRO allows you to get warnings in an operation by specifying the "warning" query parameter as "yes". For example, to get warnings while connecting to the Citrix ADC appliance, the URL is as follows:

  http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/login?warning=yes

  If any, the warnings are displayed in the response payload with the HTTP code "209 X-NITRO-WARNING".

- **Authenticated access for individual NITRO operations:**NITRO allows you to logon to the Citrix ADC appliance to perform individual operations. You can use this option instead of creating a NITRO session (using the login object) and then using that session to perform all operations,

  To do this, you must specify the username and password in the request header of the NITRO request as follows:

  X-NITRO-USER:<username>

  X-NITRO-PASS:<password>

Note: In such cases, make sure that the request header DOES not include the following:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue>

`*Note:`*

Mandatory parameters are marked in redand placeholder content is marked in <green>.

get

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/auditnslogpolicy_authenticationvserver_binding/name_value<String>

Query-parameters:

filter

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/auditnslogpolicy_authenticationvserver_binding/name_value<String>?**filter=property-name1:property-value1,property-name2:property-value2**

Use this query-parameter to get the filtered set of auditnslogpolicy_authenticationvserver_binding resources configured on Citrix ADC. Filtering can be done on any of the properties of the resource.

pagination

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/auditnslogpolicy_authenticationvserver_binding/name_value<String>?**pagesize=#no;pageno=#no**

Use this query-parameter to get the auditnslogpolicy_authenticationvserver_binding resources in chunks.

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "auditnslogpolicy_authenticationvserver_binding": [ {

"name":<String_value>,

"boundto":<String_value>,

"priority":<Double_value>,

"activepolicy":<Double_value>

}]}
```

get (all)

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/auditnslogpolicy_authenticationvserver_binding

Query-parameters:

bulkbindings

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/auditnslogpolicy_authenticationvserver_binding?**bulkbindings=yes**

NITRO allows you to fetch bindings in bulk.

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "auditnslogpolicy_authenticationvserver_binding": [ {

"name":<String_value>,

"boundto":<String_value>,

"priority":<Double_value>,

"activepolicy":<Double_value>

}]}
```

count

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/auditnslogpolicy_authenticationvserver_binding/name_value<String>?**count=yes**

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{"auditnslogpolicy_authenticationvserver_binding": [ { "__count":
"#no"} ] }
```

auditnslogpolicy_binding

Binding object which returns the resources bound to auditnslogpolicy.

Properties

(click to see Operations)

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
name	<String>	Read-write	Name of the policy. Minimum length = 1
auditnslogpolicy_tmglobal_binding	<auditnslogpolicy_tmglobal_binding[]>	Read-only	tmglobal that can be bound to auditnslogpolicy.
auditnslogpolicy_appfwglobal_binding	<auditnslogpolicy_appfwglobal_binding[]>	Read-only	appfwglobal that can be bound to auditnslogpolicy.
auditnslogpolicy_lbvserver_binding	<auditnslogpolicy_lbvserver_binding[]>	Read-only	lbvserver that can be bound to auditnslogpolicy.
auditnslogpolicy_vpnglobal_binding	<auditnslogpolicy_vpnglobal_binding[]>	Read-only	vpnglobal that can be bound to auditnslogpolicy.
auditnslogpolicy_vpnvserver_binding	<auditnslogpolicy_vpnvserver_binding[]>	Read-only	vpnvserver that can be bound to auditnslogpolicy.
auditnslogpolicy_auditnslogglobal_binding	<auditnslogpolicy_auditnslogglobal_binding[]>	Read-only	auditnslogglobal that can be bound to auditnslogpolicy.
auditnslogpolicy_csvserver_binding	<auditnslogpolicy_csvserver_binding[]>	Read-only	csvserver that can be bound to auditnslogpolicy.

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
auditnslogpolicy_aaauser_binding	<auditnslogpolicy_aaauser_binding[]>	Read-only	aaauser that can be bound to auditnslogpolicy.
auditnslogpolicy_systemglobal_binding	<auditnslogpolicy_systemglobal_binding[]>	Read-only	systemglobal that can be bound to auditnslogpolicy.
auditnslogpolicy_authenticationvserver_binding	<auditnslogpolicy_authenticationvserver_binding[]>	Read-only	authenticationvserver that can be bound to auditnslogpolicy.
auditnslogpolicy_aaagroup_binding	<auditnslogpolicy_aaagroup_binding[]>	Read-only	aaagroup that can be bound to auditnslogpolicy.

Operations

(click to see Properties)

GET| GET (ALL)

Some options that you can use for each operations:

- **Getting warnings in response:**NITRO allows you to get warnings in an operation by specifying the "warning" query parameter as "yes". For example, to get warnings while connecting to the Citrix ADC appliance, the URL is as follows:

  http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/login?warning=yes

  If any, the warnings are displayed in the response payload with the HTTP code "209 X-NITRO-WARNING".

- **Authenticated access for individual NITRO operations:**NITRO allows you to logon to the Citrix ADC appliance to perform individual operations. You can use this option instead of creating a NITRO session (using the login object) and then using that session to perform all operations,

  To do this, you must specify the username and password in the request header of the NITRO request as follows:

  X-NITRO-USER:<username>

X-NITRO-PASS:<password>

Note: In such cases, make sure that the request header DOES not include the following:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue>

`*Note:`*

Mandatory parameters are marked in redand placeholder content is marked in <green>.

get

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/auditnslogpolicy_binding/name_value<String>

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "auditnslogpolicy_binding": [ {

"name":<String_value>,

"auditnslogpolicy_systemglobal_binding":<auditnslogpolicy_systemglobal_binding[]_value>,

"auditnslogpolicy_aaagroup_binding":<auditnslogpolicy_aaagroup_binding[]_value>,

"auditnslogpolicy_lbvserver_binding":<auditnslogpolicy_lbvserver_binding[]_value>,
```

```
"auditnslogpolicy_csvserver_binding":<auditnslogpolicy_csvserver_binding[]
_value>,

"auditnslogpolicy_vpnglobal_binding":<auditnslogpolicy_vpnglobal_binding[]
_value>,

"auditnslogpolicy_aaauser_binding":<auditnslogpolicy_aaauser_binding[]_val
ue>,

"auditnslogpolicy_authenticationvserver_binding":<auditnslogpolicy_authent
icationvserver_binding[]_value>,

"auditnslogpolicy_auditnslogglobal_binding":<auditnslogpolicy_auditnsloggl
obal_binding[]_value>,

"auditnslogpolicy_vpnvserver_binding":<auditnslogpolicy_vpnvserver_binding
[]_value>,

"auditnslogpolicy_appfwglobal_binding":<auditnslogpolicy_appfwglobal_bindi
ng[]_value>,

"auditnslogpolicy_tmglobal_binding":<auditnslogpolicy_tmglobal_binding[]_v
alue>

}]}
```

get (all)

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/auditnslogpolicy_binding

Query-parameters:

bulkbindings

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/auditnslogpolicy_binding?**bulkbindings=yes**

NITRO allows you to fetch bindings in bulk.

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "auditnslogpolicy_binding": [ {

"name":<String_value>,

"auditnslogpolicy_systemglobal_binding":<auditnslogpolicy_systemglobal_bin
ding[]_value>,

"auditnslogpolicy_aaagroup_binding":<auditnslogpolicy_aaagroup_binding[]_v
alue>,

"auditnslogpolicy_lbvserver_binding":<auditnslogpolicy_lbvserver_binding[]
_value>,

"auditnslogpolicy_csvserver_binding":<auditnslogpolicy_csvserver_binding[]
_value>,

"auditnslogpolicy_vpnglobal_binding":<auditnslogpolicy_vpnglobal_binding[]
_value>,

"auditnslogpolicy_aaauser_binding":<auditnslogpolicy_aaauser_binding[]_val
ue>,

"auditnslogpolicy_authenticationvserver_binding":<auditnslogpolicy_authent
icationvserver_binding[]_value>,

"auditnslogpolicy_auditnslogglobal_binding":<auditnslogpolicy_auditnsloggl
obal_binding[]_value>,

"auditnslogpolicy_vpnvserver_binding":<auditnslogpolicy_vpnvserver_binding
[]_value>,
```

```
"auditnslogpolicy_appfwglobal_binding":<auditnslogpolicy_appfwglobal_binding[]_value>,

"auditnslogpolicy_tmglobal_binding":<auditnslogpolicy_tmglobal_binding[]_value>

}]}
```

auditnslogpolicy_csvserver_binding

Binding object showing the csvserver that can be bound to auditnslogpolicy.

Properties

(click to see Operations)

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
name	<String>	Read-write	Name of the policy. Minimum length = 1
boundto	<String>	Read-write	The entity name to which policy is bound.
priority	<Double>	Read-only	.
activepolicy	<Double>	Read-only	.
__count	<Double>	Read-only	count parameter

Operations

(click to see Properties)

GET| GET (ALL)| COUNT

Some options that you can use for each operations:

- **Getting warnings in response:**NITRO allows you to get warnings in an operation by specifying the "warning" query parameter as "yes". For example, to get warnings while connecting to the Citrix ADC appliance, the URL is as follows:

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/login?warning=yes

If any, the warnings are displayed in the response payload with the HTTP code "209 X-NITRO-WARNING".

- **Authenticated access for individual NITRO operations:**NITRO allows you to logon to the Citrix ADC appliance to perform individual operations. You can use this option instead of creating a NITRO session (using the login object) and then using that session to perform all operations,

  To do this, you must specify the username and password in the request header of the NITRO request as follows:

  X-NITRO-USER:<username>

  X-NITRO-PASS:<password>

  Note: In such cases, make sure that the request header DOES not include the following:

  Cookie:NITRO_AUTH_TOKEN=<tokenvalue>

`*Note:`*

Mandatory parameters are marked in redand placeholder content is marked in <green>.

get

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/auditnslogpolicy_csvserver_binding/name_value<String>

Query-parameters:

filter

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/auditnslogpolicy_csvserver_binding/name_value<String>?**filter=property-name1:property-value1,property-name2:property-value2**

Use this query-parameter to get the filtered set of auditnslogpolicy_csvserver_binding resources configured on Citrix ADC. Filtering can be done on any of the properties of the resource.

pagination

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/auditnslogpolicy_csvserver_binding/name_value<String>?**pagesize=#no;pageno=#no**

Use this query-parameter to get the auditnslogpolicy_csvserver_binding resources in chunks.

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "auditnslogpolicy_csvserver_binding": [ {

"name":<String_value>,

"boundto":<String_value>,

"priority":<Double_value>,

"activepolicy":<Double_value>

}]}
```

get (all)

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/auditnslogpolicy_csvserver_binding

Query-parameters:

bulkbindings

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/auditnslogpolicy_csvserver_binding?**bulkbindings=yes**

NITRO allows you to fetch bindings in bulk.

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "auditnslogpolicy_csvserver_binding": [ {

"name":<String_value>,

"boundto":<String_value>,

"priority":<Double_value>,

"activepolicy":<Double_value>

}]}
```

count

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/auditnslogpolicy_csvserver_binding/name_value<String>?**count=yes**

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{"auditnslogpolicy_csvserver_binding": [ { "__count": "#no"} ] }
```

auditnslogpolicy_lbvserver_binding

Binding object showing the lbvserver that can be bound to auditnslogpolicy.

Properties

(click to see Operations)

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
name	<String>	Read-write	Name of the policy. Minimum length = 1
boundto	<String>	Read-write	The entity name to which policy is bound.
priority	<Double>	Read-only	.
activepolicy	<Double>	Read-only	.
__count	<Double>	Read-only	count parameter

Operations

(click to see Properties)

GET| GET (ALL)| COUNT

Some options that you can use for each operations:

- **Getting warnings in response:**NITRO allows you to get warnings in an operation by specifying the "warning" query parameter as "yes". For example, to get warnings while connecting to the Citrix ADC appliance, the URL is as follows:

  http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/login?warning=yes

  If any, the warnings are displayed in the response payload with the HTTP code "209 X-NITRO-WARNING".

- **Authenticated access for individual NITRO operations:**NITRO allows you to logon to the Citrix ADC appliance to perform individual operations. You can use this option instead of creating a NITRO session (using the login object) and then using that session to perform all operations,

  To do this, you must specify the username and password in the request header of the NITRO request as follows:

  X-NITRO-USER:<username>

  X-NITRO-PASS:<password>

  Note: In such cases, make sure that the request header DOES not include the following:

  Cookie:NITRO_AUTH_TOKEN=<tokenvalue>

`*Note:`*

Mandatory parameters are marked in redand placeholder content is marked in <green>.

get

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/auditnslogpolicy_lbvserver_binding/name_value<String>

Query-parameters:

filter

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/auditnslogpolicy_lbvserver_binding/name_value<String>?**filter=property-name1:property-value1,property-name2:property-value2**

Use this query-parameter to get the filtered set of auditnslogpolicy_lbvserver_binding resources configured on Citrix ADC. Filtering can be done on any of the properties of the resource.

pagination

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/auditnslogpolicy_lbvserver_binding/name_value<String>?**pagesize=#no;pageno=#no**

Use this query-parameter to get the auditnslogpolicy_lbvserver_binding resources in chunks.

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "auditnslogpolicy_lbvserver_binding": [ {

"name":<String_value>,

"boundto":<String_value>,

"priority":<Double_value>,

"activepolicy":<Double_value>

}]}
```

get (all)

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/auditnslogpolicy_lbvserver_binding

Query-parameters:

bulkbindings

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/auditnslogpolicy_lbvserver_binding?**bulkbindings=yes**

NITRO allows you to fetch bindings in bulk.

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "auditnslogpolicy_lbvserver_binding": [ {

"name":<String_value>,

"boundto":<String_value>,

"priority":<Double_value>,

"activepolicy":<Double_value>

}]}
```

count

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/auditnslogpolicy_lbvserver_binding/name_value<String>?**count=yes**

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{"auditnslogpolicy_lbvserver_binding": [ { "__count": "#no"} ] }
```

auditnslogpolicy_systemglobal_binding

Binding object showing the systemglobal that can be bound to auditnslogpolicy.

Properties

(click to see Operations)

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
name	<String>	Read-write	Name of the policy. Minimum length = 1
boundto	<String>	Read-write	The entity name to which policy is bound.
priority	<Double>	Read-only	.
activepolicy	<Double>	Read-only	.
__count	<Double>	Read-only	count parameter

Operations

(click to see Properties)

GET| GET (ALL)| COUNT

Some options that you can use for each operations:

- **Getting warnings in response:**NITRO allows you to get warnings in an operation by specifying the "warning" query parameter as "yes". For example, to get warnings while connecting to the Citrix ADC appliance, the URL is as follows:

  http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/login?warning=yes

  If any, the warnings are displayed in the response payload with the HTTP code "209 X-NITRO-WARNING".

- **Authenticated access for individual NITRO operations:**NITRO allows you to logon to the Citrix ADC appliance to perform individual operations. You can use this option instead of creating a NITRO session (using the login object) and then using that session to perform all operations,

  To do this, you must specify the username and password in the request header of the NITRO request as follows:

  X-NITRO-USER:<username>

  X-NITRO-PASS:<password>

  Note: In such cases, make sure that the request header DOES not include the following:

  Cookie:NITRO_AUTH_TOKEN=<tokenvalue>

`*Note:`*

Mandatory parameters are marked in redand placeholder content is marked in <green>.

get

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/auditnslogpolicy_systemglobal_binding/name_value<String>

Query-parameters:

filter

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/auditnslogpolicy_systemglobal_binding/name_value<String>?**filter=property-name1:property-value1,property-name2:property-value2**

Use this query-parameter to get the filtered set of auditnslogpolicy_systemglobal_binding resources configured on Citrix ADC. Filtering can be done on any of the properties of the resource.

pagination

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/auditnslogpolicy_systemglobal_binding/name_value<String>?**pagesize=#no;pageno=#no**

Use this query-parameter to get the auditnslogpolicy_systemglobal_binding resources in chunks.

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "auditnslogpolicy_systemglobal_binding": [ {

"name":<String_value>,

"boundto":<String_value>,

"priority":<Double_value>,

"activepolicy":<Double_value>

}]}
```

get (all)

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/auditnslogpolicy_systemglobal_binding

Query-parameters:

bulkbindings

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/auditnslogpolicy_systemglobal_binding?**bulkbindings=yes**

NITRO allows you to fetch bindings in bulk.

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "auditnslogpolicy_systemglobal_binding": [ {

"name":<String_value>,

"boundto":<String_value>,

"priority":<Double_value>,

"activepolicy":<Double_value>

}]}
```

count

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/auditnslogpolicy_systemglobal_binding/name_value<String>?**count=yes**

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{"auditnslogpolicy_systemglobal_binding": [ { "__count": "#no"} ] }
```

auditnslogpolicy_tmglobal_binding

Binding object showing the tmglobal that can be bound to auditnslogpolicy.

Properties

(click to see Operations)

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
name	<String>	Read-write	Name of the policy. Minimum length = 1
boundto	<String>	Read-write	The entity name to which policy is bound.
priority	<Double>	Read-only	.
activepolicy	<Double>	Read-only	.
__count	<Double>	Read-only	count parameter

Operations

(click to see Properties)

GET| GET (ALL)| COUNT

Some options that you can use for each operations:

- **Getting warnings in response:**NITRO allows you to get warnings in an operation by specifying the "warning" query parameter as "yes". For example, to get warnings while connecting to the Citrix ADC appliance, the URL is as follows:

  http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/login?warning=yes

  If any, the warnings are displayed in the response payload with the HTTP code "209 X-NITRO-WARNING".

- **Authenticated access for individual NITRO operations:**NITRO allows you to logon to the Citrix ADC appliance to perform individual operations. You can use this option instead of creating a NITRO session (using the login object) and then using that session to perform all operations,

  To do this, you must specify the username and password in the request header of the NITRO request as follows:

  X-NITRO-USER:<username>

  X-NITRO-PASS:<password>

  Note: In such cases, make sure that the request header DOES not include the following:

  Cookie:NITRO_AUTH_TOKEN=<tokenvalue>

`*Note:`*

Mandatory parameters are marked in redand placeholder content is marked in <green>.

get

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/auditnslogpolicy_tmglobal_binding/name_value<String>

Query-parameters:

filter

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/auditnslogpolicy_tmglobal_binding/name_value<String>?**filter=property-name1:property-value1,property-name2:property-value2**

Use this query-parameter to get the filtered set of auditnslogpolicy_tmglobal_binding resources configured on Citrix ADC. Filtering can be done on any of the properties of the resource.

pagination

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/auditnslogpolicy_tmglobal_binding/name_value<String>?**pagesize=#no;pageno=#no**

Use this query-parameter to get the auditnslogpolicy_tmglobal_binding resources in chunks.

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "auditnslogpolicy_tmglobal_binding": [ {

"name":<String_value>,

"boundto":<String_value>,

"priority":<Double_value>,

"activepolicy":<Double_value>

}]}
```

get (all)

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/auditnslogpolicy_tmglobal_binding

Query-parameters:

bulkbindings

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/auditnslogpolicy_tmglobal_binding?**bulkbindings=yes**

NITRO allows you to fetch bindings in bulk.

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "auditnslogpolicy_tmglobal_binding": [ {

"name":<String_value>,

"boundto":<String_value>,

"priority":<Double_value>,

"activepolicy":<Double_value>

}]}
```

count

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/auditnslogpolicy_tmglobal_binding/name_value<String>?**count=yes**

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{"auditnslogpolicy_tmglobal_binding": [ { "__count": "#no"} ] }
```

auditnslogpolicy_vpnglobal_binding

Binding object showing the vpnglobal that can be bound to auditnslogpolicy.

Properties

(click to see Operations)

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
name	<String>	Read-write	Name of the policy. Minimum length = 1
boundto	<String>	Read-write	The entity name to which policy is bound.
priority	<Double>	Read-only	.

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
activepolicy	<Double>	Read-only	.
__count	<Double>	Read-only	count parameter

Operations

(click to see Properties)

GET| GET (ALL)| COUNT

Some options that you can use for each operations:

- **Getting warnings in response:**NITRO allows you to get warnings in an operation by specifying the "warning" query parameter as "yes". For example, to get warnings while connecting to the Citrix ADC appliance, the URL is as follows:

  http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/login?warning=yes

  If any, the warnings are displayed in the response payload with the HTTP code "209 X-NITRO-WARNING".
- **Authenticated access for individual NITRO operations:**NITRO allows you to logon to the Citrix ADC appliance to perform individual operations. You can use this option instead of creating a NITRO session (using the login object) and then using that session to perform all operations,

  To do this, you must specify the username and password in the request header of the NITRO request as follows:

  X-NITRO-USER:<username>

  X-NITRO-PASS:<password>

  Note: In such cases, make sure that the request header DOES not include the following:

  Cookie:NITRO_AUTH_TOKEN=<tokenvalue>

`*Note:`*

Mandatory parameters are marked in redand placeholder content is marked in <green>.

get

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/auditnslogpolicy_vpnglobal_binding/name_value<String>

Query-parameters:

filter

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/auditnslogpolicy_vpnglobal_binding/name_value<String>?**filter=property-name1:property-value1,property-name2:property-value2**

Use this query-parameter to get the filtered set of auditnslogpolicy_vpnglobal_binding resources configured on Citrix ADC. Filtering can be done on any of the properties of the resource.

pagination

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/auditnslogpolicy_vpnglobal_binding/name_value<String>?**pagesize=#no;pageno=#no**

Use this query-parameter to get the auditnslogpolicy_vpnglobal_binding resources in chunks.

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "auditnslogpolicy_vpnglobal_binding": [ {

"name":<String_value>,

"boundto":<String_value>,
```

```
"priority":<Double_value>,

"activepolicy":<Double_value>

}]}
```

get (all)

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/auditnslogpolicy_vpnglobal_binding

Query-parameters:

bulkbindings

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/auditnslogpolicy_vpnglobal_binding?**bulkbindings=yes**

NITRO allows you to fetch bindings in bulk.

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "auditnslogpolicy_vpnglobal_binding": [ {

"name":<String_value>,

"boundto":<String_value>,
```

```
"priority":<Double_value>,

"activepolicy":<Double_value>

}]}
```

count

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/auditnslogpolicy_vpnglobal_binding/name_value<String>?**count=yes**

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{"auditnslogpolicy_vpnglobal_binding": [ { "__count": "#no"} ] }
```

auditnslogpolicy_vpnvserver_binding

Binding object showing the vpnvserver that can be bound to auditnslogpolicy.

Properties

(click to see Operations)

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
name	<String>	Read-write	Name of the policy. Minimum length = 1
boundto	<String>	Read-write	The entity name to which policy is bound.
priority	<Double>	Read-only	.
activepolicy	<Double>	Read-only	.
__count	<Double>	Read-only	count parameter

Operations

(click to see Properties)

GET| GET (ALL)| COUNT

Some options that you can use for each operations:

- **Getting warnings in response:**NITRO allows you to get warnings in an operation by specifying the "warning" query parameter as "yes". For example, to get warnings while connecting to the Citrix ADC appliance, the URL is as follows:

  http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/login?warning=yes

  If any, the warnings are displayed in the response payload with the HTTP code "209 X-NITRO-WARNING".

- **Authenticated access for individual NITRO operations:**NITRO allows you to logon to the Citrix ADC appliance to perform individual operations. You can use this option instead of creating a NITRO session (using the login object) and then using that session to perform all operations,

  To do this, you must specify the username and password in the request header of the NITRO request as follows:

  X-NITRO-USER:<username>

  X-NITRO-PASS:<password>

Note: In such cases, make sure that the request header DOES not include the following:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue>

`*Note:`*

Mandatory parameters are marked in redand placeholder content is marked in <green>.

get

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/auditnslogpolicy_vpnvserver_binding/name_value<String>

Query-parameters:

filter

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/auditnslogpolicy_vpnvserver_binding/name_value<String>?**filter=property-name1:property-value1,property-name2:property-value2**

Use this query-parameter to get the filtered set of auditnslogpolicy_vpnvserver_binding resources configured on Citrix ADC. Filtering can be done on any of the properties of the resource.

pagination

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/auditnslogpolicy_vpnvserver_binding/name_value<String>?**pagesize=#no;pageno=#no**

Use this query-parameter to get the auditnslogpolicy_vpnvserver_binding resources in chunks.

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "auditnslogpolicy_vpnvserver_binding": [ {

"name":<String_value>,

"boundto":<String_value>,

"priority":<Double_value>,

"activepolicy":<Double_value>

}]}
```

get (all)

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/auditnslogpolicy_vpnvserver_binding

Query-parameters:

bulkbindings

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/auditnslogpolicy_vpnvserver_binding?**bulkbindings=yes**

NITRO allows you to fetch bindings in bulk.

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "auditnslogpolicy_vpnvserver_binding": [ {

"name":<String_value>,

"boundto":<String_value>,

"priority":<Double_value>,

"activepolicy":<Double_value>

}]}
```

count

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/auditnslogpolicy_vpnvserver_binding/name_value<String>?**count=yes**

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{"auditnslogpolicy_vpnvserver_binding": [ { "__count": "#no"} ] }
```

auditsyslogaction

Configuration for system log action resource.

Properties

(click to see Operations)

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
name	<String>	Read-write	Name of the syslog action. Must begin with a letter, number, or the underscore character (_), and must contain only letters, numbers, and the hyphen (-), period (.) pound (#), space ( ), at (@), equals (=), colon (:), and underscore characters. Cannot be changed after the syslog action is added. The following requirement applies only to the Citrix ADC CLI: If the name includes one or more spaces, enclose the name in double or single quotation marks (for example, "my syslog action" or 'my syslog action'). Minimum length = 1

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
serverip	<String>	Read-write	IP address of the syslog server. Minimum length = 1
serverdomainname	<String>	Read-write	SYSLOG server name as a FQDN. Mutually exclusive with serverIP/ lbVserverName. Minimum length = 1 Maximum length = 255
domainresolveretry	<Integer>	Read-write	Time, in seconds, for which the Citrix ADC waits before sending another DNS query to resolve the host name of the syslog server if the last query failed. Default value: 5 Minimum value = 5 Maximum value = 20939
lbvservername	<String>	Read-write	Name of the LB vserver. Mutually exclusive with syslog serverIP/ serverName. Minimum length = 1 Maximum length = 127

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
serverport	<Integer>	Read-write	Port on which the syslog server accepts connections. Minimum value = 1
loglevel	<String[]>	Read-write	Audit log level, which specifies the types of events to log. Available values function as follows: * ALL - All events. * EMERGENCY - Events that indicate an immediate crisis on the server. * ALERT - Events that might require action. * CRITICAL - Events that indicate an imminent server crisis. * ERROR - Events that indicate some type of error. * WARNING - Events that require action in the near future. * NOTICE - Events that the

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
			administrator should know about. * INFORMATIONAL - All but low-level events. * DEBUG - All events, in extreme detail. * NONE - No events. Possible values = ALL, EMERGENCY, ALERT, CRITICAL, ERROR, WARNING, NOTICE, INFORMATIONAL, DEBUG, NONE
dateformat	<String>	Read-write	Format of dates in the logs. Supported formats are: * MMDDYYYY. -U.S. style month/date/year format. * DDMMYYYY - European style date/month/year format.

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
			* YYYYMMDD - ISO style year/ month/date format. Possible values = MMDDYYYY, DDMMYYYY, YYYYMMDD
logfacility	<String>	Read-write	Facility value, as defined in RFC 3164, assigned to the log message. Log facility values are numbers 0 to 7 (LOCAL0 through LOCAL7). Each number indicates where a specific message originated from, such as the Citrix ADC itself, the VPN, or external. Possible values = LOCAL0, LOCAL1, LOCAL2, LOCAL3, LOCAL4, LOCAL5, LOCAL6, LOCAL7
tcp	<String>	Read-write	Log TCP messages. Possible values = NONE, ALL
acl	<String>	Read-write	Log access control list (ACL) messages.

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
			Possible values = ENABLED, DISABLED
timezone	<String>	Read-write	Time zone used for date and timestamps in the logs. Supported settings are: * GMT_TIME. Coordinated Universal time. * LOCAL_TIME. Use the server's timezone setting. Possible values = GMT_TIME, LOCAL_TIME
userdefinedauditlog	<String>	Read-write	Log user-configurable log messages to syslog. Setting this parameter to NO causes auditing to ignore all user-configured message actions. Setting this parameter to YES causes auditing to log user-configured message actions that meet the other logging criteria.

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
			Possible values = YES, NO
appflowexport	<String>	Read-write	Export log messages to AppFlow collectors. Appflow collectors are entities to which log messages can be sent so that some action can be performed on them. Possible values = ENABLED, DISABLED
lsn	<String>	Read-write	Log lsn info. Possible values = ENABLED, DISABLED
alg	<String>	Read-write	Log alg info. Possible values = ENABLED, DISABLED
subscriberlog	<String>	Read-write	Log subscriber session event information. Possible values = ENABLED, DISABLED
transport	<String>	Read-write	Transport type used to send auditlogs to syslog server. Default type is UDP.

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
			Possible values = TCP, UDP
tcpprofilename	<String>	Read-write	Name of the TCP profile whose settings are to be applied to the audit server info to tune the TCP connection parameters. Minimum length = 1 Maximum length = 127
maxlogdatasizetohold	<Double>	Read-write	Max size of log data that can be held in NSB chain of server info. Default value: 500 Minimum value = 50 Maximum value = 25600
dns	<String>	Read-write	Log DNS related syslog messages. Possible values = ENABLED, DISABLED
contentinspectionlog	<String>	Read-write	Log Content Inspection event information. Possible values = ENABLED, DISABLED

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
netprofile	<String>	Read-write	Name of the network profile. The SNIP configured in the network profile will be used as source IP while sending log messages. Minimum length = 1 Maximum length = 127
sslinterception	<String>	Read-write	Log SSL Interception event information. Possible values = ENABLED, DISABLED
urlfiltering	<String>	Read-write	Log URL filtering event information. Possible values = ENABLED, DISABLED
domainresolvenow	<Boolean>	Read-write	Immediately send a DNS query to resolve the server's domain name.
ip	<String>	Read-only	The resolved IP address of the syslog server.
__count	<Double>	Read-only	count parameter

Operations

(click to see Properties)

ADD| DELETE| UPDATE| UNSET| GET (ALL)| GET| COUNT

Some options that you can use for each operations:

- **Getting warnings in response:**NITRO allows you to get warnings in an operation by specifying the "warning" query parameter as "yes". For example, to get warnings while connecting to the Citrix ADC appliance, the URL is as follows:

  http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/login?warning=yes

  If any, the warnings are displayed in the response payload with the HTTP code "209 X-NITRO-WARNING".

- **Authenticated access for individual NITRO operations:**NITRO allows you to logon to the Citrix ADC appliance to perform individual operations. You can use this option instead of creating a NITRO session (using the login object) and then using that session to perform all operations,

  To do this, you must specify the username and password in the request header of the NITRO request as follows:

  X-NITRO-USER:<username>

  X-NITRO-PASS:<password>

  Note: In such cases, make sure that the request header DOES not include the following:

  Cookie:NITRO_AUTH_TOKEN=<tokenvalue>

`*Note:`*

Mandatory parameters are marked in redand placeholder content is marked in <green>.

add

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/auditsyslogaction

HTTP Method:POST

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Content-Type:application/json

Request Payload:

```
{"auditsyslogaction":{

<b>"name":<String_value>,

</b>"serverip":<String_value>,

"serverdomainname":<String_value>,

"domainresolveretry":<Integer_value>,

"lbvservername":<String_value>,

"serverport":<Integer_value>,

<b>"loglevel":<String[]_value>,

</b>"dateformat":<String_value>,

"logfacility":<String_value>,

"tcp":<String_value>,

"acl":<String_value>,

"timezone":<String_value>,

"userdefinedauditlog":<String_value>,

"appflowexport":<String_value>,

"lsn":<String_value>,

"alg":<String_value>,

"subscriberlog":<String_value>,

"transport":<String_value>,

"tcpprofilename":<String_value>,

"maxlogdatasizetohold":<Double_value>,
```

```
"dns":<String_value>,

"contentinspectionlog":<String_value>,

"netprofile":<String_value>,

"sslinterception":<String_value>,

"urlfiltering":<String_value>

}}
```

Response:

HTTP Status Code on Success: 201 Created HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error

delete

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/auditsyslogaction/name_value<String>

HTTP Method:DELETE

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue>

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error

update

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/auditsyslogaction

HTTP Method:PUT

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Content-Type:application/json

Request Payload:

```
{"auditsyslogaction":{

<b>"name":<String_value>,

</b>"serverip":<String_value>,

"serverdomainname":<String_value>,

"lbvservername":<String_value>,

"domainresolveretry":<Integer_value>,

"domainresolvenow":<Boolean_value>,

"serverport":<Integer_value>,

"loglevel":<String[]_value>,

"dateformat":<String_value>,

"logfacility":<String_value>,

"tcp":<String_value>,

"acl":<String_value>,

"timezone":<String_value>,

"userdefinedauditlog":<String_value>,

"appflowexport":<String_value>,

"lsn":<String_value>,

"alg":<String_value>,

"subscriberlog":<String_value>,

"tcpprofilename":<String_value>,
```

```
"maxlogdatasizetohold":<Double_value>,

"dns":<String_value>,

"contentinspectionlog":<String_value>,

"netprofile":<String_value>,

"sslinterception":<String_value>,

"urlfiltering":<String_value>

}}
```

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error

unset

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/auditsyslogaction?**action=unset**

HTTP Method:POST

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Content-Type:application/json

Request Payload:

```
{"auditsyslogaction":{

<b>"name":<String_value>,

</b>"serverport":true,

"loglevel":true,
```

```
"dateformat":true,

"logfacility":true,

"tcp":true,

"acl":true,

"timezone":true,

"userdefinedauditlog":true,

"appflowexport":true,

"lsn":true,

"alg":true,

"subscriberlog":true,

"tcpprofilename":true,

"maxlogdatasizetohold":true,

"dns":true,

"contentinspectionlog":true,

"netprofile":true,

"sslinterception":true,

"urlfiltering":true

}}
```


Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error

get (all)

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/auditsyslogaction

Query-parameters:

attrs

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/auditsyslogaction?**attrs=property-name1,property-name2**

Use this query parameter to specify the resource details that you want to retrieve.

filter

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/auditsyslogaction?**filter=property-name1:property-val1,property-name2:property-val2**

Use this query-parameter to get the filtered set of auditsyslogaction resources configured on Citrix ADC. Filtering can be done on any of the properties of the resource.

view

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/auditsyslogaction?**view=summary**

Note: By default, the retrieved results are displayed in detail view (?view=detail).

pagination

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/auditsyslogaction?**pagesize=#no;pageno=#no**

Use this query-parameter to get the auditsyslogaction resources in chunks.

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "auditsyslogaction": [ {

"name":<String_value>,

"serverip":<String_value>,

"serverdomainname":<String_value>,

"ip":<String_value>,

"lbvservername":<String_value>,

"domainresolveretry":<Integer_value>,

"domainresolvenow":<Boolean_value>,

"serverport":<Integer_value>,

"loglevel":<String[]_value>,

"dateformat":<String_value>,

"logfacility":<String_value>,

"tcp":<String_value>,

"acl":<String_value>,

"timezone":<String_value>,

"userdefinedauditlog":<String_value>,

"appflowexport":<String_value>,

"lsn":<String_value>,

"alg":<String_value>,
```

```
"subscriberlog":<String_value>,

"transport":<String_value>,

"tcpprofilename":<String_value>,

"maxlogdatasizetohold":<Double_value>,

"dns":<String_value>,

"netprofile":<String_value>,

"sslinterception":<String_value>,

"urlfiltering":<String_value>,

"contentinspectionlog":<String_value>

}]}
```

get

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/auditsyslogaction/name_value<String>

Query-parameters:

attrs

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/auditsyslogaction/name_value<String>?**attrs=property-name1,property-name2**

Use this query parameter to specify the resource details that you want to retrieve.

view

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/auditsyslogaction/name_value<String>?**view=summary**

Note: By default, the retrieved results are displayed in detail view (?view=detail).

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "auditsyslogaction": [ {

"name":<String_value>,

"serverip":<String_value>,

"serverdomainname":<String_value>,

"ip":<String_value>,

"lbvservername":<String_value>,

"domainresolveretry":<Integer_value>,

"domainresolvenow":<Boolean_value>,

"serverport":<Integer_value>,

"loglevel":<String[]_value>,

"dateformat":<String_value>,

"logfacility":<String_value>,

"tcp":<String_value>,

"acl":<String_value>,

"timezone":<String_value>,
```

```
"userdefinedauditlog":<String_value>,

"appflowexport":<String_value>,

"lsn":<String_value>,

"alg":<String_value>,

"subscriberlog":<String_value>,

"transport":<String_value>,

"tcpprofilename":<String_value>,

"maxlogdatasizetohold":<Double_value>,

"dns":<String_value>,

"netprofile":<String_value>,

"sslinterception":<String_value>,

"urlfiltering":<String_value>,

"contentinspectionlog":<String_value>

}]}
```

count

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/auditsyslogaction?**count=yes**

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "auditsyslogaction": [ { "__count": "#no"} ] }
```

auditsyslogglobal_auditsyslogpolicy_binding

Binding object showing the auditsyslogpolicy that can be bound to auditsyslogglobal.

Properties

(click to see Operations)

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
priority	<Double>	Read-write	Specifies the priority of the policy. Minimum value = 1 Maximum value = 2147483647
globalbindtype	<String>	Read-write	. Default value: SYSTEM_GLOBAL Possible values = SYSTEM_GLOBAL, VPN_GLOBAL, RNAT_GLOBAL
policyname	<String>	Read-write	Name of the audit syslog policy.

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
numpol	<Double>	Read-only	number of polices bound to label.
__count	<Double>	Read-only	count parameter

Operations

(click to see Properties)

ADD:| DELETE:| GET| GET (ALL)| COUNT

Some options that you can use for each operations:

- **Getting warnings in response:**NITRO allows you to get warnings in an operation by specifying the "warning" query parameter as "yes". For example, to get warnings while connecting to the Citrix ADC appliance, the URL is as follows:

  http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/login?warning=yes

  If any, the warnings are displayed in the response payload with the HTTP code "209 X-NITRO-WARNING".

- **Authenticated access for individual NITRO operations:**NITRO allows you to logon to the Citrix ADC appliance to perform individual operations. You can use this option instead of creating a NITRO session (using the login object) and then using that session to perform all operations,

  To do this, you must specify the username and password in the request header of the NITRO request as follows:

  X-NITRO-USER:<username>

  X-NITRO-PASS:<password>

  Note: In such cases, make sure that the request header DOES not include the following:

  Cookie:NITRO_AUTH_TOKEN=<tokenvalue>

`*Note:`*

Mandatory parameters are marked in redand placeholder content is marked in <green>.

add

URL:http://<Citrix-ADC-IP-address(NSIP)/nitro/v1/config/auditsyslogglobal_auditsyslogpolicy_binding

HTTP Method:PUT

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Content-Type:application/json

Request Payload:

```
{

"auditsyslogglobal_auditsyslogpolicy_binding":{

<b>"policyname":<String_value>,

</b><b>"priority":<Double_value>,

</b>"globalbindtype":<String_value>

}}
```

Response:

HTTP Status Code on Success: 201 Created HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error

delete

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/auditsyslogglobal_auditsyslogpolicy_binding

Query-parameters:

args

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/auditsyslogglobal_auditsyslogpolicy_binding?**args=policyname:<String_value>,globalbindtype:<String_value>**

HTTP Method:DELETE

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue>

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error

get

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/auditsyslogglobal_auditsyslogpolicy_binding

Query-parameters:

filter

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/auditsyslogglobal_auditsyslogpolicy_binding?**filter=property-name1:property-value1,property-name2:property-value2**

Use this query-parameter to get the filtered set of auditsyslogglobal_auditsyslogpolicy_binding resources configured on Citrix ADC. Filtering can be done on any of the properties of the resource.

pagination

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/auditsyslogglobal_auditsyslogpolicy_binding?**pagesize=#no;pageno=#no**

Use this query-parameter to get the auditsyslogglobal_auditsyslogpolicy_binding resources in chunks.

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "auditsyslogglobal_auditsyslogpolicy_binding": [ {

"priority":<Double_value>,

"globalbindtype":<String_value>,

"policyname":<String_value>,

"numpol":<Double_value>

}]}
```

get (all)

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/auditsyslogglobal_auditsyslogpolicy_binding

Query-parameters:

bulkbindings

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/auditsyslogglobal_auditsyslogpolicy_binding?**bulkbindings=yes**

NITRO allows you to fetch bindings in bulk.

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "auditsyslogglobal_auditsyslogpolicy_binding": [ {

"priority":<Double_value>,

"globalbindtype":<String_value>,

"policyname":<String_value>,

"numpol":<Double_value>

}]}
```

count

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/auditsyslogglobal_auditsyslogpolicy_binding?**count=yes**

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "auditsyslogglobal_auditsyslogpolicy_binding": [ { "__count": "#no"} ] }
```

auditsyslogglobal_binding

Binding object which returns the resources bound to auditsyslogglobal.

Properties

(click to see Operations)

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
auditsyslogglobal_auditsyslogpolicy_binding	<auditsyslogglobal_auditsyslogpolicy_binding[]>	Read-only	auditsyslogpolicy that can be bound to auditsyslogglobal.

Operations

(click to see Properties)

GET| GET (ALL)

Some options that you can use for each operations:

- **Getting warnings in response:**NITRO allows you to get warnings in an operation by specifying the "warning" query parameter as "yes". For example, to get warnings while connecting to the Citrix ADC appliance, the URL is as follows:

  http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/login?warning=yes

  If any, the warnings are displayed in the response payload with the HTTP code "209 X-NITRO-WARNING".

- **Authenticated access for individual NITRO operations:**NITRO allows you to logon to the Citrix ADC appliance to perform individual operations. You can use this option instead of creating a NITRO session (using the login object) and then using that session to perform all operations,

  To do this, you must specify the username and password in the request header of the NITRO request as follows:

X-NITRO-USER:<username>

X-NITRO-PASS:<password>

Note: In such cases, make sure that the request header DOES not include the following:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue>

`*Note:`*

Mandatory parameters are marked in redand placeholder content is marked in <green>.

get

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/auditsyslogglobal_binding

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "auditsyslogglobal_binding": [ {

"auditsyslogglobal_auditsyslogpolicy_binding":<auditsyslogglobal_auditsyslogpolicy_binding[]_value>

}]}
```

get (all)

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/auditsyslogglobal_binding

Query-parameters:

bulkbindings

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/auditsyslogglobal_binding?**bulkbindings=yes**

NITRO allows you to fetch bindings in bulk.

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "auditsyslogglobal_binding": [ {

"auditsyslogglobal_auditsyslogpolicy_binding":<auditsyslogglobal_auditsyslogpolicy_binding[]_value>

}]}
```

auditsyslogparams

Configuration for system log parameters resource.

Properties

(click to see Operations)

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
serverip	<String>	Read-write	IP address of the syslog server. Minimum length = 1
serverport	<Integer>	Read-write	Port on which the syslog server accepts connections. Minimum value = 1
dateformat	<String>	Read-write	Format of dates in the logs. Supported formats are: * MMDDYYYY - U.S. style month/ date/year format. * DDMMYYYY. European style -date/month/year format. * YYYYMMDD - ISO style year/ month/date format. Possible values = MMDDYYYY, DDMMYYYY, YYYYMMDD
loglevel	<String[]>	Read-write	Types of information to be logged. Available settings function as follows: * ALL - All events.

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
			* EMERGENCY - Events that indicate an immediate crisis on the server. * ALERT - Events that might require action. * CRITICAL - Events that indicate an imminent server crisis. * ERROR - Events that indicate some type of error. * WARNING - Events that require action in the near future. * NOTICE - Events that the administrator should know about. * INFORMATIONAL - All but low-level events. * DEBUG - All events, in extreme detail. * NONE - No events.

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
			Possible values = ALL, EMERGENCY, ALERT, CRITICAL, ERROR, WARNING, NOTICE, INFORMATIONAL, DEBUG, NONE
logfacility	<String>	Read-write	Facility value, as defined in RFC 3164, assigned to the log message. Log facility values are numbers 0 to 7 (LOCAL0 through LOCAL7). Each number indicates where a specific message originated from, such as the Citrix ADC itself, the VPN, or external. Possible values = LOCAL0, LOCAL1, LOCAL2, LOCAL3, LOCAL4, LOCAL5, LOCAL6, LOCAL7
tcp	<String>	Read-write	Log TCP messages. Possible values = NONE, ALL
acl	<String>	Read-write	Log access control list (ACL) messages.

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
			Possible values = ENABLED, DISABLED
timezone	<String>	Read-write	Time zone used for date and timestamps in the logs. Available settings function as follows: * GMT_TIME - Coordinated Universal Time. * LOCAL_TIME Use the server's timezone setting. Possible values = GMT_TIME, LOCAL_TIME
userdefinedauditlog	<String>	Read-write	Log user-configurable log messages to syslog. Setting this parameter to NO causes audit to ignore all user-configured message actions. Setting this parameter to YES causes audit to log user-configured message actions that meet the other logging criteria.

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
			Possible values = YES, NO
appflowexport	<String>	Read-write	Export log messages to AppFlow collectors. Appflow collectors are entities to which log messages can be sent so that some action can be performed on them. Possible values = ENABLED, DISABLED
lsn	<String>	Read-write	Log the LSN messages. Possible values = ENABLED, DISABLED
alg	<String>	Read-write	Log the ALG messages. Possible values = ENABLED, DISABLED
subscriberlog	<String>	Read-write	Log subscriber session event information. Possible values = ENABLED, DISABLED
dns	<String>	Read-write	Log DNS related syslog messages.

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
			Possible values = ENABLED, DISABLED
sslinterception	<String>	Read-write	Log SSL Interceptionn event information. Possible values = ENABLED, DISABLED
urlfiltering	<String>	Read-write	Log URL filtering event information. Possible values = ENABLED, DISABLED
contentinspectionlog	<String>	Read-write	Log Content Inspection event ifnormation. Possible values = ENABLED, DISABLED

Operations

(click to see Properties)

UPDATE| UNSET| GET (ALL)

Some options that you can use for each operations:

- **Getting warnings in response:**NITRO allows you to get warnings in an operation by specifying the "warning" query parameter as "yes". For example, to get warnings while connecting to the Citrix ADC appliance, the URL is as follows:

  http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/login?warning=yes

  If any, the warnings are displayed in the response payload with the HTTP code "209 X-NITRO-WARNING".

- **Authenticated access for individual NITRO operations:**NITRO allows you to logon to the Citrix ADC appliance to perform individual operations. You can use this option instead of creating a NITRO session (using the login object) and then using that session to perform all operations,

  To do this, you must specify the username and password in the request header of the NITRO request as follows:

  X-NITRO-USER:<username>

  X-NITRO-PASS:<password>

  Note: In such cases, make sure that the request header DOES not include the following:

  Cookie:NITRO_AUTH_TOKEN=<tokenvalue>

`*Note:`*

Mandatory parameters are marked in redand placeholder content is marked in <green>.

update

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/auditsyslogparams

HTTP Method:PUT

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Content-Type:application/json

Request Payload:

```
{"auditsyslogparams":{

"serverip":<String_value>,

"serverport":<Integer_value>,

"dateformat":<String_value>,

"loglevel":<String[]_value>,

"logfacility":<String_value>,
```

```
"tcp":<String_value>,

"acl":<String_value>,

"timezone":<String_value>,

"userdefinedauditlog":<String_value>,

"appflowexport":<String_value>,

"lsn":<String_value>,

"alg":<String_value>,

"subscriberlog":<String_value>,

"dns":<String_value>,

"sslinterception":<String_value>,

"urlfiltering":<String_value>,

"contentinspectionlog":<String_value>

}}
```

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error

unset

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/auditsyslogparams?**action=unset**

HTTP Method:POST

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Content-Type:application/json

Request Payload:

```
{"auditsyslogparams":{
"serverip":true,
"serverport":true,
"loglevel":true,
"dateformat":true,
"logfacility":true,
"tcp":true,
"acl":true,
"timezone":true,
"userdefinedauditlog":true,
"appflowexport":true,
"lsn":true,
"alg":true,
"subscriberlog":true,
"dns":true,
"contentinspectionlog":true,
"sslinterception":true,
"urlfiltering":true
}}
```

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error

get (all)

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/auditsyslogparams

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "auditsyslogparams": [ {

"name":<String_value>,

"serverip":<String_value>,

"serverport":<Integer_value>,

"dateformat":<String_value>,

"loglevel":<String[]_value>,

"logfacility":<String_value>,

"tcp":<String_value>,

"acl":<String_value>,
```

```
"timezone":<String_value>,

"userdefinedauditlog":<String_value>,

"appflowexport":<String_value>,

"lsn":<String_value>,

"alg":<String_value>,

"subscriberlog":<String_value>,

"dns":<String_value>,

"sslinterception":<String_value>,

"urlfiltering":<String_value>,

"contentinspectionlog":<String_value>

}]}
```

auditsyslogpolicy

Configuration for system log policy resource.

Properties

(click to see Operations)

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
name	<String>	Read-write	Name for the policy. Must begin with a letter, number, or the underscore character (_), and must consist only of letters, numbers, and the hyphen (-),

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
			period (.) pound (#), space ( ), at sign (@), equals (=), colon (:), and underscore characters. Cannot be changed after the syslog policy is added. The following requirement applies only to the Citrix ADC CLI: If the name includes one or more spaces, enclose the name in double or single quotation marks (for example, "my syslog policy" or 'my syslog policy'). Minimum length = 1
rule	<String>	Read-write	Name of the Citrix ADC named rule, or an expression, that defines the messages to be logged to the syslog server. Minimum length = 1
action	<String>	Read-write	Syslog server action to perform when this

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
			policy matches traffic. NOTE: A syslog server action must be associated with a syslog audit policy. Minimum length = 1
expressiontype	<String>	Read-only	Type of policy (Classic/Advanced). Possible values = Classic Policy, Advanced Policy
__count	<Double>	Read-only	count parameter

Operations

(click to see Properties)

ADD| DELETE| UPDATE| GET (ALL)| GET| COUNT

Some options that you can use for each operations:

- **Getting warnings in response:**NITRO allows you to get warnings in an operation by specifying the "warning" query parameter as "yes". For example, to get warnings while connecting to the Citrix ADC appliance, the URL is as follows:

  http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/login?warning=yes

  If any, the warnings are displayed in the response payload with the HTTP code "209 X-NITRO-WARNING".
- **Authenticated access for individual NITRO operations:**NITRO allows you to logon to the Citrix ADC appliance to perform individual operations. You can use this option instead of creating a NITRO session (using the login object) and then using that session to perform all operations,

To do this, you must specify the username and password in the request header of the NITRO request as follows:

X-NITRO-USER:<username>

X-NITRO-PASS:<password>

Note: In such cases, make sure that the request header DOES not include the following:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue>

`*Note:`*

Mandatory parameters are marked in redand placeholder content is marked in <green>.

add

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/auditsyslogpolicy

HTTP Method:POST

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Content-Type:application/json

Request Payload:

```
{"auditsyslogpolicy":{

<b>"name":<String_value>,

</b><b>"rule":<String_value>,

</b><b>"action":<String_value>

</b>}}
```

Response:

HTTP Status Code on Success: 201 Created HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error

delete

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/auditsyslogpolicy/name_value<String>

HTTP Method:DELETE

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue>

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error

update

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/auditsyslogpolicy

HTTP Method:PUT

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Content-Type:application/json

Request Payload:

```
{"auditsyslogpolicy":{

<b>"name":<String_value>,

</b>"rule":<String_value>,

"action":<String_value>

}}
```

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error

get (all)

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/auditsyslogpolicy

Query-parameters:

attrs

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/auditsyslogpolicy?**attrs=property-name1,property-name2**

Use this query parameter to specify the resource details that you want to retrieve.

filter

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/auditsyslogpolicy?**filter=property-name1:property-val1,property-name2:property-val2**

Use this query-parameter to get the filtered set of auditsyslogpolicy resources configured on Citrix ADC. Filtering can be done on any of the properties of the resource.

view

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/auditsyslogpolicy?**view=summary**

Note: By default, the retrieved results are displayed in detail view (?view=detail).

pagination

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/auditsyslogpolicy?**pagesize=#no;pageno=#no**

Use this query-parameter to get the auditsyslogpolicy resources in chunks.

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "auditsyslogpolicy": [ {

"name":<String_value>,

"rule":<String_value>,

"action":<String_value>,

"expressiontype":<String_value>

}]}
```

get

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/auditsyslogpolicy/name_value<String>

Query-parameters:

attrs

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/auditsyslogpolicy/name_value<String>?**attrs=property-name1,property-name2**

Use this query parameter to specify the resource details that you want to retrieve.

view

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/auditsyslogpolicy/name_value<String>?**view=summary**

Note: By default, the retrieved results are displayed in detail view (?view=detail).

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "auditsyslogpolicy": [ {

"name":<String_value>,

"rule":<String_value>,

"action":<String_value>,

"expressiontype":<String_value>

}]}
```

count

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/auditsyslogpolicy?**count=yes**

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "auditsyslogpolicy": [ { "__count": "#no"} ] }
```

auditsyslogpolicy_aaagroup_binding

Binding object showing the aaagroup that can be bound to auditsyslogpolicy.

Properties

(click to see Operations)

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
boundto	<String>	Read-write	The entity name to which policy is bound.
name	<String>	Read-write	Name of the policy. Minimum length = 1
priority	<Double>	Read-only	.
activepolicy	<Double>	Read-only	.
__count	<Double>	Read-only	count parameter

Operations

(click to see Properties)

GET| GET (ALL)| COUNT

Some options that you can use for each operations:

- **Getting warnings in response:**NITRO allows you to get warnings in an operation by specifying the "warning" query parameter as "yes". For example, to get warnings while connecting to the Citrix ADC appliance, the URL is as follows:

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/login?warning=yes

If any, the warnings are displayed in the response payload with the HTTP code "209 X-NITRO-WARNING".

- **Authenticated access for individual NITRO operations:**NITRO allows you to logon to the Citrix ADC appliance to perform individual operations. You can use this option instead of creating a NITRO session (using the login object) and then using that session to perform all operations,

  To do this, you must specify the username and password in the request header of the NITRO request as follows:

  X-NITRO-USER:<username>

  X-NITRO-PASS:<password>

  Note: In such cases, make sure that the request header DOES not include the following:

  Cookie:NITRO_AUTH_TOKEN=<tokenvalue>

`*Note:`*

Mandatory parameters are marked in redand placeholder content is marked in <green>.

get

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/auditsyslogpolicy_aaagroup_binding/name_value<String>

Query-parameters:

filter

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/auditsyslogpolicy_aaagroup_binding/name_value<String>?**filter=property-name1:property-value1,property-name2:property-value2**

Use this query-parameter to get the filtered set of auditsyslogpolicy_aaagroup_binding resources configured on Citrix ADC. Filtering can be done on any of the properties of the resource.

pagination

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/auditsyslogpolicy_aaagroup_binding/name_value<String>?**pagesize=#no;pageno=#no**

Use this query-parameter to get the auditsyslogpolicy_aaagroup_binding resources in chunks.

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "auditsyslogpolicy_aaagroup_binding": [ {

"boundto":<String_value>,

"name":<String_value>,

"priority":<Double_value>,

"activepolicy":<Double_value>

}]}
```

get (all)

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/auditsyslogpolicy_aaagroup_binding

Query-parameters:

bulkbindings

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/auditsyslogpolicy_aaagroup_binding?**bulkbindings=yes**

NITRO allows you to fetch bindings in bulk.

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "auditsyslogpolicy_aaagroup_binding": [ {

"boundto":<String_value>,

"name":<String_value>,

"priority":<Double_value>,

"activepolicy":<Double_value>

}]}
```

count

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/auditsyslogpolicy_aaagroup_binding/name_value<String>?**count=yes**

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{"auditsyslogpolicy_aaagroup_binding": [ { "__count": "#no"} ] }
```

auditsyslogpolicy_aaauser_binding

Binding object showing the aaauser that can be bound to auditsyslogpolicy.

Properties

(click to see Operations)

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
boundto	<String>	Read-write	The entity name to which policy is bound.
name	<String>	Read-write	Name of the policy. Minimum length = 1
priority	<Double>	Read-only	.
activepolicy	<Double>	Read-only	.
__count	<Double>	Read-only	count parameter

Operations

(click to see Properties)

GET| GET (ALL)| COUNT

Some options that you can use for each operations:

- **Getting warnings in response:**NITRO allows you to get warnings in an operation by specifying the "warning" query parameter as "yes". For example, to get warnings while connecting to the Citrix ADC appliance, the URL is as follows:

  http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/login?warning=yes

  If any, the warnings are displayed in the response payload with the HTTP code "209 X-NITRO-WARNING".

- **Authenticated access for individual NITRO operations:**NITRO allows you to logon to the Citrix ADC appliance to perform individual operations. You can use this option instead of creating a NITRO session (using the login object) and then using that session to perform all operations,

  To do this, you must specify the username and password in the request header of the NITRO request as follows:

  X-NITRO-USER:<username>

  X-NITRO-PASS:<password>

  Note: In such cases, make sure that the request header DOES not include the following:

  Cookie:NITRO_AUTH_TOKEN=<tokenvalue>

`*Note:`*

Mandatory parameters are marked in redand placeholder content is marked in <green>.

get

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/auditsyslogpolicy_aaauser_binding/name_value<String>

Query-parameters:

filter

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/auditsyslogpolicy_aaauser_binding/name_value<String>?**filter=property-name1:property-value1,property-name2:property-value2**

Use this query-parameter to get the filtered set of auditsyslogpolicy_aaauser_binding resources configured on Citrix ADC. Filtering can be done on any of the properties of the resource.

pagination

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/auditsyslogpolicy_aaauser_binding/name_value<String>?**pagesize=#no;pageno=#no**

Use this query-parameter to get the auditsyslogpolicy_aaauser_binding resources in chunks.

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "auditsyslogpolicy_aaauser_binding": [ {

"boundto":<String_value>,

"name":<String_value>,

"priority":<Double_value>,

"activepolicy":<Double_value>

}]}
```

get (all)

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/auditsyslogpolicy_aaauser_binding

Query-parameters:

bulkbindings

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/auditsyslogpolicy_aaauser_binding?**bulkbindings=yes**

NITRO allows you to fetch bindings in bulk.

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "auditsyslogpolicy_aaauser_binding": [ {

"boundto":<String_value>,

"name":<String_value>,

"priority":<Double_value>,

"activepolicy":<Double_value>

}]}
```

count

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/auditsyslogpolicy_aaauser_binding/name_value<String>?**count=yes**

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{"auditsyslogpolicy_aaauser_binding": [ { "__count": "#no"} ] }
```

auditsyslogpolicy_auditsyslogglobal_binding

Binding object showing the auditsyslogglobal that can be bound to auditsyslogpolicy.

Properties

(click to see Operations)

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
boundto	<String>	Read-write	The entity name to which policy is bound.
name	<String>	Read-write	Name of the policy. Minimum length = 1
priority	<Double>	Read-only	.
activepolicy	<Double>	Read-only	.
__count	<Double>	Read-only	count parameter

Operations

(click to see Properties)

GET| GET (ALL)| COUNT

Some options that you can use for each operations:

- **Getting warnings in response:**NITRO allows you to get warnings in an operation by specifying the "warning" query parameter as "yes". For example, to get warnings while connecting to the Citrix ADC appliance, the URL is as follows:

  http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/login?warning=yes

  If any, the warnings are displayed in the response payload with the HTTP code "209 X-NITRO-WARNING".

- **Authenticated access for individual NITRO operations:**NITRO allows you to logon to the Citrix ADC appliance to perform individual operations. You can use this option instead of creating a NITRO session (using the login object) and then using that session to perform all operations,

  To do this, you must specify the username and password in the request header of the NITRO request as follows:

  X-NITRO-USER:<username>

  X-NITRO-PASS:<password>

  Note: In such cases, make sure that the request header DOES not include the following:

  Cookie:NITRO_AUTH_TOKEN=<tokenvalue>

`*Note:`*

Mandatory parameters are marked in redand placeholder content is marked in <green>.

get

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/auditsyslogpolicy_auditsyslogglobal_binding/name_value<String>

Query-parameters:

filter

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/auditsyslogpolicy_auditsyslogglobal_binding/name_value<String>?**filter=property-name1:property-value1,property-name2:property-value2**

Use this query-parameter to get the filtered set of auditsyslogpolicy_auditsyslogglobal_binding resources configured on Citrix ADC. Filtering can be done on any of the properties of the resource.

pagination

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/auditsyslogpolicy_auditsyslogglobal_binding/name_value<String>?**pagesize=#no;pageno=#no**

Use this query-parameter to get the auditsyslogpolicy_auditsyslogglobal_binding resources in chunks.

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "auditsyslogpolicy_auditsyslogglobal_binding": [ {

"boundto":<String_value>,

"name":<String_value>,

"priority":<Double_value>,

"activepolicy":<Double_value>

}]}
```

get (all)

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/auditsyslogpolicy_auditsyslogglobal_binding

Query-parameters:

bulkbindings

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/auditsyslogpolicy_auditsyslogglobal_binding?**bulkbindings=yes**

NITRO allows you to fetch bindings in bulk.

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "auditsyslogpolicy_auditsyslogglobal_binding": [ {

"boundto":<String_value>,

"name":<String_value>,

"priority":<Double_value>,

"activepolicy":<Double_value>

}]}
```

count

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/auditsyslogpolicy_auditsyslogglobal_binding/name_value<String>?**count=yes**

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{"auditsyslogpolicy_auditsyslogglobal_binding": [ { "__count": "#no"} ] }
```

auditsyslogpolicy_authenticationvserver_binding

Binding object showing the authenticationvserver that can be bound to auditsyslogpolicy.

Properties

(click to see Operations)

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
boundto	<String>	Read-write	The entity name to which policy is bound.
name	<String>	Read-write	Name of the policy. Minimum length = 1
priority	<Double>	Read-only	.

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
activepolicy	<Double>	Read-only	.
__count	<Double>	Read-only	count parameter

Operations

(click to see Properties)

GET| GET (ALL)| COUNT

Some options that you can use for each operations:

- **Getting warnings in response:**NITRO allows you to get warnings in an operation by specifying the "warning" query parameter as "yes". For example, to get warnings while connecting to the Citrix ADC appliance, the URL is as follows:

  http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/login?warning=yes

  If any, the warnings are displayed in the response payload with the HTTP code "209 X-NITRO-WARNING".
- **Authenticated access for individual NITRO operations:**NITRO allows you to logon to the Citrix ADC appliance to perform individual operations. You can use this option instead of creating a NITRO session (using the login object) and then using that session to perform all operations,

  To do this, you must specify the username and password in the request header of the NITRO request as follows:

  X-NITRO-USER:<username>

  X-NITRO-PASS:<password>

  Note: In such cases, make sure that the request header DOES not include the following:

  Cookie:NITRO_AUTH_TOKEN=<tokenvalue>

`*Note:`*

Mandatory parameters are marked in redand placeholder content is marked in <green>.

get

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/auditsyslogpolicy_authenticationvserver_binding/name_value<String>

Query-parameters:

filter

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/auditsyslogpolicy_authenticationvserver_binding/name_value<String>?**filter=property-name1:property-value1,property-name2:property-value2**

Use this query-parameter to get the filtered set of auditsyslogpolicy_authenticationvserver_binding resources configured on Citrix ADC. Filtering can be done on any of the properties of the resource.

pagination

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/auditsyslogpolicy_authenticationvserver_binding/name_value<String>?**pagesize=#no;pageno=#no**

Use this query-parameter to get the auditsyslogpolicy_authenticationvserver_binding resources in chunks.

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "auditsyslogpolicy_authenticationvserver_binding": [ {
```

```
"boundto":<String_value>,

"name":<String_value>,

"priority":<Double_value>,

"activepolicy":<Double_value>

}]}
```

get (all)

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/auditsyslogpolicy_authenticationvserver_binding

Query-parameters:

bulkbindings

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/auditsyslogpolicy_authenticationvserver_binding?**bulkbindings=yes**

NITRO allows you to fetch bindings in bulk.

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "auditsyslogpolicy_authenticationvserver_binding": [ {

"boundto":<String_value>,

"name":<String_value>,

"priority":<Double_value>,

"activepolicy":<Double_value>

}]}
```

count

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/auditsyslogpolicy_authenticationvserver_binding/name_value<String>?**count=yes**

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{"auditsyslogpolicy_authenticationvserver_binding": [ { "__count":
"#no"} ] }
```

auditsyslogpolicy_binding

Binding object which returns the resources bound to auditsyslogpolicy.

Properties

(click to see Operations)

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
name	<String>	Read-write	Name of the policy. Minimum length = 1
auditsyslogpolicy_rnatglobal_binding	<auditsyslogpolicy_rnatglobal_binding[]>	Read-only	rnatglobal that can be bound to auditsyslogpolicy.
auditsyslogpolicy_vpnvserver_binding	<auditsyslogpolicy_vpnvserver_binding[]>	Read-only	vpnvserver that can be bound to auditsyslogpolicy.
auditsyslogpolicy_csvserver_binding	<auditsyslogpolicy_csvserver_binding[]>	Read-only	csvserver that can be bound to auditsyslogpolicy.
auditsyslogpolicy_tmglobal_binding	<auditsyslogpolicy_tmglobal_binding[]>	Read-only	tmglobal that can be bound to auditsyslogpolicy.
auditsyslogpolicy_systemglobal_binding	<auditsyslogpolicy_systemglobal_binding[]>	Read-only	systemglobal that can be bound to auditsyslogpolicy.
auditsyslogpolicy_lbvserver_binding	<auditsyslogpolicy_lbvserver_binding[]>	Read-only	lbvserver that can be bound to auditsyslogpolicy.
auditsyslogpolicy_authenticationvserver_binding	<auditsyslogpolicy_authenticationvserver_binding[]>	Read-only	authenticationvserver that can be bound to auditsyslogpolicy.
auditsyslogpolicy_aaagroup_binding	<auditsyslogpolicy_aaagroup_binding[]>	Read-only	aaagroup that can be bound to auditsyslogpolicy.

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
auditsyslogpolicy_aaauser_binding	<auditsyslogpolicy_aaauser_binding[]>	Read-only	aaauser that can be bound to auditsyslogpolicy.
auditsyslogpolicy_auditsysloglobal_binding	<auditsyslogpolicy_auditsysloglobal_binding[]>	Read-only	auditsysloglobal that can be bound to auditsyslogpolicy.
auditsyslogpolicy_vpnglobal_binding	<auditsyslogpolicy_vpnglobal_binding[]>	Read-only	vpnglobal that can be bound to auditsyslogpolicy.

Operations

(click to see Properties)

GET| GET (ALL)

Some options that you can use for each operations:

- **Getting warnings in response:**NITRO allows you to get warnings in an operation by specifying the "warning" query parameter as "yes". For example, to get warnings while connecting to the Citrix ADC appliance, the URL is as follows:

  http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/login?warning=yes

  If any, the warnings are displayed in the response payload with the HTTP code "209 X-NITRO-WARNING".

- **Authenticated access for individual NITRO operations:**NITRO allows you to logon to the Citrix ADC appliance to perform individual operations. You can use this option instead of creating a NITRO session (using the login object) and then using that session to perform all operations,

  To do this, you must specify the username and password in the request header of the NITRO request as follows:

  X-NITRO-USER:<username>

  X-NITRO-PASS:<password>

  Note: In such cases, make sure that the request header DOES not include the following:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue>

`*Note:`*

Mandatory parameters are marked in redand placeholder content is marked in <green>.

get

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/auditsyslogpolicy_binding/name_value<String>

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "auditsyslogpolicy_binding": [ {

"name":<String_value>,

"auditsyslogpolicy_aaauser_binding":<auditsyslogpolicy_aaauser_binding[]_value>,

"auditsyslogpolicy_vpnglobal_binding":<auditsyslogpolicy_vpnglobal_binding[]_value>,

"auditsyslogpolicy_authenticationvserver_binding":<auditsyslogpolicy_authenticationvserver_binding[]_value>,

"auditsyslogpolicy_lbvserver_binding":<auditsyslogpolicy_lbvserver_binding[]_value>,
```

```
"auditsyslogpolicy_aaagroup_binding":<auditsyslogpolicy_aaagroup_binding[]
_value>,

"auditsyslogpolicy_auditsyslogglobal_binding":<auditsyslogpolicy_auditsysl
ogglobal_binding[]_value>,

"auditsyslogpolicy_tmglobal_binding":<auditsyslogpolicy_tmglobal_binding[]
_value>,

"auditsyslogpolicy_rnatglobal_binding":<auditsyslogpolicy_rnatglobal_bindi
ng[]_value>,

"auditsyslogpolicy_csvserver_binding":<auditsyslogpolicy_csvserver_binding
[]_value>,

"auditsyslogpolicy_systemglobal_binding":<auditsyslogpolicy_systemglobal_b
inding[]_value>,

"auditsyslogpolicy_vpnvserver_binding":<auditsyslogpolicy_vpnvserver_bindi
ng[]_value>

}]}
```

get (all)

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/auditsyslogpolicy_binding

Query-parameters:

bulkbindings

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/auditsyslogpolicy_binding?**bulkbindings=yes**

NITRO allows you to fetch bindings in bulk.

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "auditsyslogpolicy_binding": [ {

"name":<String_value>,

"auditsyslogpolicy_aaauser_binding":<auditsyslogpolicy_aaauser_binding[]_v
alue>,

"auditsyslogpolicy_vpnglobal_binding":<auditsyslogpolicy_vpnglobal_binding
[]_value>,

"auditsyslogpolicy_authenticationvserver_binding":<auditsyslogpolicy_authe
nticationvserver_binding[]_value>,

"auditsyslogpolicy_lbvserver_binding":<auditsyslogpolicy_lbvserver_binding
[]_value>,

"auditsyslogpolicy_aaagroup_binding":<auditsyslogpolicy_aaagroup_binding[]
_value>,

"auditsyslogpolicy_auditsyslogglobal_binding":<auditsyslogpolicy_auditsysl
ogglobal_binding[]_value>,

"auditsyslogpolicy_tmglobal_binding":<auditsyslogpolicy_tmglobal_binding[]
_value>,

"auditsyslogpolicy_rnatglobal_binding":<auditsyslogpolicy_rnatglobal_bindi
ng[]_value>,

"auditsyslogpolicy_csvserver_binding":<auditsyslogpolicy_csvserver_binding
[]_value>,

"auditsyslogpolicy_systemglobal_binding":<auditsyslogpolicy_systemglobal_b
inding[]_value>,
```

```
"auditsyslogpolicy_vpnvserver_binding":<auditsyslogpolicy_vpnvserver_binding[]_value>

}]}
```

auditsyslogpolicy_csvserver_binding

Binding object showing the csvserver that can be bound to auditsyslogpolicy.

Properties

(click to see Operations)

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
boundto	<String>	Read-write	The entity name to which policy is bound.
name	<String>	Read-write	Name of the policy. Minimum length = 1
priority	<Double>	Read-only	.
activepolicy	<Double>	Read-only	.
__count	<Double>	Read-only	count parameter

Operations

(click to see Properties)

GET| GET (ALL)| COUNT

Some options that you can use for each operations:

- **Getting warnings in response:**NITRO allows you to get warnings in an operation by specifying the "warning" query parameter as "yes". For example, to get warnings while connecting to the Citrix ADC appliance, the URL is as follows:

  http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/login?warning=yes

If any, the warnings are displayed in the response payload with the HTTP code "209 X-NITRO-WARNING".

- **Authenticated access for individual NITRO operations:**NITRO allows you to logon to the Citrix ADC appliance to perform individual operations. You can use this option instead of creating a NITRO session (using the login object) and then using that session to perform all operations,

  To do this, you must specify the username and password in the request header of the NITRO request as follows:

  X-NITRO-USER:<username>

  X-NITRO-PASS:<password>

  Note: In such cases, make sure that the request header DOES not include the following:

  Cookie:NITRO_AUTH_TOKEN=<tokenvalue>

`*Note:`*

Mandatory parameters are marked in redand placeholder content is marked in <green>.

get

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/auditsyslogpolicy_csvserver_binding/name_value<String>

Query-parameters:

filter

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/auditsyslogpolicy_csvserver_binding/name_value<String>?**filter=property-name1:property-value1,property-name2:property-value2**

Use this query-parameter to get the filtered set of auditsyslogpolicy_csvserver_binding resources configured on Citrix ADC. Filtering can be done on any of the properties of the resource.

pagination

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/auditsyslogpolicy_csvserver_binding/name_value<String>?**pagesize=#no;pageno=#no**

Use this query-parameter to get the auditsyslogpolicy_csvserver_binding resources in chunks.

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "auditsyslogpolicy_csvserver_binding": [ {

"boundto":<String_value>,

"name":<String_value>,

"priority":<Double_value>,

"activepolicy":<Double_value>

}]}
```

get (all)

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/auditsyslogpolicy_csvserver_binding

Query-parameters:

bulkbindings

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/auditsyslogpolicy_csvserver_binding?**bulkbindings=yes**

NITRO allows you to fetch bindings in bulk.

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "auditsyslogpolicy_csvserver_binding": [ {

"boundto":<String_value>,

"name":<String_value>,

"priority":<Double_value>,

"activepolicy":<Double_value>

}]}
```

count

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/auditsyslogpolicy_csvserver_binding/name_value<String>?**count=yes**

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{"auditsyslogpolicy_csvserver_binding": [ { "__count": "#no"} ] }
```

auditsyslogpolicy_lbvserver_binding

Binding object showing the lbvserver that can be bound to auditsyslogpolicy.

Properties

(click to see Operations)

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
boundto	<String>	Read-write	The entity name to which policy is bound.
name	<String>	Read-write	Name of the policy. Minimum length = 1
priority	<Double>	Read-only	.
activepolicy	<Double>	Read-only	.
__count	<Double>	Read-only	count parameter

Operations

(click to see Properties)

GET| GET (ALL)| COUNT

Some options that you can use for each operations:

- **Getting warnings in response:**NITRO allows you to get warnings in an operation by specifying the "warning" query parameter as "yes". For example, to get warnings while connecting to the Citrix ADC appliance, the URL is as follows:

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/login?warning=yes

If any, the warnings are displayed in the response payload with the HTTP code "209 X-NITRO-WARNING".

- **Authenticated access for individual NITRO operations:**NITRO allows you to logon to the Citrix ADC appliance to perform individual operations. You can use this option instead of creating a NITRO session (using the login object) and then using that session to perform all operations,

  To do this, you must specify the username and password in the request header of the NITRO request as follows:

  X-NITRO-USER:<username>

  X-NITRO-PASS:<password>

  Note: In such cases, make sure that the request header DOES not include the following:

  Cookie:NITRO_AUTH_TOKEN=<tokenvalue>

`*Note:`*

Mandatory parameters are marked in redand placeholder content is marked in <green>.

get

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/auditsyslogpolicy_lbvserver_binding/name_value<String>

Query-parameters:

filter

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/auditsyslogpolicy_lbvserver_binding/name_value<String>?**filter=property-name1:property-value1,property-name2:property-value2**

Use this query-parameter to get the filtered set of auditsyslogpolicy_lbvserver_binding resources configured on Citrix ADC. Filtering can be done on any of the properties of the resource.

pagination

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/auditsyslogpolicy_lbvserver_binding/name_value<String>?**pagesize=#no;pageno=#no**

Use this query-parameter to get the auditsyslogpolicy_lbvserver_binding resources in chunks.

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "auditsyslogpolicy_lbvserver_binding": [ {

"boundto":<String_value>,

"name":<String_value>,

"priority":<Double_value>,

"activepolicy":<Double_value>

}]}
```

get (all)

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/auditsyslogpolicy_lbvserver_binding

Query-parameters:

bulkbindings

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/auditsyslogpolicy_lbvserver_binding?**bulkbindings=yes**

NITRO allows you to fetch bindings in bulk.

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "auditsyslogpolicy_lbvserver_binding": [ {

"boundto":<String_value>,

"name":<String_value>,

"priority":<Double_value>,

"activepolicy":<Double_value>

}]}
```

count

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/auditsyslogpolicy_lbvserver_binding/name_value<String>?**count=yes**

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{"auditsyslogpolicy_lbvserver_binding": [ { "__count": "#no"} ] }
```

auditsyslogpolicy_rnatglobal_binding

Binding object showing the rnatglobal that can be bound to auditsyslogpolicy.

Properties

(click to see Operations)

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
boundto	<String>	Read-write	The entity name to which policy is bound.
name	<String>	Read-write	Name of the policy. Minimum length = 1
priority	<Double>	Read-only	.
activepolicy	<Double>	Read-only	.
__count	<Double>	Read-only	count parameter

Operations

(click to see Properties)

GET| GET (ALL)| COUNT

Some options that you can use for each operations:

- **Getting warnings in response:**NITRO allows you to get warnings in an operation by specifying the "warning" query parameter as "yes". For example, to get warnings while connecting to the Citrix ADC appliance, the URL is as follows:

  http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/login?warning=yes

  If any, the warnings are displayed in the response payload with the HTTP code "209 X-NITRO-WARNING".

- **Authenticated access for individual NITRO operations:**NITRO allows you to logon to the Citrix ADC appliance to perform individual operations. You can use this option instead of creating a NITRO session (using the login object) and then using that session to perform all operations,

  To do this, you must specify the username and password in the request header of the NITRO request as follows:

  X-NITRO-USER:<username>

  X-NITRO-PASS:<password>

  Note: In such cases, make sure that the request header DOES not include the following:

  Cookie:NITRO_AUTH_TOKEN=<tokenvalue>

`*Note:`*

Mandatory parameters are marked in redand placeholder content is marked in <green>.

get

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/auditsyslogpolicy_rnatglobal_binding/name_value<String>

Query-parameters:

filter

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/auditsyslogpolicy_rnatglobal_binding/name_value<String>?**filter=property-name1:property-value1,property-name2:property-value2**

Use this query-parameter to get the filtered set of auditsyslogpolicy_rnatglobal_binding resources configured on Citrix ADC. Filtering can be done on any of the properties of the resource.

pagination

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/auditsyslogpolicy_rnatglobal_binding/name_value<String>?**pagesize=#no;pageno=#no**

Use this query-parameter to get the auditsyslogpolicy_rnatglobal_binding resources in chunks.

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "auditsyslogpolicy_rnatglobal_binding": [ {

"boundto":<String_value>,

"name":<String_value>,

"priority":<Double_value>,

"activepolicy":<Double_value>

}]}
```

get (all)

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/auditsyslogpolicy_rnatglobal_binding

Query-parameters:

bulkbindings

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/auditsyslogpolicy_rnatglobal_binding?**bulkbindings=yes**

NITRO allows you to fetch bindings in bulk.

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "auditsyslogpolicy_rnatglobal_binding": [ {

"boundto":<String_value>,

"name":<String_value>,

"priority":<Double_value>,

"activepolicy":<Double_value>

}]}
```

count

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/auditsyslogpolicy_rnatglobal_binding/name_value<String>?**count=yes**

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{"auditsyslogpolicy_rnatglobal_binding": [ { "__count": "#no"} ] }
```

auditsyslogpolicy_systemglobal_binding

Binding object showing the systemglobal that can be bound to auditsyslogpolicy.

Properties

(click to see Operations)

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
boundto	<String>	Read-write	The entity name to which policy is bound.
name	<String>	Read-write	Name of the policy. Minimum length = 1
priority	<Double>	Read-only	.
activepolicy	<Double>	Read-only	.
__count	<Double>	Read-only	count parameter

Operations

(click to see Properties)

GET| GET (ALL)| COUNT

Some options that you can use for each operations:

- **Getting warnings in response:**NITRO allows you to get warnings in an operation by specifying the "warning" query parameter as "yes". For example, to get warnings while connecting to the Citrix ADC appliance, the URL is as follows:

  http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/login?warning=yes

  If any, the warnings are displayed in the response payload with the HTTP code "209 X-NITRO-WARNING".

- **Authenticated access for individual NITRO operations:**NITRO allows you to logon to the Citrix ADC appliance to perform individual operations. You can use this option instead of creating a NITRO session (using the login object) and then using that session to perform all operations,

  To do this, you must specify the username and password in the request header of the NITRO request as follows:

  X-NITRO-USER:<username>

  X-NITRO-PASS:<password>

  Note: In such cases, make sure that the request header DOES not include the following:

  Cookie:NITRO_AUTH_TOKEN=<tokenvalue>

`*Note:`*

Mandatory parameters are marked in redand placeholder content is marked in <green>.

get

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/auditsyslogpolicy_systemglobal_binding/name_value<String>

Query-parameters:

filter

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/auditsyslogpolicy_systemglobal_binding/name_value<String>?**filter=property-name1:property-value1,property-name2:property-value2**

Use this query-parameter to get the filtered set of auditsyslogpolicy_systemglobal_binding resources configured on Citrix ADC. Filtering can be done on any of the properties of the resource.

pagination

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/auditsyslogpolicy_systemglobal_binding/name_value<String>?**pagesize=#no;pageno=#no**

Use this query-parameter to get the auditsyslogpolicy_systemglobal_binding resources in chunks.

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "auditsyslogpolicy_systemglobal_binding": [ {

"boundto":<String_value>,

"name":<String_value>,

"priority":<Double_value>,

"activepolicy":<Double_value>

}]}
```

get (all)

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/auditsyslogpolicy_systemglobal_binding

Query-parameters:

bulkbindings

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/auditsyslogpolicy_systemglobal_binding?**bulkbindings=yes**

NITRO allows you to fetch bindings in bulk.

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "auditsyslogpolicy_systemglobal_binding": [ {

"boundto":<String_value>,

"name":<String_value>,

"priority":<Double_value>,

"activepolicy":<Double_value>

}]}
```

count

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/auditsyslogpolicy_systemglobal_binding/name_value<String>?**count=yes**

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{"auditsyslogpolicy_systemglobal_binding": [ { "__count": "#no"} ] }
```

auditsyslogpolicy_tmglobal_binding

Binding object showing the tmglobal that can be bound to auditsyslogpolicy.

Properties

(click to see Operations)

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
boundto	<String>	Read-write	The entity name to which policy is bound.
name	<String>	Read-write	Name of the policy. Minimum length = 1
priority	<Double>	Read-only	.
activepolicy	<Double>	Read-only	.
__count	<Double>	Read-only	count parameter

Operations

(click to see Properties)

GET| GET (ALL)| COUNT

Some options that you can use for each operations:

- **Getting warnings in response:**NITRO allows you to get warnings in an operation by specifying the "warning" query parameter as "yes". For example, to get warnings while connecting to the Citrix ADC appliance, the URL is as follows:

  http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/login?warning=yes

  If any, the warnings are displayed in the response payload with the HTTP code "209 X-NITRO-WARNING".

- **Authenticated access for individual NITRO operations:**NITRO allows you to logon to the Citrix ADC appliance to perform individual operations. You can use this option instead of creating a NITRO session (using the login object) and then using that session to perform all operations,

  To do this, you must specify the username and password in the request header of the NITRO request as follows:

  X-NITRO-USER:<username>

  X-NITRO-PASS:<password>

  Note: In such cases, make sure that the request header DOES not include the following:

  Cookie:NITRO_AUTH_TOKEN=<tokenvalue>

`*Note:`*

Mandatory parameters are marked in redand placeholder content is marked in <green>.

get

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/auditsyslogpolicy_tmglobal_binding/name_value<String>

Query-parameters:

filter

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/auditsyslogpolicy_tmglobal_binding/name_value<String>?**filter=property-name1:property-value1,property-name2:property-value2**

Use this query-parameter to get the filtered set of auditsyslogpolicy_tmglobal_binding resources configured on Citrix ADC. Filtering can be done on any of the properties of the resource.

pagination

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/auditsyslogpolicy_tmglobal_binding/name_value<String>?**pagesize=#no;pageno=#no**

Use this query-parameter to get the auditsyslogpolicy_tmglobal_binding resources in chunks.

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "auditsyslogpolicy_tmglobal_binding": [ {

"boundto":<String_value>,

"name":<String_value>,

"priority":<Double_value>,

"activepolicy":<Double_value>

}]}
```

get (all)

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/auditsyslogpolicy_tmglobal_binding

Query-parameters:

bulkbindings

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/auditsyslogpolicy_tmglobal_binding?**bulkbindings=yes**

NITRO allows you to fetch bindings in bulk.

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "auditsyslogpolicy_tmglobal_binding": [ {

"boundto":<String_value>,

"name":<String_value>,

"priority":<Double_value>,

"activepolicy":<Double_value>

}]}
```

count

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/auditsyslogpolicy_tmglobal_binding/name_value<String>?**count=yes**

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{"auditsyslogpolicy_tmglobal_binding": [ { "__count": "#no"} ] }
```

auditsyslogpolicy_vpnglobal_binding

Binding object showing the vpnglobal that can be bound to auditsyslogpolicy.

Properties

(click to see Operations)

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
boundto	<String>	Read-write	The entity name to which policy is bound.
name	<String>	Read-write	Name of the policy. Minimum length = 1
priority	<Double>	Read-only	.

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
activepolicy	<Double>	Read-only	.
__count	<Double>	Read-only	count parameter

Operations

(click to see Properties)

GET| GET (ALL)| COUNT

Some options that you can use for each operations:

- **Getting warnings in response:**NITRO allows you to get warnings in an operation by specifying the "warning" query parameter as "yes". For example, to get warnings while connecting to the Citrix ADC appliance, the URL is as follows:

  http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/login?warning=yes

  If any, the warnings are displayed in the response payload with the HTTP code "209 X-NITRO-WARNING".
- **Authenticated access for individual NITRO operations:**NITRO allows you to logon to the Citrix ADC appliance to perform individual operations. You can use this option instead of creating a NITRO session (using the login object) and then using that session to perform all operations,

  To do this, you must specify the username and password in the request header of the NITRO request as follows:

  X-NITRO-USER:<username>

  X-NITRO-PASS:<password>

  Note: In such cases, make sure that the request header DOES not include the following:

  Cookie:NITRO_AUTH_TOKEN=<tokenvalue>

`*Note:`*

Mandatory parameters are marked in redand placeholder content is marked in <green>.

get

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/auditsyslogpolicy_vpnglobal_binding/name_value<String>

Query-parameters:

filter

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/auditsyslogpolicy_vpnglobal_binding/name_value<String>?**filter=property-name1:property-value1,property-name2:property-value2**

Use this query-parameter to get the filtered set of auditsyslogpolicy_vpnglobal_binding resources configured on Citrix ADC. Filtering can be done on any of the properties of the resource.

pagination

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/auditsyslogpolicy_vpnglobal_binding/name_value<String>?**pagesize=#no;pageno=#no**

Use this query-parameter to get the auditsyslogpolicy_vpnglobal_binding resources in chunks.

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "auditsyslogpolicy_vpnglobal_binding": [ {

"boundto":<String_value>,

"name":<String_value>,
```

```
"priority":<Double_value>,

"activepolicy":<Double_value>

}]}
```

get (all)

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/auditsyslogpolicy_vpnglobal_binding

Query-parameters:

bulkbindings

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/auditsyslogpolicy_vpnglobal_binding?**bulkbindings=yes**

NITRO allows you to fetch bindings in bulk.

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "auditsyslogpolicy_vpnglobal_binding": [ {

"boundto":<String_value>,

"name":<String_value>,
```

```
"priority":<Double_value>,

"activepolicy":<Double_value>

}]}
```

count

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/auditsyslogpolicy_vpnglobal_binding/name_value<String>?**count=yes**

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{"auditsyslogpolicy_vpnglobal_binding": [ { "__count": "#no"} ] }
```

auditsyslogpolicy_vpnvserver_binding

Binding object showing the vpnvserver that can be bound to auditsyslogpolicy.

Properties

(click to see Operations)

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
boundto	<String>	Read-write	The entity name to which policy is bound.
name	<String>	Read-write	Name of the policy. Minimum length = 1
priority	<Double>	Read-only	.
activepolicy	<Double>	Read-only	.
__count	<Double>	Read-only	count parameter

Operations

(click to see Properties)

GET| GET (ALL)| COUNT

Some options that you can use for each operations:

- **Getting warnings in response:**NITRO allows you to get warnings in an operation by specifying the "warning" query parameter as "yes". For example, to get warnings while connecting to the Citrix ADC appliance, the URL is as follows:

  http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/login?warning=yes

  If any, the warnings are displayed in the response payload with the HTTP code "209 X-NITRO-WARNING".

- **Authenticated access for individual NITRO operations:**NITRO allows you to logon to the Citrix ADC appliance to perform individual operations. You can use this option instead of creating a NITRO session (using the login object) and then using that session to perform all operations,

  To do this, you must specify the username and password in the request header of the NITRO request as follows:

  X-NITRO-USER:<username>

  X-NITRO-PASS:<password>

Note: In such cases, make sure that the request header DOES not include the following:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue>

`*Note:`*

Mandatory parameters are marked in redand placeholder content is marked in <green>.

get

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/auditsyslogpolicy_vpnvserver_binding/name_value<String>

Query-parameters:

filter

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/auditsyslogpolicy_vpnvserver_binding/name_value<String>?**filter=property-name1:property-value1,property-name2:property-value2**

Use this query-parameter to get the filtered set of auditsyslogpolicy_vpnvserver_binding resources configured on Citrix ADC. Filtering can be done on any of the properties of the resource.

pagination

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/auditsyslogpolicy_vpnvserver_binding/name_value<String>?**pagesize=#no;pageno=#no**

Use this query-parameter to get the auditsyslogpolicy_vpnvserver_binding resources in chunks.

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "auditsyslogpolicy_vpnvserver_binding": [ {

"boundto":<String_value>,

"name":<String_value>,

"priority":<Double_value>,

"activepolicy":<Double_value>

}]}
```

get (all)

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/auditsyslogpolicy_vpnvserver_binding

Query-parameters:

bulkbindings

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/auditsyslogpolicy_vpnvserver_binding?**bulkbindings=yes**

NITRO allows you to fetch bindings in bulk.

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "auditsyslogpolicy_vpnvserver_binding": [ {

"boundto":<String_value>,

"name":<String_value>,

"priority":<Double_value>,

"activepolicy":<Double_value>

}]}
```

count

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/auditsyslogpolicy_vpnvserver_binding/name_value<String>?**count=yes**

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{"auditsyslogpolicy_vpnvserver_binding": [ { "__count": "#no"} ] }
```

Authentication

Authentication

Authentication configuration.

NAME	DESCRIPTION
authenticationadfsproxyprofile	Configuration for ADFSProxy Profile
authenticationauthnprofile	Configuration for Authentication profile
authenticationazurekeyvault	Configuration for Azure Key Vault entity
authenticationcaptchaaction	Configuration for Captcha Action
authenticationcertaction	Configuration for CERT action
authenticationcertpolicy	Configuration for CERT policy
authenticationcertpolicy_authenticationvserver_binding	Binding object showing the authenticationvserver that can be bound to authenticationcertpolicy
authenticationcertpolicy_binding	Binding object showing the resources that can be bound to authenticationcertpolicy
authenticationcertpolicy_vpnglobal_binding	Binding object showing the vpnglobal that can be bound to authenticationcertpolicy
authenticationcertpolicy_vpnvserver_binding	Binding object showing the vpnvserver that can be bound to authenticationcertpolicy
authenticationcitrixauthaction	Configuration for Citrix Authentication action
authenticationdfaaction	Configuration for Dfa authentication action
authenticationdfapolicy	Configuration for Dfa authentication policy
authenticationdfapolicy_binding	Binding object showing the resources that can be bound to authenticationdfapolicy
authenticationdfapolicy_vpnvserver_binding	Binding object showing the vpnvserver that can be bound to authenticationdfapolicy

NAME	DESCRIPTION
authenticationemailaction	Configuration for Email entity
authenticationepaaction	Configuration for epa action
authenticationldapaction	Configuration for LDAP action
authenticationldappolicy	Configuration for LDAP policy
authenticationldappolicy_authenticationvserver_binding	Binding object showing the authenticationvserver that can be bound to authenticationldappolicy
authenticationldappolicy_binding	Binding object showing the resources that can be bound to authenticationldappolicy
authenticationldappolicy_systemglobal_binding	Binding object showing the systemglobal that can be bound to authenticationldappolicy
authenticationldappolicy_vpnglobal_binding	Binding object showing the vpnglobal that can be bound to authenticationldappolicy
authenticationldappolicy_vpnvserver_binding	Binding object showing the vpnvserver that can be bound to authenticationldappolicy
authenticationlocalpolicy	Configuration for LOCAL policy
authenticationlocalpolicy_authenticationvserver_binding	Binding object showing the authenticationvserver that can be bound to authenticationlocalpolicy
authenticationlocalpolicy_binding	Binding object showing the resources that can be bound to authenticationlocalpolicy
authenticationlocalpolicy_systemglobal_binding	Binding object showing the systemglobal that can be bound to authenticationlocalpolicy
authenticationlocalpolicy_vpnglobal_binding	Binding object showing the vpnglobal that can be bound to authenticationlocalpolicy
authenticationlocalpolicy_vpnvserver_binding	Binding object showing the vpnvserver that can be bound to authenticationlocalpolicy

NAME	DESCRIPTION
authenticationloginschema	
authenticationloginschemapolicy	
authenticationloginschemapolicy_authenticationvserver_binding	
authenticationloginschemapolicy_binding	
authenticationloginschemapolicy_vpnvserver_binding	
authenticationnegotiateaction	Configuration for Negotiate action
authenticationnegotiatepolicy	Configuration for Negotiate Policy
authenticationnegotiatepolicy_authenticationvserver_binding	Binding object showing the authenticationvserver that can be bound to authenticationnegotiatepolicy
authenticationnegotiatepolicy_binding	Binding object showing the resources that can be bound to authenticationnegotiatepolicy
authenticationnegotiatepolicy_vpnglobal_binding	Binding object showing the vpnglobal that can be bound to authenticationnegotiatepolicy
authenticationnegotiatepolicy_vpnvserver_binding	Binding object showing the vpnvserver that can be bound to authenticationnegotiatepolicy
authenticationnoauthaction	Configuration for no authentication action
authenticationoauthaction	Configuration for OAuth authentication action
authenticationoauthidppolicy	Configuration for AAA OAuth IdentityProvider (IdP) policy
authenticationoauthidppolicy_authenticationvserver_binding	Binding object showing the authenticationvserver that can be bound to authenticationoauthidppolicy
authenticationoauthidppolicy_binding	Binding object showing the resources that can be bound to authenticationoauthidppolicy

NAME	DESCRIPTION
authenticationoauthidppolicy_vpnvserver_binding	Binding object showing the vpnvserver that can be bound to authenticationoauthidppolicy
authenticationoauthidpprofile	Configuration for OAuth Identity Provider (IdP) profile
authenticationpolicy	Configuration for Authentication Policy
authenticationpolicy_authenticationpolicylabel_binding	Binding object showing the authenticationpolicylabel that can be bound to authenticationpolicy
authenticationpolicy_authenticationvserver_binding	Binding object showing the authenticationvserver that can be bound to authenticationpolicy
authenticationpolicy_binding	Binding object showing the resources that can be bound to authenticationpolicy
authenticationpolicy_systemglobal_binding	Binding object showing the systemglobal that can be bound to authenticationpolicy
authenticationpolicylabel	Configuration for authentication policy label
authenticationpolicylabel_authenticationpolicy_binding	Binding object showing the authenticationpolicy that can be bound to authenticationpolicylabel
authenticationpolicylabel_binding	Binding object showing the resources that can be bound to authenticationpolicylabel
authenticationpushservice	Configuration for Service details for sending push notifications
authenticationradiusaction	Configuration for RADIUS action
authenticationradiuspolicy	Configuration for RADIUS policy
authenticationradiuspolicy_authenticationvserver_binding	Binding object showing the authenticationvserver that can be bound to authenticationradiuspolicy

NAME	DESCRIPTION
authenticationradiuspolicy_binding	Binding object showing the resources that can be bound to authenticationradiuspolicy
authenticationradiuspolicy_systemglobal_binding	Binding object showing the systemglobal that can be bound to authenticationradiuspolicy
authenticationradiuspolicy_vpnglobal_binding	Binding object showing the vpnglobal that can be bound to authenticationradiuspolicy
authenticationradiuspolicy_vpnvserver_binding	Binding object showing the vpnvserver that can be bound to authenticationradiuspolicy
authenticationsamlaction	Configuration for AAA Saml action
authenticationsamlidppolicy	Configuration for AAA Saml IdentityProvider (IdP) policy
authenticationsamlidppolicy_authenticationvserver_binding	Binding object showing the authenticationvserver that can be bound to authenticationsamlidppolicy
authenticationsamlidppolicy_binding	Binding object showing the resources that can be bound to authenticationsamlidppolicy
authenticationsamlidppolicy_vpnvserver_binding	Binding object showing the vpnvserver that can be bound to authenticationsamlidppolicy
authenticationsamlidpprofile	Configuration for AAA Saml IdentityProvider (IdP) profile
authenticationsamlpolicy	Configuration for AAA Saml policy
authenticationsamlpolicy_authenticationvserver_binding	Binding object showing the authenticationvserver that can be bound to authenticationsamlpolicy
authenticationsamlpolicy_binding	Binding object showing the resources that can be bound to authenticationsamlpolicy
authenticationsamlpolicy_vpnglobal_binding	Binding object showing the vpnglobal that can be bound to authenticationsamlpolicy

NAME	DESCRIPTION
authenticationsamlpolicy_vpnvserver_binding	Binding object showing the vpnvserver that can be bound to authenticationsamlpolicy
authenticationstorefrontauthaction	Configuration for Storefront authentication action
authenticationtacacsaction	Configuration for TACACS action
authenticationtacacspolicy	Configuration for TACACS policy
authenticationtacacspolicy_authenticationvserver_binding	Binding object showing the authenticationvserver that can be bound to authenticationtacacspolicy
authenticationtacacspolicy_binding	Binding object showing the resources that can be bound to authenticationtacacspolicy
authenticationtacacspolicy_systemglobal_binding	Binding object showing the systemglobal that can be bound to authenticationtacacspolicy
authenticationtacacspolicy_vpnglobal_binding	Binding object showing the vpnglobal that can be bound to authenticationtacacspolicy
authenticationtacacspolicy_vpnvserver_binding	Binding object showing the vpnvserver that can be bound to authenticationtacacspolicy
authenticationvserver	Configuration for authentication virtual server
authenticationvserver_auditnslogpolicy_binding	Binding object showing the auditnslogpolicy that can be bound to authenticationvserver
authenticationvserver_auditsyslogpolicy_binding	Binding object showing the auditsyslogpolicy that can be bound to authenticationvserver
authenticationvserver_authenticationcertpolicy_binding	Binding object showing the authenticationcertpolicy that can be bound to authenticationvserver
authenticationvserver_authenticationldappolicy_binding	Binding object showing the authenticationldappolicy that can be bound to authenticationvserver

NAME	DESCRIPTION
authenticationvserver_authenticationlocalpolicy_binding	Binding object showing the authenticationlocalpolicy that can be bound to authenticationvserver
authenticationvserver_authenticationloginschemapolicy_binding	Binding object showing the authenticationloginschemapolicy that can be bound to authenticationvserver
authenticationvserver_authenticationnegotiatepolicy_binding	Binding object showing the authenticationnegotiatepolicy that can be bound to authenticationvserver
authenticationvserver_authenticationoauthidppolicy_binding	Binding object showing the authenticationoauthidppolicy that can be bound to authenticationvserver
authenticationvserver_authenticationpolicy_binding	Binding object showing the authenticationpolicy that can be bound to authenticationvserver
authenticationvserver_authenticationradiuspolicy_binding	Binding object showing the authenticationradiuspolicy that can be bound to authenticationvserver
authenticationvserver_authenticationsamlidppolicy_binding	Binding object showing the authenticationsamlidppolicy that can be bound to authenticationvserver
authenticationvserver_authenticationsamlpolicy_binding	Binding object showing the authenticationsamlpolicy that can be bound to authenticationvserver
authenticationvserver_authenticationtacacspolicy_binding	Binding object showing the authenticationtacacspolicy that can be bound to authenticationvserver
authenticationvserver_authenticationwebauthpolicy_binding	Binding object showing the authenticationwebauthpolicy that can be bound to authenticationvserver

NAME	DESCRIPTION
authenticationvserver_binding	Binding object showing the resources that can be bound to authenticationvserver
authenticationvserver_cspolicy_binding	Binding object showing the cspolicy that can be bound to authenticationvserver
authenticationvserver_tmsessionpolicy_binding	Binding object showing the tmsessionpolicy that can be bound to authenticationvserver
authenticationvserver_vpnportaltheme_binding	Binding object showing the vpnportaltheme that can be bound to authenticationvserver
authenticationwebauthaction	Configuration for Web authentication action
authenticationwebauthpolicy	Configuration for Web authentication policy
authenticationwebauthpolicy_authenticationvserver_binding	Binding object showing the authenticationvserver that can be bound to authenticationwebauthpolicy
authenticationwebauthpolicy_binding	Binding object showing the resources that can be bound to authenticationwebauthpolicy
authenticationwebauthpolicy_systemglobal_binding	Binding object showing the systemglobal that can be bound to authenticationwebauthpolicy
authenticationwebauthpolicy_vpnglobal_binding	Binding object showing the vpnglobal that can be bound to authenticationwebauthpolicy
authenticationwebauthpolicy_vpnvserver_binding	Binding object showing the vpnvserver that can be bound to authenticationwebauthpolicy

authenticationauthnprofile

Configuration for Authentication profile resource.

Properties

(click to see Operations)

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
name	<String>	Read-write	Name for the authentication profile. Must begin with a letter, number, or the underscore character (_), and must contain only letters, numbers, and the hyphen (-), period (.) pound (#), space ( ), at (@), equals (=), colon (:), and underscore characters. Cannot be changed after the RADIUS action is added. Minimum length = 1
authnvsname	<String>	Read-write	Name of the authentication vserver at which authentication should be done. Minimum length = 1 Maximum length = 128
authenticationhost	<String>	Read-write	Hostname of the authentication vserver to which user must be redirected for authentication. Minimum length = 1

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
			Maximum length = 256
authenticationdomain	<String>	Read-write	Domain for which TM cookie must to be set. If unspecified, cookie will be set for FQDN. Minimum length = 1 Maximum length = 256
authenticationlevel	<Double>	Read-write	Authentication weight or level of the vserver to which this will bound. This is used to order TM vservers based on the protection required. A session that is created by authenticating against TM vserver at given level cannot be used to access TM vserver at a higher level. Minimum value = 0 Maximum value = 255
__count	<Double>	Read-only	count parameter

Operations

(click to see Properties)

ADD| DELETE| UPDATE| UNSET| GET (ALL)| GET| COUNT

Some options that you can use for each operations:

- **Getting warnings in response:**NITRO allows you to get warnings in an operation by specifying the "warning" query parameter as "yes". For example, to get warnings while connecting to the Citrix ADC appliance, the URL is as follows:

  http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/login?warning=yes

  If any, the warnings are displayed in the response payload with the HTTP code "209 X-NITRO-WARNING".

- **Authenticated access for individual NITRO operations:**NITRO allows you to logon to the Citrix ADC appliance to perform individual operations. You can use this option instead of creating a NITRO session (using the login object) and then using that session to perform all operations,

  To do this, you must specify the username and password in the request header of the NITRO request as follows:

  X-NITRO-USER:<username>

  X-NITRO-PASS:<password>

  Note: In such cases, make sure that the request header DOES not include the following:

  Cookie:NITRO_AUTH_TOKEN=<tokenvalue>

`*Note:`*

Mandatory parameters are marked in redand placeholder content is marked in <green>.

add

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/authenticationauthnprofile

HTTP Method:POST

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Content-Type:application/json

Request Payload:

```
{"authenticationauthnprofile":{

<b>"name":<String_value>,

</b><b>"authnvsname":<String_value>,

</b>"authenticationhost":<String_value>,

"authenticationdomain":<String_value>,

"authenticationlevel":<Double_value>

}}
```

Response:

HTTP Status Code on Success: 201 Created HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error

delete

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/authenticationauthnprofile/name_value<String>

HTTP Method:DELETE

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue>

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error

update

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/authenticationauthnprofile

HTTP Method:PUT

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Content-Type:application/json

Request Payload:

```
{"authenticationauthnprofile":{

<b>"name":<String_value>,

</b>"authnvsname":<String_value>,

"authenticationhost":<String_value>,

"authenticationdomain":<String_value>,

"authenticationlevel":<Double_value>

}}
```

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error

unset

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/authenticationauthnprofile?**action=unset**

HTTP Method:POST

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Content-Type:application/json

Request Payload:

```
{"authenticationauthnprofile":{

<b>"name":<String_value>,

</b>"authenticationdomain":true,

"authenticationlevel":true

}}
```

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error

get (all)

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/authenticationauthnprofile

Query-parameters:

attrs

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/authenticationauthnprofile?**attrs=property-name1,property-name2**

Use this query parameter to specify the resource details that you want to retrieve.

filter

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/authenticationauthnprofile?**filter=property-name1:property-val1,property-name2:property-val2**

Use this query-parameter to get the filtered set of authenticationauthnprofile resources configured on Citrix ADC. Filtering can be done on any of the properties of the resource.

view

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/authenticationauthnprofile?**view=summary**

Note: By default, the retrieved results are displayed in detail view (?view=detail).

pagination

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/authenticationauthnprofile?**pagesize=#no;pageno=#no**

Use this query-parameter to get the authenticationauthnprofile resources in chunks.

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "authenticationauthnprofile": [ {

"name":<String_value>,

"authnvsname":<String_value>,

"authenticationhost":<String_value>,

"authenticationdomain":<String_value>,

"authenticationlevel":<Double_value>

}]}
```

get

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/authenticationauthnprofile/name_value<String>

Query-parameters:

attrs

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/authenticationauthnprofile/name_value<String>?**attrs=property-name1,property-name2**

Use this query parameter to specify the resource details that you want to retrieve.

view

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/authenticationauthnprofile/name_value<String>?**view=summary**

Note: By default, the retrieved results are displayed in detail view (?view=detail).

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "authenticationauthnprofile": [ {

"name":<String_value>,

"authnvsname":<String_value>,
```

```
"authenticationhost":<String_value>,

"authenticationdomain":<String_value>,

"authenticationlevel":<Double_value>

}]}
```

count

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/authenticationauthnprofile?**count=yes**

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "authenticationauthnprofile": [ { "__count": "#no"} ] }
```

authenticationcertaction

Configuration for CERT action resource.

Properties

(click to see Operations)

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
name	<String>	Read-write	Name for the client cert authentication server profile (action). Must begin with a letter, number, or the underscore character (_), and must contain only letters, numbers, and the hyphen (-), period (.) pound (#), space ( ), at (@), equals (=), colon (:), and underscore characters. Cannot be changed after certifcate action is created. The following requirement applies only to the Citrix ADC CLI: If the name includes one or more spaces, enclose the name in double or single quotation marks (for example, "my authentication action" or 'my authentication action'). Minimum length = 1

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
twofactor	<String>	Read-write	Enables or disables two-factor authentication. Two factor authentication is client cert authentication followed by password authentication. Default value: OFF Possible values = ON, OFF
usernamefield	<String>	Read-write	Client-cert field from which the username is extracted. Must be set to either ""Subject"" and ""Issuer"" (include both sets of double quotation marks). Format: <field>:<subfield>. Minimum length = 1
groupnamefield	<String>	Read-write	Client-cert field from which the group is extracted. Must be set to either ""Subject"" and ""Issuer"" (include both sets of double quotation marks). Format: <field>:<subfield>.

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
			Minimum length = 1
defaultauthentication group	<String>	Read-write	This is the default group that is chosen when the authentication succeeds in addition to extracted groups.
__count	<Double>	Read-only	count parameter

Operations

(click to see Properties)

ADD| DELETE| UPDATE| UNSET| GET (ALL)| GET| COUNT

Some options that you can use for each operations:

- **Getting warnings in response:**NITRO allows you to get warnings in an operation by specifying the "warning" query parameter as "yes". For example, to get warnings while connecting to the Citrix ADC appliance, the URL is as follows:

  http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/login?warning=yes

  If any, the warnings are displayed in the response payload with the HTTP code "209 X-NITRO-WARNING".

- **Authenticated access for individual NITRO operations:**NITRO allows you to logon to the Citrix ADC appliance to perform individual operations. You can use this option instead of creating a NITRO session (using the login object) and then using that session to perform all operations,

  To do this, you must specify the username and password in the request header of the NITRO request as follows:

  X-NITRO-USER:<username>

  X-NITRO-PASS:<password>

  Note: In such cases, make sure that the request header DOES not include the following:

  Cookie:NITRO_AUTH_TOKEN=<tokenvalue>

`*Note:` *

Mandatory parameters are marked in redand placeholder content is marked in <green>.

add

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/authenticationcertaction

HTTP Method:POST

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Content-Type:application/json

Request Payload:

```
{"authenticationcertaction":{

<b>"name":<String_value>,

</b>"twofactor":<String_value>,

"usernamefield":<String_value>,

"groupnamefield":<String_value>,

"defaultauthenticationgroup":<String_value>

}}
```

Response:

HTTP Status Code on Success: 201 Created HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error

delete

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/authenticationcertaction/name_value<String>

HTTP Method:DELETE

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue>

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error

update

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/authenticationcertaction

HTTP Method:PUT

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Content-Type:application/json

Request Payload:

```
{"authenticationcertaction":{

<b>"name":<String_value>,

</b>"twofactor":<String_value>,

"usernamefield":<String_value>,

"groupnamefield":<String_value>,

"defaultauthenticationgroup":<String_value>

}}
```

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error

unset

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/authenticationcertaction?**action=unset**

HTTP Method:POST

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Content-Type:application/json

Request Payload:

```
{"authenticationcertaction":{

<b>"name":<String_value>,

</b>"twofactor":true,

"usernamefield":true,

"groupnamefield":true,

"defaultauthenticationgroup":true

}}
```

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error

get (all)

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/authenticationcertaction

Query-parameters:

attrs

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/authenticationcertaction?**attrs=property-name1,property-name2**

Use this query parameter to specify the resource details that you want to retrieve.

filter

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/authenticationcertaction?**filter=property-name1:property-val1,property-name2:property-val2**

Use this query-parameter to get the filtered set of authenticationcertaction resources configured on Citrix ADC. Filtering can be done on any of the properties of the resource.

view

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/authenticationcertaction?**view=summary**

Note: By default, the retrieved results are displayed in detail view (?view=detail).

pagination

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/authenticationcertaction?**pagesize=#no;pageno=#no**

Use this query-parameter to get the authenticationcertaction resources in chunks.

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "authenticationcertaction": [ {

"name":<String_value>,
```

```
"twofactor":<String_value>,

"usernamefield":<String_value>,

"groupnamefield":<String_value>,

"defaultauthenticationgroup":<String_value>

}]}
```

get

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/authenticationcertaction/name_value<String>

Query-parameters:

attrs

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/authenticationcertaction/name_value<String>?**attrs=property-name1,property-name2**

Use this query parameter to specify the resource details that you want to retrieve.

view

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/authenticationcertaction/name_value<String>?**view=summary**

Note: By default, the retrieved results are displayed in detail view (?view=detail).

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "authenticationcertaction": [ {

"name":<String_value>,

"twofactor":<String_value>,

"usernamefield":<String_value>,

"groupnamefield":<String_value>,

"defaultauthenticationgroup":<String_value>

}]}
```

count

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/authenticationcertaction?**count=yes**

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "authenticationcertaction": [ { "__count": "#no"} ] }
```

authenticationcertpolicy

Configuration for CERT policy resource.

Properties

(click to see Operations)

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
name	<String>	Read-write	Name for the client certificate authentication policy. Must begin with a letter, number, or the underscore character (_), and must contain only letters, numbers, and the hyphen (-), period (.) pound (#), space ( ), at (@), equals (=), colon (:), and underscore characters. Cannot be changed after cert authentication policy is created. The following requirement applies only to the Citrix ADC CLI: If the name includes one or more spaces, enclose the name in double or

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
			single quotation marks (for example, "my authentication policy" or 'my authentication policy'). Minimum length = 1
rule	<String>	Read-write	Name of the Citrix ADC named rule, or an expression, that the policy uses to determine whether to attempt to authenticate the user with the authentication server. Minimum length = 1
reqaction	<String>	Read-write	Name of the client cert authentication action to be performed if the policy matches. Minimum length = 1
__count	<Double>	Read-only	count parameter

Operations

(click to see Properties)

ADD| DELETE| UPDATE| UNSET| GET (ALL)| GET| COUNT

Some options that you can use for each operations:

- **Getting warnings in response:**NITRO allows you to get warnings in an operation by specifying the "warning" query parameter as "yes". For example, to get warnings while connecting to the Citrix ADC appliance, the URL is as follows:

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/login?warning=yes

If any, the warnings are displayed in the response payload with the HTTP code "209 X-NITRO-WARNING".

- **Authenticated access for individual NITRO operations:**NITRO allows you to logon to the Citrix ADC appliance to perform individual operations. You can use this option instead of creating a NITRO session (using the login object) and then using that session to perform all operations,

  To do this, you must specify the username and password in the request header of the NITRO request as follows:

  X-NITRO-USER:<username>

  X-NITRO-PASS:<password>

  Note: In such cases, make sure that the request header DOES not include the following:

  Cookie:NITRO_AUTH_TOKEN=<tokenvalue>

`*Note:`*

Mandatory parameters are marked in redand placeholder content is marked in <green>.

add

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/authenticationcertpolicy

HTTP Method:POST

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Content-Type:application/json

Request Payload:

```
{"authenticationcertpolicy":{

<b>"name":<String_value>,

</b><b>"rule":<String_value>,

</b>"reqaction":<String_value>
```

```
}}
```

Response:

HTTP Status Code on Success: 201 Created HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error

delete

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/authenticationcertpolicy/name_value<String>

HTTP Method:DELETE

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue>

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error

update

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/authenticationcertpolicy

HTTP Method:PUT

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Content-Type:application/json

Request Payload:

```
{"authenticationcertpolicy":{

<b>"name":<String_value>,
```

```
</b>"rule":<String_value>,

"reqaction":<String_value>

}}
```

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error

unset

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/authenticationcertpolicy?**action=unset**

HTTP Method:POST

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Content-Type:application/json

Request Payload:

```
{"authenticationcertpolicy":{

<b>"name":<String_value>,

</b>"rule":true,

"reqaction":true

}}
```

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error

get (all)

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/authenticationcertpolicy

Query-parameters:

attrs

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/authenticationcertpolicy?**attrs=property-name1,property-name2**

Use this query parameter to specify the resource details that you want to retrieve.

filter

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/authenticationcertpolicy?**filter=property-name1:property-val1,property-name2:property-val2**

Use this query-parameter to get the filtered set of authenticationcertpolicy resources configured on Citrix ADC. Filtering can be done on any of the properties of the resource.

view

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/authenticationcertpolicy?**view=summary**

Note: By default, the retrieved results are displayed in detail view (?view=detail).

pagination

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/authenticationcertpolicy?**pagesize=#no;pageno=#no**

Use this query-parameter to get the authenticationcertpolicy resources in chunks.

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "authenticationcertpolicy": [ {

"name":<String_value>,

"rule":<String_value>,

"reqaction":<String_value>

}]}
```

get

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/authenticationcertpolicy/name_value<String>

Query-parameters:

attrs

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/authenticationcertpolicy/name_value<String>?**attrs=property-name1,property-name2**

Use this query parameter to specify the resource details that you want to retrieve.

view

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/authenticationcertpolicy/name_value<String>?**view=summary**

Note: By default, the retrieved results are displayed in detail view (?view=detail).

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "authenticationcertpolicy": [ {

"name":<String_value>,

"rule":<String_value>,

"reqaction":<String_value>

}]}
```

count

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/authenticationcertpolicy?**count=yes**

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "authenticationcertpolicy": [ { "__count": "#no"} ] }
```

authenticationcertpolicy_authenticationvserver_binding

Binding object showing the authenticationvserver that can be bound to authenticationcertpolicy.

Properties

(click to see Operations)

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
name	<String>	Read-write	Name of the client cert authentication policy. Minimum length = 1
boundto	<String>	Read-write	The entity name to which policy is bound.
priority	<Double>	Read-only	.
activepolicy	<Double>	Read-only	.
__count	<Double>	Read-only	count parameter

Operations

(click to see Properties)

GET| GET (ALL)| COUNT

Some options that you can use for each operations:

- **Getting warnings in response:**NITRO allows you to get warnings in an operation by specifying the "warning" query parameter as "yes". For example, to get warnings while connecting to the Citrix ADC appliance, the URL is as follows:

  http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/login?warning=yes

If any, the warnings are displayed in the response payload with the HTTP code "209 X-NITRO-WARNING".

- **Authenticated access for individual NITRO operations:**NITRO allows you to logon to the Citrix ADC appliance to perform individual operations. You can use this option instead of creating a NITRO session (using the login object) and then using that session to perform all operations,

  To do this, you must specify the username and password in the request header of the NITRO request as follows:

  X-NITRO-USER:<username>

  X-NITRO-PASS:<password>

  Note: In such cases, make sure that the request header DOES not include the following:

  Cookie:NITRO_AUTH_TOKEN=<tokenvalue>

`*Note:`*

Mandatory parameters are marked in redand placeholder content is marked in <green>.

get

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/authenticationcertpolicy_authenticationvserver_binding/name_value<String>

Query-parameters:

filter

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/authenticationcertpolicy_authenticationvserver_binding/name_value<String>?**filter=property-name1:property-value1,property-name2:property-value2**

Use this query-parameter to get the filtered set of authenticationcertpolicy_authenticationvserver_binding resources configured on Citrix ADC. Filtering can be done on any of the properties of the resource.

pagination

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/authenticationcertpolicy_authenticationvserver_binding/name_value<String>?**pagesize=#no;pageno=#no**

Use this query-parameter to get the authenticationcertpolicy_authenticationvserver_binding resources in chunks.

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "authenticationcertpolicy_authenticationvserver_binding": [ {

"name":<String_value>,

"boundto":<String_value>,

"priority":<Double_value>,

"activepolicy":<Double_value>

}]}
```

get (all)

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/authenticationcertpolicy_authenticationvserver_binding

Query-parameters:

bulkbindings

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/authenticationcertpolicy_authenticationvserver_binding?**bulkbindings=yes**

NITRO allows you to fetch bindings in bulk.

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "authenticationcertpolicy_authenticationvserver_binding": [ {

"name":<String_value>,

"boundto":<String_value>,

"priority":<Double_value>,

"activepolicy":<Double_value>

}]}
```

count

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/authenticationcertpolicy_authenticationvserver_binding/name_value<String>?**count=yes**

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{"authenticationcertpolicy_authenticationvserver_binding": [ { "__count":
"#no"} ] }
```

authenticationcertpolicy_binding

Binding object which returns the resources bound to authenticationcertpolicy.

Properties

(click to see Operations)

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
name	<String>	Read-write	Name of the client cert authentication policy. Minimum length = 1
authenticationcertpolicy_vpnglobal_binding	<authenticationcertpolicy_vpnglobal_binding[]>	Read-only	vpnglobal that can be bound to authenticationcertpolicy.
authenticationcertpolicy_vpnvserver_binding	<authenticationcertpolicy_vpnvserver_binding[]>	Read-only	vpnvserver that can be bound to authenticationcertpolicy.

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
authenticationcertpolicy_authenticationvserver_binding	<authenticationcertpolicy_authenticationvserver_binding[]>	Read-only	authenticationvserver that can be bound to authenticationcertpolicy.

Operations

(click to see Properties)

GET| GET (ALL)

Some options that you can use for each operations:

- **Getting warnings in response:**NITRO allows you to get warnings in an operation by specifying the "warning" query parameter as "yes". For example, to get warnings while connecting to the Citrix ADC appliance, the URL is as follows:

  http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/login?warning=yes

  If any, the warnings are displayed in the response payload with the HTTP code "209 X-NITRO-WARNING".

- **Authenticated access for individual NITRO operations:**NITRO allows you to logon to the Citrix ADC appliance to perform individual operations. You can use this option instead of creating a NITRO session (using the login object) and then using that session to perform all operations,

  To do this, you must specify the username and password in the request header of the NITRO request as follows:

  X-NITRO-USER:<username>

  X-NITRO-PASS:<password>

  Note: In such cases, make sure that the request header DOES not include the following:

  Cookie:NITRO_AUTH_TOKEN=<tokenvalue>

`*Note:`*

Mandatory parameters are marked in redand placeholder content is marked in <green>.

get

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/authenticationcertpolicy_binding/name_value<String>

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "authenticationcertpolicy_binding": [ {

"name":<String_value>,

"authenticationcertpolicy_authenticationvserver_binding":<authenticationcertpolicy_authenticationvserver_binding[]_value>,

"authenticationcertpolicy_vpnvserver_binding":<authenticationcertpolicy_vpnvserver_binding[]_value>,

"authenticationcertpolicy_vpnglobal_binding":<authenticationcertpolicy_vpnglobal_binding[]_value>

}]}
```

get (all)

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/authenticationcertpolicy_binding

Query-parameters:

bulkbindings

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/authenticationcertpolicy_binding?**bulkbindings=yes**

NITRO allows you to fetch bindings in bulk.

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "authenticationcertpolicy_binding": [ {

"name":<String_value>,

"authenticationcertpolicy_authenticationvserver_binding":<authenticationcertpolicy_authenticationvserver_binding[]_value>,

"authenticationcertpolicy_vpnvserver_binding":<authenticationcertpolicy_vpnvserver_binding[]_value>,

"authenticationcertpolicy_vpnglobal_binding":<authenticationcertpolicy_vpnglobal_binding[]_value>

}]}
```

authenticationcertpolicy_vpnglobal_binding

Binding object showing the vpnglobal that can be bound to authenticationcertpolicy.

Properties

(click to see Operations)

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
name	<String>	Read-write	Name of the client cert authentication policy. Minimum length = 1
boundto	<String>	Read-write	The entity name to which policy is bound.
priority	<Double>	Read-only	.
activepolicy	<Double>	Read-only	.
__count	<Double>	Read-only	count parameter

Operations

(click to see Properties)

GET| GET (ALL)| COUNT

Some options that you can use for each operations:

- **Getting warnings in response:**NITRO allows you to get warnings in an operation by specifying the "warning" query parameter as "yes". For example, to get warnings while connecting to the Citrix ADC appliance, the URL is as follows:

  http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/login?warning=yes

  If any, the warnings are displayed in the response payload with the HTTP code "209 X-NITRO-WARNING".
- **Authenticated access for individual NITRO operations:**NITRO allows you to logon to the Citrix ADC appliance to perform individual operations. You can use this option instead of creating a NITRO session (using the login object) and then using that session to perform all operations,

To do this, you must specify the username and password in the request header of the NITRO request as follows:

X-NITRO-USER:<username>

X-NITRO-PASS:<password>

Note: In such cases, make sure that the request header DOES not include the following:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue>

`*Note:`*

Mandatory parameters are marked in redand placeholder content is marked in <green>.

get

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/authenticationcertpolicy_vpnglobal_binding/name_value<String>

Query-parameters:

filter

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/authenticationcertpolicy_vpnglobal_binding/name_value<String>?**filter=property-name1:property-value1,property-name2:property-value2**

Use this query-parameter to get the filtered set of authenticationcertpolicy_vpnglobal_binding resources configured on Citrix ADC. Filtering can be done on any of the properties of the resource.

pagination

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/authenticationcertpolicy_vpnglobal_binding/name_value<String>?**pagesize=#no;pageno=#no**

Use this query-parameter to get the authenticationcertpolicy_vpnglobal_binding resources in chunks.

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "authenticationcertpolicy_vpnglobal_binding": [ {

"name":<String_value>,

"boundto":<String_value>,

"priority":<Double_value>,

"activepolicy":<Double_value>

}]}
```

get (all)

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/authenticationcertpolicy_vpnglobal_binding

Query-parameters:

bulkbindings

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/authenticationcertpolicy_vpnglobal_binding?**bulkbindings=yes**

NITRO allows you to fetch bindings in bulk.

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "authenticationcertpolicy_vpnglobal_binding": [ {

"name":<String_value>,

"boundto":<String_value>,

"priority":<Double_value>,

"activepolicy":<Double_value>

}]}
```

count

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/authenticationcertpolicy_vpnglobal_binding/name_value<String>?**count=yes**

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{"authenticationcertpolicy_vpnglobal_binding": [ { "__count": "#no"} ] }
```


authenticationcertpolicy_vpnvserver_binding

Binding object showing the vpnvserver that can be bound to authenticationcertpolicy.

Properties

(click to see Operations)

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
name	<String>	Read-write	Name of the client cert authentication policy. Minimum length = 1
boundto	<String>	Read-write	The entity name to which policy is bound.
priority	<Double>	Read-only	.
activepolicy	<Double>	Read-only	.
__count	<Double>	Read-only	count parameter

Operations

(click to see Properties)

GET| GET (ALL)| COUNT

Some options that you can use for each operations:

- **Getting warnings in response:**NITRO allows you to get warnings in an operation by specifying the "warning" query parameter as "yes". For example, to get warnings while connecting to the Citrix ADC appliance, the URL is as follows:

  http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/login?warning=yes

If any, the warnings are displayed in the response payload with the HTTP code "209 X-NITRO-WARNING".

- **Authenticated access for individual NITRO operations:**NITRO allows you to logon to the Citrix ADC appliance to perform individual operations. You can use this option instead of creating a NITRO session (using the login object) and then using that session to perform all operations,

  To do this, you must specify the username and password in the request header of the NITRO request as follows:

  X-NITRO-USER:<username>

  X-NITRO-PASS:<password>

  Note: In such cases, make sure that the request header DOES not include the following:

  Cookie:NITRO_AUTH_TOKEN=<tokenvalue>

`*Note:`*

Mandatory parameters are marked in redand placeholder content is marked in <green>.

get

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/authenticationcertpolicy_vpnvserver_binding/name_value<String>

Query-parameters:

filter

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/authenticationcertpolicy_vpnvserver_binding/name_value<String>?**filter=property-name1:property-value1,property-name2:property-value2**

Use this query-parameter to get the filtered set of authenticationcertpolicy_vpnvserver_binding resources configured on Citrix ADC. Filtering can be done on any of the properties of the resource.

pagination

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/authenticationcertpolicy_vpnvserver_binding/name_value<String>?**pagesize=#no;pageno=#no**

Use this query-parameter to get the authenticationcertpolicy_vpnvserver_binding resources in chunks.

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "authenticationcertpolicy_vpnvserver_binding": [ {

"name":<String_value>,

"boundto":<String_value>,

"priority":<Double_value>,

"activepolicy":<Double_value>

}]}
```

get (all)

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/authenticationcertpolicy_vpnvserver_binding

Query-parameters:

bulkbindings

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/authenticationcertpolicy_vpnvserver_binding?**bulkbindings=yes**

NITRO allows you to fetch bindings in bulk.

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "authenticationcertpolicy_vpnvserver_binding": [ {

"name":<String_value>,

"boundto":<String_value>,

"priority":<Double_value>,

"activepolicy":<Double_value>

}]}
```

count

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/authenticationcertpolicy_vpnvserver_binding/name_value<String>?**count=yes**

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{"authenticationcertpolicy_vpnvserver_binding": [ { "__count": "#no"} ] }
```

authenticationdfaaction

Configuration for Dfa authentication action resource.

Properties

(click to see Operations)

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
name	<String>	Read-write	Name for the DFA action. Must begin with a letter, number, or the underscore character (_), and must contain only letters, numbers, and the hyphen (-), period (.) pound (#), space ( ), at (@), equals (=), colon (:), and underscore characters. Cannot be changed after

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
			the DFA action is added. Minimum length = 1
clientid	<String>	Read-write	If configured, this string is sent to the DFA server as the X-Citrix-Exchange header value.
serverurl	<String>	Read-write	DFA Server URL.
passphrase	<String>	Read-write	Key shared between the DFA server and the Citrix ADC. Required to allow the Citrix ADC to communicate with the DFA server. Minimum length = 1
defaultauthentication group	<String>	Read-write	This is the default group that is chosen when the authentication succeeds in addition to extracted groups.
success	<Double>	Read-only	.
failure	<Double>	Read-only	.
__count	<Double>	Read-only	count parameter

Operations

(click to see Properties)

ADD| DELETE| UPDATE| UNSET| GET (ALL)| GET| COUNT

Some options that you can use for each operations:

- **Getting warnings in response:**NITRO allows you to get warnings in an operation by specifying the "warning" query parameter as "yes". For example, to get warnings while connecting to the Citrix ADC appliance, the URL is as follows:

  http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/login?warning=yes

  If any, the warnings are displayed in the response payload with the HTTP code "209 X-NITRO-WARNING".

- **Authenticated access for individual NITRO operations:**NITRO allows you to logon to the Citrix ADC appliance to perform individual operations. You can use this option instead of creating a NITRO session (using the login object) and then using that session to perform all operations,

  To do this, you must specify the username and password in the request header of the NITRO request as follows:

  X-NITRO-USER:<username>

  X-NITRO-PASS:<password>

  Note: In such cases, make sure that the request header DOES not include the following:

  Cookie:NITRO_AUTH_TOKEN=<tokenvalue>

`*Note:`*

Mandatory parameters are marked in redand placeholder content is marked in <green>.

add

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/authenticationdfaaction

HTTP Method:POST

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Content-Type:application/json

Request Payload:

```
{"authenticationdfaaction":{

<b>"name":<String_value>,

</b><b>"clientid":<String_value>,

</b><b>"serverurl":<String_value>,

</b><b>"passphrase":<String_value>,

</b>"defaultauthenticationgroup":<String_value>

}}
```

Response:

HTTP Status Code on Success: 201 Created HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error

delete

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/authenticationdfaaction/name_value<String>

HTTP Method:DELETE

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue>

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error

update

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/authenticationdfaaction

HTTP Method:PUT

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Content-Type:application/json

Request Payload:

```
{"authenticationdfaaction":{

<b>"name":<String_value>,

</b>"clientid":<String_value>,

"serverurl":<String_value>,

"passphrase":<String_value>,

"defaultauthenticationgroup":<String_value>

}}
```

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error

unset

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/authenticationdfaaction?**action=unset**

HTTP Method:POST

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Content-Type:application/json

Request Payload:

```
{"authenticationdfaaction":{

<b>"name":<String_value>,

</b>"clientid":true,

"serverurl":true,

"defaultauthenticationgroup":true

}}
```

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error

get (all)

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/authenticationdfaaction

Query-parameters:

attrs

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/authenticationdfaaction?**attrs=property-name1,property-name2**

Use this query parameter to specify the resource details that you want to retrieve.

filter

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/authenticationdfaaction?**filter=property-name1:property-val1,property-name2:property-val2**

Use this query-parameter to get the filtered set of authenticationdfaaction resources configured on Citrix ADC. Filtering can be done on any of the properties of the resource.

view

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/authenticationdfaaction?**view=summary**

Note: By default, the retrieved results are displayed in detail view (?view=detail).

pagination

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/authenticationdfaaction?**pagesize=#no;pageno=#no**

Use this query-parameter to get the authenticationdfaaction resources in chunks.

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "authenticationdfaaction": [ {

"name":<String_value>,

"passphrase":<String_value>,

"clientid":<String_value>,

"serverurl":<String_value>,

"success":<Double_value>,

"failure":<Double_value>,

"defaultauthenticationgroup":<String_value>
```

```
}]}
```

get

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/authenticationdfaaction/name_value<String>

Query-parameters:

attrs

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/authenticationdfaaction/name_value<String>?**attrs=property-name1,property-name2**

Use this query parameter to specify the resource details that you want to retrieve.

view

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/authenticationdfaaction/name_value<String>?**view=summary**

Note: By default, the retrieved results are displayed in detail view (?view=detail).

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "authenticationdfaaction": [ {
```

```
"name":<String_value>,

"passphrase":<String_value>,

"clientid":<String_value>,

"serverurl":<String_value>,

"success":<Double_value>,

"failure":<Double_value>,

"defaultauthenticationgroup":<String_value>

}]}
```

count

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/authenticationdfaaction?**count=yes**

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "authenticationdfaaction": [ { "__count": "#no"} ] }
```

authenticationdfapolicy

Configuration for Dfa authentication policy resource.

Properties

(click to see Operations)

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
name	<String>	Read-write	Name for the DFA policy. Must begin with a letter, number, or the underscore character (_), and must contain only letters, numbers, and the hyphen (-), period (.) pound (#), space ( ), at (@), equals (=), colon (:), and underscore characters. Cannot be changed after DFA policy is created. The following requirement applies only to the Citrix ADC CLI: If the name includes one or more spaces, enclose the name in double or single quotation marks (for example, "my authentication policy" or 'my

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
			authentication policy'). Minimum length = 1
rule	<String>	Read-write	Name of the Citrix ADC named rule, or an expression, that the policy uses to determine whether to attempt to authenticate the user with the Web server. Minimum length = 1
action	<String>	Read-write	Name of the DFA action to perform if the policy matches. Minimum length = 1
__count	<Double>	Read-only	count parameter

Operations

(click to see Properties)

ADD| DELETE| UPDATE| GET (ALL)| GET| COUNT

Some options that you can use for each operations:

- **Getting warnings in response:**NITRO allows you to get warnings in an operation by specifying the "warning" query parameter as "yes". For example, to get warnings while connecting to the Citrix ADC appliance, the URL is as follows:

  http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/login?warning=yes

  If any, the warnings are displayed in the response payload with the HTTP code "209 X-NITRO-WARNING".
- **Authenticated access for individual NITRO operations:**NITRO allows you to logon to the Citrix ADC appliance to perform individual operations. You can use this option instead of

creating a NITRO session (using the login object) and then using that session to perform all operations,

To do this, you must specify the username and password in the request header of the NITRO request as follows:

X-NITRO-USER:<username>

X-NITRO-PASS:<password>

Note: In such cases, make sure that the request header DOES not include the following:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue>

`*Note:`*

Mandatory parameters are marked in redand placeholder content is marked in <green>.

add

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/authenticationdfapolicy

HTTP Method:POST

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Content-Type:application/json

Request Payload:

```
{"authenticationdfapolicy":{

<b>"name":<String_value>,

</b><b>"rule":<String_value>,

</b><b>"action":<String_value>

</b>}}
```

Response:

HTTP Status Code on Success: 201 Created HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error

delete

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/authenticationdfapolicy/name_value<String>

HTTP Method:DELETE

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue>

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error

update

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/authenticationdfapolicy

HTTP Method:PUT

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Content-Type:application/json

Request Payload:

```
{"authenticationdfapolicy":{

<b>"name":<String_value>,

</b>"rule":<String_value>,

"action":<String_value>

}}
```

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error

get (all)

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/authenticationdfapolicy

Query-parameters:

attrs

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/authenticationdfapolicy?**attrs=property-name1,property-name2**

Use this query parameter to specify the resource details that you want to retrieve.

filter

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/authenticationdfapolicy?**filter=property-name1:property-val1,property-name2:property-val2**

Use this query-parameter to get the filtered set of authenticationdfapolicy resources configured on Citrix ADC. Filtering can be done on any of the properties of the resource.

view

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/authenticationdfapolicy?**view=summary**

Note: By default, the retrieved results are displayed in detail view (?view=detail).

pagination

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/authenticationdfapolicy?**pagesize=#no;pageno=#no**

Use this query-parameter to get the authenticationdfapolicy resources in chunks.

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "authenticationdfapolicy": [ {

"name":<String_value>,

"rule":<String_value>,

"action":<String_value>

}]}
```

get

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/authenticationdfapolicy/name_value<String>

Query-parameters:

attrs

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/authenticationdfapolicy/name_value<String>?**attrs=property-name1,property-name2**

Use this query parameter to specify the resource details that you want to retrieve.

view

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/authenticationdfapolicy/name_value<String>?**view=summary**

Note: By default, the retrieved results are displayed in detail view (?view=detail).

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "authenticationdfapolicy": [ {

"name":<String_value>,

"rule":<String_value>,

"action":<String_value>

}]}
```

count

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/authenticationdfapolicy?**count=yes**

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "authenticationdfapolicy": [ { "__count": "#no"} ] }
```

authenticationdfapolicy_binding

Binding object which returns the resources bound to authenticationdfapolicy.

Properties

(click to see Operations)

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
name	<String>	Read-write	Name of the WebAuth policy. Minimum length = 1
authenticationdfapolicy_vpnvserver_binding	<authenticationdfapolicy_vpnvserver_binding[]>	Read-only	vpnvserver that can be bound to authenticationdfapolicy.

Operations

(click to see Properties)

GET| GET (ALL)

Some options that you can use for each operations:

- **Getting warnings in response:**NITRO allows you to get warnings in an operation by specifying the "warning" query parameter as "yes". For example, to get warnings while connecting to the Citrix ADC appliance, the URL is as follows:

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/login?warning=yes

If any, the warnings are displayed in the response payload with the HTTP code "209 X-NITRO-WARNING".

- **Authenticated access for individual NITRO operations:**NITRO allows you to logon to the Citrix ADC appliance to perform individual operations. You can use this option instead of creating a NITRO session (using the login object) and then using that session to perform all operations,

  To do this, you must specify the username and password in the request header of the NITRO request as follows:

  X-NITRO-USER:<username>

  X-NITRO-PASS:<password>

  Note: In such cases, make sure that the request header DOES not include the following:

  Cookie:NITRO_AUTH_TOKEN=<tokenvalue>

`*Note:`*

Mandatory parameters are marked in redand placeholder content is marked in <green>.

get

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/authenticationdfapolicy_binding/name_value<String>

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "authenticationdfapolicy_binding": [ {

"name":<String_value>,

"authenticationdfapolicy_vpnvserver_binding":<authenticationdfapolicy_vpnv
server_binding[]_value>

}]}
```

get (all)

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/authenticationdfapolicy_binding

Query-parameters:

bulkbindings

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/authenticationdfapolicy_binding?**bulkbindings=yes**

NITRO allows you to fetch bindings in bulk.

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "authenticationdfapolicy_binding": [ {
```

```
"name":<String_value>,

"authenticationdfapolicy_vpnvserver_binding":<authenticationdfapolicy_vpnv
server_binding[]_value>

}]}
```

authenticationdfapolicy_vpnvserver_binding

Binding object showing the vpnvserver that can be bound to authenticationdfapolicy.

Properties

(click to see Operations)

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
name	<String>	Read-write	Name of the WebAuth policy. Minimum length = 1
boundto	<String>	Read-write	The entity name to which policy is bound.
priority	<Double>	Read-only	.
activepolicy	<Double>	Read-only	.
__count	<Double>	Read-only	count parameter

Operations

(click to see Properties)

GET| GET (ALL)| COUNT

Some options that you can use for each operations:

- **Getting warnings in response:**NITRO allows you to get warnings in an operation by specifying the "warning" query parameter as "yes". For example, to get warnings while connecting to the Citrix ADC appliance, the URL is as follows:

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/login?warning=yes

If any, the warnings are displayed in the response payload with the HTTP code "209 X-NITRO-WARNING".

- **Authenticated access for individual NITRO operations:**NITRO allows you to logon to the Citrix ADC appliance to perform individual operations. You can use this option instead of creating a NITRO session (using the login object) and then using that session to perform all operations,

  To do this, you must specify the username and password in the request header of the NITRO request as follows:

  X-NITRO-USER:<username>

  X-NITRO-PASS:<password>

  Note: In such cases, make sure that the request header DOES not include the following:

  Cookie:NITRO_AUTH_TOKEN=<tokenvalue>

`*Note:`*

Mandatory parameters are marked in redand placeholder content is marked in <green>.

get

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/authenticationdfapolicy_vpnvserver_binding/name_value<String>

Query-parameters:

filter

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/authenticationdfapolicy_vpnvserver_binding/name_value<String>?**filter=property-name1:property-value1,property-name2:property-value2**

Use this query-parameter to get the filtered set of authenticationdfapolicy_vpnvserver_binding resources configured on Citrix ADC. Filtering can be done on any of the properties of the resource.

pagination

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/authenticationdfapolicy_vpnvserver_binding/name_value<String>?**pagesize=#no;pageno=#no**

Use this query-parameter to get the authenticationdfapolicy_vpnvserver_binding resources in chunks.

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "authenticationdfapolicy_vpnvserver_binding": [ {

"name":<String_value>,

"boundto":<String_value>,

"priority":<Double_value>,

"activepolicy":<Double_value>

}]}
```

get (all)

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/authenticationdfapolicy_vpnvserver_binding

Query-parameters:

bulkbindings

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/authenticationdfapolicy_vpnvserver_binding?**bulkbindings=yes**

NITRO allows you to fetch bindings in bulk.

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "authenticationdfapolicy_vpnvserver_binding": [ {

"name":<String_value>,

"boundto":<String_value>,

"priority":<Double_value>,

"activepolicy":<Double_value>

}]}
```

count

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/authenticationdfapolicy_vpnvserver_binding/name_value<String>?**count=yes**

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{"authenticationdfapolicy_vpnvserver_binding": [ { "__count": "#no"} ] }
```

authenticationepaaction

Configuration for epa action resource.

Properties

(click to see Operations)

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
name	<String>	Read-write	Name for the epa action. Must begin with a letter, number, or the underscore character (_), and must consist only of letters, numbers, and the hyphen (-), period (.) pound

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
			(#), space ( ), at (@), equals (=), colon (:), and underscore characters. Cannot be changed after epa action is created.The following requirement applies only to the Citrix ADC CLI:If the name includes one or more spaces, enclose the name in double or single quotation marks (for example, "my aaa action" or 'my aaa action'). Minimum length = 1
csecexpr	<String>	Read-write	it holds the ClientSecurityExpression to be sent to the client.
killprocess	<String>	Read-write	String specifying the name of a process to be terminated by the endpoint analysis (EPA) tool. Multiple processes to be delimited by comma.
deletefiles	<String>	Read-write	String specifying the path(s) and name(s)

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
			of the files to be deleted by the endpoint analysis (EPA) tool. Multiple files to be delimited by comma.
defaultepagroup	<String>	Read-write	This is the default group that is chosen when the EPA check succeeds. Maximum length = 64
quarantinegroup	<String>	Read-write	This is the quarantine group that is chosen when the EPA check fails if configured. Maximum length = 64
__count	<Double>	Read-only	count parameter

Operations

(click to see Properties)

ADD| DELETE| UPDATE| UNSET| GET (ALL)| GET| COUNT

Some options that you can use for each operations:

- **Getting warnings in response:**NITRO allows you to get warnings in an operation by specifying the "warning" query parameter as "yes". For example, to get warnings while connecting to the Citrix ADC appliance, the URL is as follows:

  http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/login?warning=yes

  If any, the warnings are displayed in the response payload with the HTTP code "209 X-NITRO-WARNING".

- **Authenticated access for individual NITRO operations:**NITRO allows you to logon to the Citrix ADC appliance to perform individual operations. You can use this option instead of creating a NITRO session (using the login object) and then using that session to perform all operations,

  To do this, you must specify the username and password in the request header of the NITRO request as follows:

  X-NITRO-USER:<username>

  X-NITRO-PASS:<password>

  Note: In such cases, make sure that the request header DOES not include the following:

  Cookie:NITRO_AUTH_TOKEN=<tokenvalue>

`*Note:`*

Mandatory parameters are marked in redand placeholder content is marked in <green>.

add

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/authenticationepaaction

HTTP Method:POST

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Content-Type:application/json

Request Payload:

```
{"authenticationepaaction":{

<b>"name":<String_value>,

</b><b>"csecexpr":<String_value>,

</b>"killprocess":<String_value>,

"deletefiles":<String_value>,

"defaultepagroup":<String_value>,
```

```
"quarantinegroup":<String_value>

}}
```

Response:

HTTP Status Code on Success: 201 Created HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error

delete

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/authenticationepaaction/name_value<String>

HTTP Method:DELETE

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue>

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error

update

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/authenticationepaaction

HTTP Method:PUT

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Content-Type:application/json

Request Payload:

```
{"authenticationepaaction":{
```

```
<b>"name":<String_value>,

</b>"csecexpr":<String_value>,

"killprocess":<String_value>,

"deletefiles":<String_value>,

"defaultepagroup":<String_value>,

"quarantinegroup":<String_value>

}}
```

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error

unset

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/authenticationepaaction?**action=unset**

HTTP Method:POST

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Content-Type:application/json

Request Payload:

```
{"authenticationepaaction":{

<b>"name":<String_value>,

</b>"killprocess":true,

"deletefiles":true,
```

```
"defaultepagroup":true,

"quarantinegroup":true

}}
```

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error

get (all)

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/authenticationepaaction

Query-parameters:

attrs

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/authenticationepaaction?**attrs=property-name1,property-name2**

Use this query parameter to specify the resource details that you want to retrieve.

filter

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/authenticationepaaction?**filter=property-name1:property-val1,property-name2:property-val2**

Use this query-parameter to get the filtered set of authenticationepaaction resources configured on Citrix ADC. Filtering can be done on any of the properties of the resource.

view

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/authenticationepaaction?**view=summary**

Note: By default, the retrieved results are displayed in detail view (?view=detail).

pagination

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/authenticationepaaction?**pagesize=#no;pageno=#no**

Use this query-parameter to get the authenticationepaaction resources in chunks.

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "authenticationepaaction": [ {

"name":<String_value>,

"killprocess":<String_value>,

"deletefiles":<String_value>,

"defaultepagroup":<String_value>,

"quarantinegroup":<String_value>,

"csecexpr":<String_value>

}]}
```

get

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/authenticationepaaction/name_value<String>

Query-parameters:

attrs

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/authenticationepaaction/name_value<String>?**attrs=property-name1,property-name2**

Use this query parameter to specify the resource details that you want to retrieve.

view

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/authenticationepaaction/name_value<String>?**view=summary**

Note: By default, the retrieved results are displayed in detail view (?view=detail).

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "authenticationepaaction": [ {

"name":<String_value>,

"killprocess":<String_value>,

"deletefiles":<String_value>,

"defaultepagroup":<String_value>,

"quarantinegroup":<String_value>,
```

```
"csecexpr":<String_value>

}]}
```

count

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/authenticationepaaction?**count=yes**

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "authenticationepaaction": [ { "__count": "#no"} ] }
```

authenticationldapaction

Configuration for LDAP action resource.

Properties

(click to see Operations)

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
name	<String>	Read-write	Name for the new LDAP action.

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
			Must begin with a letter, number, or the underscore character (_), and must contain only letters, numbers, and the hyphen (-), period (.) pound (#), space ( ), at (@), equals (=), colon (:), and underscore characters. Cannot be changed after the LDAP action is added. The following requirement applies only to the Citrix ADC CLI: If the name includes one or more spaces, enclose the name in double or single quotation marks (for example, "my authentication action" or 'my authentication action'). Minimum length = 1
serverip	<String>	Read-write	IP address assigned to the LDAP server. Minimum length = 1

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
servername	<String>	Read-write	LDAP server name as a FQDN. Mutually exclusive with LDAP IP address. Minimum length = 1
serverport	<Integer>	Read-write	Port on which the LDAP server accepts connections. Default value: 389 Minimum value = 1
authtimeout	<Double>	Read-write	Number of seconds the Citrix ADC waits for a response from the RADIUS server. Default value: 3 Minimum value = 1
ldapbase	<String>	Read-write	Base (node) from which to start LDAP searches. If the LDAP server is running locally, the default value of base is dc=netscaler, dc=com.
ldapbinddn	<String>	Read-write	Full distinguished name (DN) that is used to bind to the LDAP server. Default: cn=Manager,dc=netscaler,dc=com.

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
ldapbinddnpassword	<String>	Read-write	Password used to bind to the LDAP server. Minimum length = 1
ldaploginname	<String>	Read-write	LDAP login name attribute. The Citrix ADC uses the LDAP login name to query external LDAP servers or Active Directories.
searchfilter	<String>	Read-write	String to be combined with the default LDAP user search string to form the search value. For example, if the search filter "vpnallowed=true" is combined with the LDAP login name "samaccount" and the user-supplied username is "bob", the result is the LDAP search string ""(;(vpnallowed=true)(samaccount=bob)"" (Be sure to enclose the search string in two sets of double quotation marks; both sets are needed.). Minimum length = 1

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
groupattrname	<String>	Read-write	LDAP group attribute name. Used for group extraction on the LDAP server.
subattributename	<String>	Read-write	LDAP group sub-attribute name. Used for group extraction from the LDAP server.
sectype	<String>	Read-write	Type of security used for communications between the Citrix ADC and the LDAP server. For the PLAINTEXT setting, no encryption is required. Default value: PLAINTEXT Possible values = PLAINTEXT, TLS, SSL
svrtype	<String>	Read-write	The type of LDAP server. Default value: AAA_LDAP_SERVER_TYPE_DEFAULT Possible values = AD, NDS
ssonameattribute	<String>	Read-write	LDAP single signon (SSO) attribute.

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
			The Citrix ADC uses the SSO name attribute to query external LDAP servers or Active Directories for an alternate username.
authentication	<String>	Read-write	Perform LDAP authentication. If authentication is disabled, any LDAP authentication attempt returns authentication success if the user is found. CAUTION! Authentication should be disabled only for authorization group extraction or where other (non-LDAP) authentication methods are in use and either bound to a primary list or flagged as secondary. Default value: ENABLED

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
			Possible values = ENABLED, DISABLED
requireuser	<String>	Read-write	Require a successful user search for authentication. CAUTION! This field should be set to NO only if usersearch not required [Both username validation as well as password validation skipped] and (non-LDAP) authentication methods are in use and either bound to a primary list or flagged as secondary. Default value: YES Possible values = YES, NO
passwdchange	<String>	Read-write	Allow password change requests. Default value: DISABLED Possible values = ENABLED, DISABLED

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
nestedgroupextraction	<String>	Read-write	Allow nested group extraction, in which the Citrix ADC queries external LDAP servers to determine whether a group is part of another group. Default value: OFF Possible values = ON, OFF
maxnestinglevel	<Double>	Read-write	If nested group extraction is ON, specifies the number of levels up to which group extraction is performed. Default value: 2 Minimum value = 2
followreferrals	<String>	Read-write	Setting this option to ON enables following LDAP referrals received from the LDAP server. Default value: OFF Possible values = ON, OFF
maxldapreferrals	<Double>	Read-write	Specifies the maximum number of nested referrals to follow. Default value: 1 Minimum value = 1

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
referraldnslookup	<String>	Read-write	Specifies the DNS Record lookup Type for the referrals. Default value: A-REC Possible values = A-REC, SRV-REC, MSSRV-REC
mssrvrecordlocation	<String>	Read-write	MSSRV Specific parameter. Used to locate the DNS node to which the SRV record pertains in the domainname. The domainname is appended to it to form the srv record. Example : For "dc._msdcs", the srv record formed is _ldap._tcp.dc._msdcs.<domainname>.
validateservercert	<String>	Read-write	When to validate LDAP server certs. Default value: NO Possible values = YES, NO
ldaphostname	<String>	Read-write	Hostname for the LDAP server. If -validateServerCert is ON then this must be the host name on the certificate from the LDAP server.

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
			A hostname mismatch will cause a connection failure.
groupnameidentifier	<String>	Read-write	Name that uniquely identifies a group in LDAP or Active Directory.
groupsearchattribute	<String>	Read-write	LDAP group search attribute. Used to determine to which groups a group belongs.
groupsearchsubattribute	<String>	Read-write	LDAP group search subattribute. Used to determine to which groups a group belongs.
groupsearchfilter	<String>	Read-write	String to be combined with the default LDAP group search string to form the search value. For example, the group search filter ""vpnallowed=true"" when combined with the group identifier ""samaccount"" and the group name ""g1"" yields the LDAP search string ""(;(vpnallowed=true)(samaccount=g1)"". (Be sure to enclose the search string in two sets of double

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
			quotation marks; both sets are needed.).
defaultauthentication group	<String>	Read-write	This is the default group that is chosen when the authentication succeeds in addition to extracted groups.
attribute1	<String>	Read-write	Expression that would be evaluated to extract attribute1 from the ldap response.
attribute2	<String>	Read-write	Expression that would be evaluated to extract attribute2 from the ldap response.
attribute3	<String>	Read-write	Expression that would be evaluated to extract attribute3 from the ldap response.
attribute4	<String>	Read-write	Expression that would be evaluated to extract attribute4 from the ldap response.
attribute5	<String>	Read-write	Expression that would be evaluated to extract attribute5 from the ldap response.

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
attribute6	<String>	Read-write	Expression that would be evaluated to extract attribute6 from the ldap response.
attribute7	<String>	Read-write	Expression that would be evaluated to extract attribute7 from the ldap response.
attribute8	<String>	Read-write	Expression that would be evaluated to extract attribute8 from the ldap response.
attribute9	<String>	Read-write	Expression that would be evaluated to extract attribute9 from the ldap response.
attribute10	<String>	Read-write	Expression that would be evaluated to extract attribute10 from the ldap response.
attribute11	<String>	Read-write	Expression that would be evaluated to extract attribute11 from the ldap response.
attribute12	<String>	Read-write	Expression that would be evaluated to extract attribute12 from the ldap response.

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
attribute13	<String>	Read-write	Expression that would be evaluated to extract attribute13 from the ldap response.
attribute14	<String>	Read-write	Expression that would be evaluated to extract attribute14 from the ldap response.
attribute15	<String>	Read-write	Expression that would be evaluated to extract attribute15 from the ldap response.
attribute16	<String>	Read-write	Expression that would be evaluated to extract attribute16 from the ldap response.
attributes	<String>	Read-write	List of attribute names separated by ',' which needs to be fetched from ldap server. Note that preceeding and trailing spaces will be removed. Attribute name can be 127 bytes and total length of this string should not cross 2047 bytes.

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
			These attributes have multi-value support separated by ',' and stored as key-value pair in AAA session.
sshpublickey	<String>	Read-write	SSH PublicKey is attribute on AD. This attribute is used to retrieve ssh PublicKey for RBA authentication. Minimum length = 1
pushservice	<String>	Read-write	Name of the service used to send push notifications. Minimum length = 1
otpsecret	<String>	Read-write	OneTimePassword(OTP) Secret key attribute on AD. This attribute is used to store and retrieve secret key used for OTP check. Minimum length = 1
email	<String>	Read-write	The Citrix ADC uses the email attribute to query the Active Directory for the email id of a user. Default value: mail Maximum length = 128

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
kbattribute	<String>	Read-write	KnowledgeBasedAuthentication(KBA) attribute on AD. This attribute is used to store and retrieve preconfigured Question and Answer knowledge base used for KBA authentication. Minimum length = 1 Maximum length = 127
alternateemailattr	<String>	Read-write	The Citrix ADC appliance uses the alternateive email attribute to query the Active Directory for the alternative email id of a user. Maximum length = 128
success	<Double>	Read-only	.
failure	<Double>	Read-only	.
__count	<Double>	Read-only	count parameter

Operations

(click to see Properties)

ADD| DELETE| UPDATE| UNSET| GET (ALL)| GET| COUNT

Some options that you can use for each operations:

- **Getting warnings in response:**NITRO allows you to get warnings in an operation by specifying the "warning" query parameter as "yes". For example, to get warnings while connecting to the Citrix ADC appliance, the URL is as follows:

  http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/login?warning=yes

  If any, the warnings are displayed in the response payload with the HTTP code "209 X-NITRO-WARNING".

- **Authenticated access for individual NITRO operations:**NITRO allows you to logon to the Citrix ADC appliance to perform individual operations. You can use this option instead of creating a NITRO session (using the login object) and then using that session to perform all operations,

  To do this, you must specify the username and password in the request header of the NITRO request as follows:

  X-NITRO-USER:<username>

  X-NITRO-PASS:<password>

  Note: In such cases, make sure that the request header DOES not include the following:

  Cookie:NITRO_AUTH_TOKEN=<tokenvalue>

`*Note:`*

Mandatory parameters are marked in redand placeholder content is marked in <green>.

add

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/authenticationldapaction

HTTP Method:POST

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Content-Type:application/json

Request Payload:

```
{"authenticationldapaction":{
```

```
<b>"name":<String_value>,

</b>"serverip":<String_value>,

"servername":<String_value>,

"serverport":<Integer_value>,

"authtimeout":<Double_value>,

"ldapbase":<String_value>,

"ldapbinddn":<String_value>,

"ldapbinddnpassword":<String_value>,

"ldaploginname":<String_value>,

"searchfilter":<String_value>,

"groupattrname":<String_value>,

"subattributename":<String_value>,

"sectype":<String_value>,

"svrtype":<String_value>,

"ssonameattribute":<String_value>,

"authentication":<String_value>,

"requireuser":<String_value>,

"passwdchange":<String_value>,

"nestedgroupextraction":<String_value>,

"maxnestinglevel":<Double_value>,

"followreferrals":<String_value>,

"maxldapreferrals":<Double_value>,
```

```
"referraldnslookup":<String_value>,

"mssrvrecordlocation":<String_value>,

"validateservercert":<String_value>,

"ldaphostname":<String_value>,

"groupnameidentifier":<String_value>,

"groupsearchattribute":<String_value>,

"groupsearchsubattribute":<String_value>,

"groupsearchfilter":<String_value>,

"defaultauthenticationgroup":<String_value>,

"attribute1":<String_value>,

"attribute2":<String_value>,

"attribute3":<String_value>,

"attribute4":<String_value>,

"attribute5":<String_value>,

"attribute6":<String_value>,

"attribute7":<String_value>,

"attribute8":<String_value>,

"attribute9":<String_value>,

"attribute10":<String_value>,

"attribute11":<String_value>,

"attribute12":<String_value>,

"attribute13":<String_value>,
```

```
"attribute14":<String_value>,

"attribute15":<String_value>,

"attribute16":<String_value>,

"attributes":<String_value>,

"sshpublickey":<String_value>,

"pushservice":<String_value>,

"otpsecret":<String_value>,

"email":<String_value>,

"kbattribute":<String_value>,

"alternateemailattr":<String_value>

}}
```

Response:

HTTP Status Code on Success: 201 Created HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error

delete

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/authenticationldapaction/name_value<String>

HTTP Method:DELETE

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue>

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error

update

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/authenticationldapaction

HTTP Method:PUT

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Content-Type:application/json

Request Payload:

```
{"authenticationldapaction":{

<b>"name":<String_value>,

</b>"serverip":<String_value>,

"servername":<String_value>,

"serverport":<Integer_value>,

"authtimeout":<Double_value>,

"ldapbase":<String_value>,

"ldapbinddn":<String_value>,

"ldapbinddnpassword":<String_value>,

"ldaploginname":<String_value>,

"searchfilter":<String_value>,

"groupattrname":<String_value>,

"subattributename":<String_value>,
```

```
"sectype":<String_value>,

"svrtype":<String_value>,

"ssonameattribute":<String_value>,

"authentication":<String_value>,

"requireuser":<String_value>,

"passwdchange":<String_value>,

"validateservercert":<String_value>,

"ldaphostname":<String_value>,

"nestedgroupextraction":<String_value>,

"maxnestinglevel":<Double_value>,

"groupnameidentifier":<String_value>,

"groupsearchattribute":<String_value>,

"groupsearchsubattribute":<String_value>,

"groupsearchfilter":<String_value>,

"followreferrals":<String_value>,

"maxldapreferrals":<Double_value>,

"referraldnslookup":<String_value>,

"mssrvrecordlocation":<String_value>,

"defaultauthenticationgroup":<String_value>,

"attribute1":<String_value>,

"attribute2":<String_value>,

"attribute3":<String_value>,
```

```
"attribute4":<String_value>,

"attribute5":<String_value>,

"attribute6":<String_value>,

"attribute7":<String_value>,

"attribute8":<String_value>,

"attribute9":<String_value>,

"attribute10":<String_value>,

"attribute11":<String_value>,

"attribute12":<String_value>,

"attribute13":<String_value>,

"attribute14":<String_value>,

"attribute15":<String_value>,

"attribute16":<String_value>,

"attributes":<String_value>,

"otpsecret":<String_value>,

"sshpublickey":<String_value>,

"pushservice":<String_value>,

"email":<String_value>,

"kbattribute":<String_value>,

"alternateemailattr":<String_value>

}}
```

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error

unset

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/authenticationldapaction?**action=unset**

HTTP Method:POST

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Content-Type:application/json

Request Payload:

```
{"authenticationldapaction":{

<b>"name":<String_value>,

</b>"serverport":true,

"authtimeout":true,

"ldapbase":true,

"ldapbinddn":true,

"ldapbinddnpassword":true,

"ldaploginname":true,

"searchfilter":true,

"groupattrname":true,

"subattributename":true,
```

```
"sectype":true,

"svrtype":true,

"ssonameattribute":true,

"authentication":true,

"requireuser":true,

"passwdchange":true,

"validateservercert":true,

"ldaphostname":true,

"nestedgroupextraction":true,

"maxnestinglevel":true,

"groupnameidentifier":true,

"groupsearchattribute":true,

"groupsearchsubattribute":true,

"groupsearchfilter":true,

"followreferrals":true,

"maxldapreferrals":true,

"referraldnslookup":true,

"mssrvrecordlocation":true,

"defaultauthenticationgroup":true,

"attribute1":true,

"attribute2":true,

"attribute3":true,
```

```
"attribute4":true,

"attribute5":true,

"attribute6":true,

"attribute7":true,

"attribute8":true,

"attribute9":true,

"attribute10":true,

"attribute11":true,

"attribute12":true,

"attribute13":true,

"attribute14":true,

"attribute15":true,

"attribute16":true,

"attributes":true,

"otpsecret":true,

"pushservice":true,

"email":true,

"kbattribute":true,

"alternateemailattr":true

}}
```


Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error

get (all)

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/authenticationldapaction

Query-parameters:

attrs

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/authenticationldapaction?**attrs=property-name1,property-name2**

Use this query parameter to specify the resource details that you want to retrieve.

filter

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/authenticationldapaction?**filter=property-name1:property-val1,property-name2:property-val2**

Use this query-parameter to get the filtered set of authenticationldapaction resources configured on Citrix ADC. Filtering can be done on any of the properties of the resource.

view

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/authenticationldapaction?**view=summary**

Note: By default, the retrieved results are displayed in detail view (?view=detail).

pagination

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/authenticationldapaction?**pagesize=#no;pageno=#no**

Use this query-parameter to get the authenticationldapaction resources in chunks.

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "authenticationldapaction": [ {

"name":<String_value>,

"serverip":<String_value>,

"servername":<String_value>,

"serverport":<Integer_value>,

"authtimeout":<Double_value>,

"ldapbinddn":<String_value>,

"ldapbinddnpassword":<String_value>,

"ldaploginname":<String_value>,

"ldapbase":<String_value>,

"searchfilter":<String_value>,

"groupattrname":<String_value>,

"subattributename":<String_value>,

"sectype":<String_value>,

"svrtype":<String_value>,

"ssonameattribute":<String_value>,

"authentication":<String_value>,
```

```
"requireuser":<String_value>,

"success":<Double_value>,

"failure":<Double_value>,

"nestedgroupextraction":<String_value>,

"maxnestinglevel":<Double_value>,

"followreferrals":<String_value>,

"maxldapreferrals":<Double_value>,

"referraldnslookup":<String_value>,

"mssrvrecordlocation":<String_value>,

"validateservercert":<String_value>,

"ldaphostname":<String_value>,

"groupnameidentifier":<String_value>,

"groupsearchattribute":<String_value>,

"groupsearchsubattribute":<String_value>,

"groupsearchfilter":<String_value>,

"passwdchange":<String_value>,

"defaultauthenticationgroup":<String_value>,

"attribute1":<String_value>,

"attribute2":<String_value>,

"attribute3":<String_value>,

"attribute4":<String_value>,

"attribute5":<String_value>,
```

```
"attribute6":<String_value>,

"attribute7":<String_value>,

"attribute8":<String_value>,

"attribute9":<String_value>,

"attribute10":<String_value>,

"attribute11":<String_value>,

"attribute12":<String_value>,

"attribute13":<String_value>,

"attribute14":<String_value>,

"attribute15":<String_value>,

"attribute16":<String_value>,

"attributes":<String_value>,

"otpsecret":<String_value>,

"sshpublickey":<String_value>,

"pushservice":<String_value>,

"email":<String_value>,

"kbattribute":<String_value>,

"alternateemailattr":<String_value>

}]}
```

get

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/authenticationldapaction/name_value<String>

Query-parameters:

attrs

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/authenticationldapaction/name_value<String>?**attrs=property-name1,property-name2**

Use this query parameter to specify the resource details that you want to retrieve.

view

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/authenticationldapaction/name_value<String>?**view=summary**

Note: By default, the retrieved results are displayed in detail view (?view=detail).

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "authenticationldapaction": [ {

"name":<String_value>,

"serverip":<String_value>,

"servername":<String_value>,

"serverport":<Integer_value>,

"authtimeout":<Double_value>,
```

```
"ldapbinddn":<String_value>,

"ldapbinddnpassword":<String_value>,

"ldaploginname":<String_value>,

"ldapbase":<String_value>,

"searchfilter":<String_value>,

"groupattrname":<String_value>,

"subattributename":<String_value>,

"sectype":<String_value>,

"svrtype":<String_value>,

"ssonameattribute":<String_value>,

"authentication":<String_value>,

"requireuser":<String_value>,

"success":<Double_value>,

"failure":<Double_value>,

"nestedgroupextraction":<String_value>,

"maxnestinglevel":<Double_value>,

"followreferrals":<String_value>,

"maxldapreferrals":<Double_value>,

"referraldnslookup":<String_value>,

"mssrvrecordlocation":<String_value>,

"validateservercert":<String_value>,

"ldaphostname":<String_value>,
```

```
"groupnameidentifier":<String_value>,

"groupsearchattribute":<String_value>,

"groupsearchsubattribute":<String_value>,

"groupsearchfilter":<String_value>,

"passwdchange":<String_value>,

"defaultauthenticationgroup":<String_value>,

"attribute1":<String_value>,

"attribute2":<String_value>,

"attribute3":<String_value>,

"attribute4":<String_value>,

"attribute5":<String_value>,

"attribute6":<String_value>,

"attribute7":<String_value>,

"attribute8":<String_value>,

"attribute9":<String_value>,

"attribute10":<String_value>,

"attribute11":<String_value>,

"attribute12":<String_value>,

"attribute13":<String_value>,

"attribute14":<String_value>,

"attribute15":<String_value>,

"attribute16":<String_value>,
```

```
"attributes":<String_value>,

"otpsecret":<String_value>,

"sshpublickey":<String_value>,

"pushservice":<String_value>,

"email":<String_value>,

"kbattribute":<String_value>,

"alternateemailattr":<String_value>

}]}
```

count

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/authenticationldapaction?**count=yes**

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "authenticationldapaction": [ { "__count": "#no"} ] }
```

authenticationldappolicy

Configuration for LDAP policy resource.

Properties

(click to see Operations)

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
name	<String>	Read-write	Name for the LDAP policy. Must begin with a letter, number, or the underscore character (_), and must contain only letters, numbers, and the hyphen (-), period (.) pound (#), space ( ), at (@), equals (=), colon (:), and underscore characters. Cannot be changed after LDAP policy is created. The following requirement applies only to the Citrix ADC CLI: If the name includes one or more spaces, enclose the name in double or single quotation marks (for example, "my authentication policy" or 'my

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
			authentication policy'). Minimum length = 1
rule	<String>	Read-write	Name of the Citrix ADC named rule, or an expression, that the policy uses to determine whether to attempt to authenticate the user with the LDAP server. Minimum length = 1
reqaction	<String>	Read-write	Name of the LDAP action to perform if the policy matches. Minimum length = 1
__count	<Double>	Read-only	count parameter

Operations

(click to see Properties)

ADD| DELETE| UPDATE| UNSET| GET (ALL)| GET| COUNT

Some options that you can use for each operations:

- **Getting warnings in response:**NITRO allows you to get warnings in an operation by specifying the "warning" query parameter as "yes". For example, to get warnings while connecting to the Citrix ADC appliance, the URL is as follows:

  http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/login?warning=yes

  If any, the warnings are displayed in the response payload with the HTTP code "209 X-NITRO-WARNING".

- **Authenticated access for individual NITRO operations:**NITRO allows you to logon to the Citrix ADC appliance to perform individual operations. You can use this option instead of creating a NITRO session (using the login object) and then using that session to perform all operations,

  To do this, you must specify the username and password in the request header of the NITRO request as follows:

  X-NITRO-USER:<username>

  X-NITRO-PASS:<password>

  Note: In such cases, make sure that the request header DOES not include the following:

  Cookie:NITRO_AUTH_TOKEN=<tokenvalue>

`*Note:`*

Mandatory parameters are marked in redand placeholder content is marked in <green>.

add

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/authenticationldappolicy

HTTP Method:POST

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Content-Type:application/json

Request Payload:

```
{"authenticationldappolicy":{

<b>"name":<String_value>,

</b><b>"rule":<String_value>,

</b>"reqaction":<String_value>

}}
```

Response:

HTTP Status Code on Success: 201 Created HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error

delete

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/authenticationldappolicy/name_value<String>

HTTP Method:DELETE

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue>

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error

update

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/authenticationldappolicy

HTTP Method:PUT

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Content-Type:application/json

Request Payload:

```
{"authenticationldappolicy":{

<b>"name":<String_value>,

</b>"rule":<String_value>,

"reqaction":<String_value>

}}
```

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error

unset

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/authenticationldappolicy?**action=unset**

HTTP Method:POST

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Content-Type:application/json

Request Payload:

```
{"authenticationldappolicy":{

<b>"name":<String_value>,

</b>"rule":true,

"reqaction":true

}}
```

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error

get (all)

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/authenticationldappolicy

Query-parameters:

attrs

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/authenticationldappolicy?**attrs=property-name1,property-name2**

Use this query parameter to specify the resource details that you want to retrieve.

filter

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/authenticationldappolicy?**filter=property-name1:property-val1,property-name2:property-val2**

Use this query-parameter to get the filtered set of authenticationldappolicy resources configured on Citrix ADC. Filtering can be done on any of the properties of the resource.

view

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/authenticationldappolicy?**view=summary**

Note: By default, the retrieved results are displayed in detail view (?view=detail).

pagination

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/authenticationldappolicy?**pagesize=#no;pageno=#no**

Use this query-parameter to get the authenticationldappolicy resources in chunks.

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "authenticationldappolicy": [ {

"name":<String_value>,

"rule":<String_value>,

"reqaction":<String_value>

}]}
```

get

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/authenticationldappolicy/name_value<String>

Query-parameters:

attrs

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/authenticationldappolicy/name_value<String>?**attrs=property-name1,property-name2**

Use this query parameter to specify the resource details that you want to retrieve.

view

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/authenticationldappolicy/name_value<String>?**view=summary**

Note: By default, the retrieved results are displayed in detail view (?view=detail).

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "authenticationldappolicy": [ {

"name":<String_value>,

"rule":<String_value>,

"reqaction":<String_value>

}]}
```

count

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/authenticationldappolicy?**count=yes**

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "authenticationldappolicy": [ { "__count": "#no"} ] }
```

authenticationldappolicy_authenticationvserver_binding

Binding object showing the authenticationvserver that can be bound to authenticationldappolicy.

Properties

(click to see Operations)

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
boundto	<String>	Read-write	The entity name to which policy is bound.
name	<String>	Read-write	Name of the LDAP policy. Minimum length = 1
priority	<Double>	Read-only	.
activepolicy	<Double>	Read-only	.
__count	<Double>	Read-only	count parameter

Operations

(click to see Properties)

GET| GET (ALL)| COUNT

Some options that you can use for each operations:

- **Getting warnings in response:**NITRO allows you to get warnings in an operation by specifying the "warning" query parameter as "yes". For example, to get warnings while connecting to the Citrix ADC appliance, the URL is as follows:

  http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/login?warning=yes

If any, the warnings are displayed in the response payload with the HTTP code "209 X-NITRO-WARNING".

- **Authenticated access for individual NITRO operations:**NITRO allows you to logon to the Citrix ADC appliance to perform individual operations. You can use this option instead of creating a NITRO session (using the login object) and then using that session to perform all operations,

  To do this, you must specify the username and password in the request header of the NITRO request as follows:

  X-NITRO-USER:<username>

  X-NITRO-PASS:<password>

  Note: In such cases, make sure that the request header DOES not include the following:

  Cookie:NITRO_AUTH_TOKEN=<tokenvalue>

`*Note:`*

Mandatory parameters are marked in redand placeholder content is marked in <green>.

get

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/authenticationldappolicy_authenticationvserver_binding/name_value<String>

Query-parameters:

filter

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/authenticationldappolicy_authenticationvserver_binding/name_value<String>?**filter=property-name1:property-value1,property-name2:property-value2**

Use this query-parameter to get the filtered set of authenticationldappolicy_authenticationvserver_binding resources configured on Citrix ADC. Filtering can be done on any of the properties of the resource.

pagination

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/authenticationldappolicy_authenticationvserver_binding/name_value<String>?**pagesize=#no;pageno=#no**

Use this query-parameter to get the authenticationldappolicy_authenticationvserver_binding resources in chunks.

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "authenticationldappolicy_authenticationvserver_binding": [ {

"boundto":<String_value>,

"name":<String_value>,

"priority":<Double_value>,

"activepolicy":<Double_value>

}]}
```

get (all)

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/authenticationldappolicy_authenticationvserver_binding

Query-parameters:

bulkbindings

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/authenticationldappolicy_authenticationvserver_binding?**bulkbindings=yes**

NITRO allows you to fetch bindings in bulk.

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "authenticationldappolicy_authenticationvserver_binding": [ {

"boundto":<String_value>,

"name":<String_value>,

"priority":<Double_value>,

"activepolicy":<Double_value>

}]}
```

count

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/authenticationldappolicy_authenticationvserver_binding/name_value<String>?**count=yes**

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{"authenticationldappolicy_authenticationvserver_binding": [ { "__count":
"#no"} ] }
```

authenticationldappolicy_binding

Binding object which returns the resources bound to authenticationldappolicy.

Properties

(click to see Operations)

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
name	<String>	Read-write	Name of the LDAP policy. Minimum length = 1
authenticationldappolicy_vpnglobal_binding	<authenticationldappolicy_vpnglobal_binding[]>	Read-only	vpnglobal that can be bound to authenticationldappolicy.
authenticationldappolicy_vpnvserver_binding	<authenticationldappolicy_vpnvserver_binding[]>	Read-only	vpnvserver that can be bound to authenticationldappolicy.

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
authenticationldappolicy_systemglobal_binding	<authenticationldappolicy_systemglobal_binding[]>	Read-only	systemglobal that can be bound to authenticationldappolicy.
authenticationldappolicy_authenticationvserver_binding	<authenticationldappolicy_authenticationvserver_binding[]>	Read-only	authenticationvserver that can be bound to authenticationldappolicy.

Operations

(click to see Properties)

GET| GET (ALL)

Some options that you can use for each operations:

- **Getting warnings in response:**NITRO allows you to get warnings in an operation by specifying the "warning" query parameter as "yes". For example, to get warnings while connecting to the Citrix ADC appliance, the URL is as follows:

  http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/login?warning=yes

  If any, the warnings are displayed in the response payload with the HTTP code "209 X-NITRO-WARNING".

- **Authenticated access for individual NITRO operations:**NITRO allows you to logon to the Citrix ADC appliance to perform individual operations. You can use this option instead of creating a NITRO session (using the login object) and then using that session to perform all operations,

  To do this, you must specify the username and password in the request header of the NITRO request as follows:

  X-NITRO-USER:<username>

  X-NITRO-PASS:<password>

  Note: In such cases, make sure that the request header DOES not include the following:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue>

`*Note:`*

Mandatory parameters are marked in redand placeholder content is marked in <green>.

get

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/authenticationldappolicy_binding/name_value<String>

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "authenticationldappolicy_binding": [ {

"name":<String_value>,

"authenticationldappolicy_systemglobal_binding":<authenticationldappolicy_systemglobal_binding[]_value>,

"authenticationldappolicy_vpnvserver_binding":<authenticationldappolicy_vpnvserver_binding[]_value>,

"authenticationldappolicy_authenticationvserver_binding":<authenticationldappolicy_authenticationvserver_binding[]_value>,

"authenticationldappolicy_vpnglobal_binding":<authenticationldappolicy_vpnglobal_binding[]_value>
```

```
}]}
```

get (all)

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/authenticationldappolicy_binding

Query-parameters:

bulkbindings

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/authenticationldappolicy_binding?**bulkbindings=yes**

NITRO allows you to fetch bindings in bulk.

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "authenticationldappolicy_binding": [ {

"name":<String_value>,

"authenticationldappolicy_systemglobal_binding":<authenticationldappolicy_
systemglobal_binding[]_value>,

"authenticationldappolicy_vpnvserver_binding":<authenticationldappolicy_vp
nvserver_binding[]_value>,
```

```
"authenticationldappolicy_authenticationvserver_binding":<authenticationldappolicy_authenticationvserver_binding[]_value>,

"authenticationldappolicy_vpnglobal_binding":<authenticationldappolicy_vpnglobal_binding[]_value>

}]}
```

authenticationldappolicy_systemglobal_binding

Binding object showing the systemglobal that can be bound to authenticationldappolicy.

Properties

(click to see Operations)

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
boundto	<String>	Read-write	The entity name to which policy is bound.
name	<String>	Read-write	Name of the LDAP policy. Minimum length = 1
priority	<Double>	Read-only	.
activepolicy	<Double>	Read-only	.
__count	<Double>	Read-only	count parameter

Operations

(click to see Properties)

GET| GET (ALL)| COUNT

Some options that you can use for each operations:

- **Getting warnings in response:**NITRO allows you to get warnings in an operation by specifying the "warning" query parameter as "yes". For example, to get warnings while connecting to the Citrix ADC appliance, the URL is as follows:

  http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/login?warning=yes

  If any, the warnings are displayed in the response payload with the HTTP code "209 X-NITRO-WARNING".

- **Authenticated access for individual NITRO operations:**NITRO allows you to logon to the Citrix ADC appliance to perform individual operations. You can use this option instead of creating a NITRO session (using the login object) and then using that session to perform all operations,

  To do this, you must specify the username and password in the request header of the NITRO request as follows:

  X-NITRO-USER:<username>

  X-NITRO-PASS:<password>

  Note: In such cases, make sure that the request header DOES not include the following:

  Cookie:NITRO_AUTH_TOKEN=<tokenvalue>

`*Note:`*

Mandatory parameters are marked in redand placeholder content is marked in <green>.

get

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/authenticationldappolicy_systemglobal_binding/name_value<String>

Query-parameters:

filter

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/authenticationldappolicy_systemglobal_binding/name_value<String>?**filter=property-name1:property-value1,property-name2:property-value2**

Use this query-parameter to get the filtered set of authenticationldappolicy_systemglobal_binding resources configured on Citrix ADC. Filtering can be done on any of the properties of the resource.

pagination

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/authenticationldappolicy_systemglobal_binding/name_value<String>?**pagesize=#no;pageno=#no**

Use this query-parameter to get the authenticationldappolicy_systemglobal_binding resources in chunks.

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "authenticationldappolicy_systemglobal_binding": [ {

"boundto":<String_value>,

"name":<String_value>,

"priority":<Double_value>,

"activepolicy":<Double_value>

}]}
```

get (all)

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/authenticationldappolicy_systemglobal_binding

Query-parameters:

bulkbindings

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/authenticationldappolicy_systemglobal_binding?**bulkbindings=yes**

NITRO allows you to fetch bindings in bulk.

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "authenticationldappolicy_systemglobal_binding": [ {

"boundto":<String_value>,

"name":<String_value>,

"priority":<Double_value>,

"activepolicy":<Double_value>

}]}
```

count

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/authenticationldappolicy_systemglobal_binding/name_value<String>?**count=yes**

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{"authenticationldappolicy_systemglobal_binding": [ { "__count":
"#no"} ] }
```

authenticationldappolicy_vpnglobal_binding

Binding object showing the vpnglobal that can be bound to authenticationldappolicy.

Properties

(click to see Operations)

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
boundto	<String>	Read-write	The entity name to which policy is bound.
name	<String>	Read-write	Name of the LDAP policy. Minimum length = 1
priority	<Double>	Read-only	.
activepolicy	<Double>	Read-only	.

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
__count	<Double>	Read-only	count parameter

Operations

(click to see Properties)

GET| GET (ALL)| COUNT

Some options that you can use for each operations:

- **Getting warnings in response:**NITRO allows you to get warnings in an operation by specifying the "warning" query parameter as "yes". For example, to get warnings while connecting to the Citrix ADC appliance, the URL is as follows:

  http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/login?warning=yes

  If any, the warnings are displayed in the response payload with the HTTP code "209 X-NITRO-WARNING".
- **Authenticated access for individual NITRO operations:**NITRO allows you to logon to the Citrix ADC appliance to perform individual operations. You can use this option instead of creating a NITRO session (using the login object) and then using that session to perform all operations,

  To do this, you must specify the username and password in the request header of the NITRO request as follows:

  X-NITRO-USER:<username>

  X-NITRO-PASS:<password>

  Note: In such cases, make sure that the request header DOES not include the following:

  Cookie:NITRO_AUTH_TOKEN=<tokenvalue>

`*Note:`*

Mandatory parameters are marked in redand placeholder content is marked in <green>.

get

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/authenticationldappolicy_vpnglobal_binding/name_value<String>

Query-parameters:

filter

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/authenticationldappolicy_vpnglobal_binding/name_value<String>?**filter=property-name1:property-value1,property-name2:property-value2**

Use this query-parameter to get the filtered set of authenticationldappolicy_vpnglobal_binding resources configured on Citrix ADC. Filtering can be done on any of the properties of the resource.

pagination

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/authenticationldappolicy_vpnglobal_binding/name_value<String>?**pagesize=#no;pageno=#no**

Use this query-parameter to get the authenticationldappolicy_vpnglobal_binding resources in chunks.

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "authenticationldappolicy_vpnglobal_binding": [ {

"boundto":<String_value>,

"name":<String_value>,

"priority":<Double_value>,
```

```
"activepolicy":<Double_value>

}]}
```

get (all)

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/authenticationldappolicy_vpnglobal_binding

Query-parameters:

bulkbindings

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/authenticationldappolicy_vpnglobal_binding?**bulkbindings=yes**

NITRO allows you to fetch bindings in bulk.

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "authenticationldappolicy_vpnglobal_binding": [ {

"boundto":<String_value>,

"name":<String_value>,

"priority":<Double_value>,
```

```
"activepolicy":<Double_value>

}]}
```

count

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/authenticationldappolicy_vpnglobal_binding/name_value<String>?**count=yes**

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{"authenticationldappolicy_vpnglobal_binding": [ { "__count": "#no"} ] }
```

authenticationldappolicy_vpnvserver_binding

Binding object showing the vpnvserver that can be bound to authenticationldappolicy.

Properties

(click to see Operations)

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
boundto	<String>	Read-write	The entity name to which policy is bound.
name	<String>	Read-write	Name of the LDAP policy. Minimum length = 1
priority	<Double>	Read-only	.
activepolicy	<Double>	Read-only	.
__count	<Double>	Read-only	count parameter

Operations

(click to see Properties)

GET| GET (ALL)| COUNT

Some options that you can use for each operations:

- **Getting warnings in response:**NITRO allows you to get warnings in an operation by specifying the "warning" query parameter as "yes". For example, to get warnings while connecting to the Citrix ADC appliance, the URL is as follows:

  http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/login?warning=yes

  If any, the warnings are displayed in the response payload with the HTTP code "209 X-NITRO-WARNING".

- **Authenticated access for individual NITRO operations:**NITRO allows you to logon to the Citrix ADC appliance to perform individual operations. You can use this option instead of creating a NITRO session (using the login object) and then using that session to perform all operations,

  To do this, you must specify the username and password in the request header of the NITRO request as follows:

  X-NITRO-USER:<username>

  X-NITRO-PASS:<password>

Note: In such cases, make sure that the request header DOES not include the following:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue>

`*Note:`*

Mandatory parameters are marked in redand placeholder content is marked in <green>.

get

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/authenticationldappolicy_vpnvserver_binding/name_value<String>

Query-parameters:

filter

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/authenticationldappolicy_vpnvserver_binding/name_value<String>?**filter=property-name1:property-value1,property-name2:property-value2**

Use this query-parameter to get the filtered set of authenticationldappolicy_vpnvserver_binding resources configured on Citrix ADC. Filtering can be done on any of the properties of the resource.

pagination

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/authenticationldappolicy_vpnvserver_binding/name_value<String>?**pagesize=#no;pageno=#no**

Use this query-parameter to get the authenticationldappolicy_vpnvserver_binding resources in chunks.

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "authenticationldappolicy_vpnvserver_binding": [ {

"boundto":<String_value>,

"name":<String_value>,

"priority":<Double_value>,

"activepolicy":<Double_value>

}]}
```

get (all)

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/authenticationldappolicy_vpnvserver_binding

Query-parameters:

bulkbindings

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/authenticationldappolicy_vpnvserver_binding?**bulkbindings=yes**

NITRO allows you to fetch bindings in bulk.

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "authenticationldappolicy_vpnvserver_binding": [ {

"boundto":<String_value>,

"name":<String_value>,

"priority":<Double_value>,

"activepolicy":<Double_value>

}]}
```

count

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/authenticationldappolicy_vpnvserver_binding/name_value<String>?**count=yes**

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{"authenticationldappolicy_vpnvserver_binding": [ { "__count": "#no"} ] }
```

authenticationlocalpolicy

Configuration for LOCAL policy resource.

Properties

(click to see Operations)

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
name	<String>	Read-write	Name for the local authentication policy. Must begin with a letter, number, or the underscore character (_), and must contain only letters, numbers, and the hyphen (-), period (.) pound (#), space ( ), at (@), equals (=), colon (:), and underscore characters. Cannot be changed after local policy is created. The following requirement applies only to the Citrix ADC CLI: If the name includes one or more spaces, enclose the name in double or single quotation marks (for example, "my authentication policy" or 'my

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
			authentication policy').
rule	<String>	Read-write	Name of the Citrix ADC named rule, or an expression, that the policy uses to perform the authentication.
reqaction	<String>	Read-only	The name of the RADIUS action the policy uses.
__count	<Double>	Read-only	count parameter

Operations

(click to see Properties)

ADD| DELETE| UPDATE| GET (ALL)| GET| COUNT

Some options that you can use for each operations:

- **Getting warnings in response:**NITRO allows you to get warnings in an operation by specifying the "warning" query parameter as "yes". For example, to get warnings while connecting to the Citrix ADC appliance, the URL is as follows:

  http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/login?warning=yes

  If any, the warnings are displayed in the response payload with the HTTP code "209 X-NITRO-WARNING".

- **Authenticated access for individual NITRO operations:**NITRO allows you to logon to the Citrix ADC appliance to perform individual operations. You can use this option instead of creating a NITRO session (using the login object) and then using that session to perform all operations,

  To do this, you must specify the username and password in the request header of the NITRO request as follows:

  X-NITRO-USER:<username>

X-NITRO-PASS:<password>

Note: In such cases, make sure that the request header DOES not include the following:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue>

`*Note:`*

Mandatory parameters are marked in redand placeholder content is marked in <green>.

add

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/authenticationlocalpolicy

HTTP Method:POST

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Content-Type:application/json

Request Payload:

```
{"authenticationlocalpolicy":{

<b>"name":<String_value>,

</b><b>"rule":<String_value>

</b>}}
```

Response:

HTTP Status Code on Success: 201 Created HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error

delete

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/authenticationlocalpolicy/name_value<String>

HTTP Method:DELETE

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue>

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error

update

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/authenticationlocalpolicy

HTTP Method:PUT

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Content-Type:application/json

Request Payload:

```
{"authenticationlocalpolicy":{

<b>"name":<String_value>,

</b>"rule":<String_value>

}}
```

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error

get (all)

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/authenticationlocalpolicy

Query-parameters:

attrs

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/authenticationlocalpolicy?**attrs=property-name1,property-name2**

Use this query parameter to specify the resource details that you want to retrieve.

filter

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/authenticationlocalpolicy?**filter=property-name1:property-val1,property-name2:property-val2**

Use this query-parameter to get the filtered set of authenticationlocalpolicy resources configured on Citrix ADC. Filtering can be done on any of the properties of the resource.

view

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/authenticationlocalpolicy?**view=summary**

Note: By default, the retrieved results are displayed in detail view (?view=detail).

pagination

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/authenticationlocalpolicy?**pagesize=#no;pageno=#no**

Use this query-parameter to get the authenticationlocalpolicy resources in chunks.

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "authenticationlocalpolicy": [ {

"name":<String_value>,

"rule":<String_value>,

"reqaction":<String_value>

}]}
```

get

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/authenticationlocalpolicy/name_value<String>

Query-parameters:

attrs

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/authenticationlocalpolicy/name_value<String>?**attrs=property-name1,property-name2**

Use this query parameter to specify the resource details that you want to retrieve.

view

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/authenticationlocalpolicy/name_value<String>?**view=summary**

Note: By default, the retrieved results are displayed in detail view (?view=detail).

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "authenticationlocalpolicy": [ {

"name":<String_value>,

"rule":<String_value>,

"reqaction":<String_value>

}]}
```

count

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/authenticationlocalpolicy?**count=yes**

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "authenticationlocalpolicy": [ { "__count": "#no"} ] }
```

authenticationlocalpolicy_authenticationvserver_binding

Binding object showing the authenticationvserver that can be bound to authenticationlocalpolicy.

Properties

(click to see Operations)

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
boundto	<String>	Read-write	The entity name to which policy is bound.
name	<String>	Read-write	Name of the local authentication policy.
priority	<Double>	Read-only	.
activepolicy	<Double>	Read-only	.
__count	<Double>	Read-only	count parameter

Operations

(click to see Properties)

GET| GET (ALL)| COUNT

Some options that you can use for each operations:

- **Getting warnings in response:**NITRO allows you to get warnings in an operation by specifying the "warning" query parameter as "yes". For example, to get warnings while connecting to the Citrix ADC appliance, the URL is as follows:

  http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/login?warning=yes

  If any, the warnings are displayed in the response payload with the HTTP code "209 X-NITRO-WARNING".

- **Authenticated access for individual NITRO operations:**NITRO allows you to logon to the Citrix ADC appliance to perform individual operations. You can use this option instead of creating a NITRO session (using the login object) and then using that session to perform all operations,

  To do this, you must specify the username and password in the request header of the NITRO request as follows:

  X-NITRO-USER:<username>

  X-NITRO-PASS:<password>

  Note: In such cases, make sure that the request header DOES not include the following:

  Cookie:NITRO_AUTH_TOKEN=<tokenvalue>

`*Note:`*

Mandatory parameters are marked in redand placeholder content is marked in <green>.

get

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/authenticationlocalpolicy_authenticationvserver_binding/name_value<String>

Query-parameters:

filter

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/authenticationlocalpolicy_authenticationvserver_binding/name_value<String>?**filter=property-name1:property-value1,property-name2:property-value2**

Use this query-parameter to get the filtered set of authenticationlocalpolicy_authenticationvserver_binding resources configured on Citrix ADC. Filtering can be done on any of the properties of the resource.

pagination

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/authenticationlocalpolicy_authenticationvserver_binding/name_value<String>?**pagesize=#no;pageno=#no**

Use this query-parameter to get the authenticationlocalpolicy_authenticationvserver_binding resources in chunks.

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "authenticationlocalpolicy_authenticationvserver_binding": [ {

"boundto":<String_value>,

"name":<String_value>,

"priority":<Double_value>,

"activepolicy":<Double_value>

}]}
```

get (all)

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/authenticationlocalpolicy_authenticationvserver_binding

Query-parameters:

bulkbindings

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/authenticationlocalpolicy_authenticationvserver_binding?**bulkbindings=yes**

NITRO allows you to fetch bindings in bulk.

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "authenticationlocalpolicy_authenticationvserver_binding": [ {

"boundto":<String_value>,

"name":<String_value>,

"priority":<Double_value>,

"activepolicy":<Double_value>

}]}
```

count

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/authenticationlocalpolicy_authenticationvserver_binding/name_value<String>?**count=yes**

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{"authenticationlocalpolicy_authenticationvserver_binding": [ { "__count": "#no"} ] }
```

authenticationlocalpolicy_binding

Binding object which returns the resources bound to authenticationlocalpolicy.

Properties

(click to see Operations)

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
name	<String>	Read-write	Name of the local authentication policy.
authenticationlocalpolicy_vpnvserver_binding	<authenticationlocalpolicy_vpnvserver_binding[]>	Read-only	vpnvserver that can be bound to authenticationlocalpolicy.
authenticationlocalpolicy_systemglobal_binding	<authenticationlocalpolicy_systemglobal_binding[]>	Read-only	systemglobal that can be bound to authenticationlocalpolicy.
authenticationlocalpolicy_authenticationvserver_binding	<authenticationlocalpolicy_authenticationvserver_binding[]>	Read-only	authenticationvserver that can be bound to authenticationlocalpolicy.

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
authenticationlocalpolicy_vpnglobal_binding	<authenticationlocalpolicy_vpnglobal_binding[]>	Read-only	vpnglobal that can be bound to authenticationlocalpolicy.

Operations

(click to see Properties)

GET| GET (ALL)

Some options that you can use for each operations:

- **Getting warnings in response:**NITRO allows you to get warnings in an operation by specifying the "warning" query parameter as "yes". For example, to get warnings while connecting to the Citrix ADC appliance, the URL is as follows:

  http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/login?warning=yes

  If any, the warnings are displayed in the response payload with the HTTP code "209 X-NITRO-WARNING".

- **Authenticated access for individual NITRO operations:**NITRO allows you to logon to the Citrix ADC appliance to perform individual operations. You can use this option instead of creating a NITRO session (using the login object) and then using that session to perform all operations,

  To do this, you must specify the username and password in the request header of the NITRO request as follows:

  X-NITRO-USER:<username>

  X-NITRO-PASS:<password>

  Note: In such cases, make sure that the request header DOES not include the following:

  Cookie:NITRO_AUTH_TOKEN=<tokenvalue>

`*Note:`*

Mandatory parameters are marked in redand placeholder content is marked in <green>.

get

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/authenticationlocalpolicy_binding/name_value<String>

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "authenticationlocalpolicy_binding": [ {

"name":<String_value>,

"authenticationlocalpolicy_vpnvserver_binding":<authenticationlocalpolicy_vpnvserver_binding[]_value>,

"authenticationlocalpolicy_vpnglobal_binding":<authenticationlocalpolicy_vpnglobal_binding[]_value>,

"authenticationlocalpolicy_systemglobal_binding":<authenticationlocalpolicy_systemglobal_binding[]_value>,

"authenticationlocalpolicy_authenticationvserver_binding":<authenticationlocalpolicy_authenticationvserver_binding[]_value>

}]}
```

get (all)

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/authenticationlocalpolicy_binding

Query-parameters:

bulkbindings

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/authenticationlocalpolicy_binding?**bulkbindings=yes**

NITRO allows you to fetch bindings in bulk.

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "authenticationlocalpolicy_binding": [ {

"name":<String_value>,

"authenticationlocalpolicy_vpnvserver_binding":<authenticationlocalpolicy_
vpnvserver_binding[]_value>,

"authenticationlocalpolicy_vpnglobal_binding":<authenticationlocalpolicy_v
pnglobal_binding[]_value>,

"authenticationlocalpolicy_systemglobal_binding":<authenticationlocalpolic
y_systemglobal_binding[]_value>,

"authenticationlocalpolicy_authenticationvserver_binding":<authenticationl
ocalpolicy_authenticationvserver_binding[]_value>

}]}
```

authenticationlocalpolicy_systemglobal_binding

Binding object showing the systemglobal that can be bound to authenticationlocalpolicy.

Properties

(click to see Operations)

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
boundto	<String>	Read-write	The entity name to which policy is bound.
name	<String>	Read-write	Name of the local authentication policy.
priority	<Double>	Read-only	.
activepolicy	<Double>	Read-only	.
__count	<Double>	Read-only	count parameter

Operations

(click to see Properties)

GET| GET (ALL)| COUNT

Some options that you can use for each operations:

- **Getting warnings in response:**NITRO allows you to get warnings in an operation by specifying the "warning" query parameter as "yes". For example, to get warnings while connecting to the Citrix ADC appliance, the URL is as follows:

  http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/login?warning=yes

  If any, the warnings are displayed in the response payload with the HTTP code "209 X-NITRO-WARNING".

- **Authenticated access for individual NITRO operations:**NITRO allows you to logon to the Citrix ADC appliance to perform individual operations. You can use this option instead of

creating a NITRO session (using the login object) and then using that session to perform all operations,

To do this, you must specify the username and password in the request header of the NITRO request as follows:

X-NITRO-USER:<username>

X-NITRO-PASS:<password>

Note: In such cases, make sure that the request header DOES not include the following:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue>

`*Note:`*

Mandatory parameters are marked in redand placeholder content is marked in <green>.

get

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/authenticationlocalpolicy_systemglobal_binding/name_value<String>

Query-parameters:

filter

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/authenticationlocalpolicy_systemglobal_binding/name_value<String>?**filter=property-name1:property-value1,property-name2:property-value2**

Use this query-parameter to get the filtered set of authenticationlocalpolicy_systemglobal_binding resources configured on Citrix ADC. Filtering can be done on any of the properties of the resource.

pagination

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/authenticationlocalpolicy_systemglobal_binding/name_value<String>?**pagesize=#no;pageno=#no**

Use this query-parameter to get the authenticationlocalpolicy_systemglobal_binding resources in chunks.

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "authenticationlocalpolicy_systemglobal_binding": [ {

"boundto":<String_value>,

"name":<String_value>,

"priority":<Double_value>,

"activepolicy":<Double_value>

}]}
```

get (all)

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/authenticationlocalpolicy_systemglobal_binding

Query-parameters:

bulkbindings

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/authenticationlocalpolicy_systemglobal_binding?**bulkbindings=yes**

NITRO allows you to fetch bindings in bulk.

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "authenticationlocalpolicy_systemglobal_binding": [ {

"boundto":<String_value>,

"name":<String_value>,

"priority":<Double_value>,

"activepolicy":<Double_value>

}]}
```

count

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/authenticationlocalpolicy_systemglobal_binding/name_value<String>?**count=yes**

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{"authenticationlocalpolicy_systemglobal_binding": [ { "__count":
"#no"} ] }
```

authenticationlocalpolicy_vpnglobal_binding

Binding object showing the vpnglobal that can be bound to authenticationlocalpolicy.

Properties

(click to see Operations)

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
boundto	<String>	Read-write	The entity name to which policy is bound.
name	<String>	Read-write	Name of the local authentication policy.
priority	<Double>	Read-only	.
activepolicy	<Double>	Read-only	.
__count	<Double>	Read-only	count parameter

Operations

(click to see Properties)

GET| GET (ALL)| COUNT

Some options that you can use for each operations:

- **Getting warnings in response:**NITRO allows you to get warnings in an operation by specifying the "warning" query parameter as "yes". For example, to get warnings while connecting to the Citrix ADC appliance, the URL is as follows:

  http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/login?warning=yes

  If any, the warnings are displayed in the response payload with the HTTP code "209 X-NITRO-WARNING".

- **Authenticated access for individual NITRO operations:**NITRO allows you to logon to the Citrix ADC appliance to perform individual operations. You can use this option instead of creating a NITRO session (using the login object) and then using that session to perform all operations,

  To do this, you must specify the username and password in the request header of the NITRO request as follows:

  X-NITRO-USER:<username>

  X-NITRO-PASS:<password>

  Note: In such cases, make sure that the request header DOES not include the following:

  Cookie:NITRO_AUTH_TOKEN=<tokenvalue>

`*Note:`*

Mandatory parameters are marked in redand placeholder content is marked in <green>.

get

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/authenticationlocalpolicy_vpnglobal_binding/name_value<String>

Query-parameters:

filter

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/authenticationlocalpolicy_vpnglobal_binding/name_value<String>?**filter=property-name1:property-value1,property-name2:property-value2**

Use this query-parameter to get the filtered set of authenticationlocalpolicy_vpnglobal_binding resources configured on Citrix ADC. Filtering can be done on any of the properties of the resource.

pagination

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/authenticationlocalpolicy_vpnglobal_binding/name_value<String>?**pagesize=#no;pageno=#no**

Use this query-parameter to get the authenticationlocalpolicy_vpnglobal_binding resources in chunks.

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "authenticationlocalpolicy_vpnglobal_binding": [ {

"boundto":<String_value>,

"name":<String_value>,

"priority":<Double_value>,

"activepolicy":<Double_value>

}]}
```

get (all)

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/authenticationlocalpolicy_vpnglobal_binding

Query-parameters:

bulkbindings

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/authenticationlocalpolicy_vpnglobal_binding?**bulkbindings=yes**

NITRO allows you to fetch bindings in bulk.

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "authenticationlocalpolicy_vpnglobal_binding": [ {

"boundto":<String_value>,

"name":<String_value>,

"priority":<Double_value>,

"activepolicy":<Double_value>

}]}
```

count

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/authenticationlocalpolicy_vpnglobal_binding/name_value<String>?**count=yes**

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{"authenticationlocalpolicy_vpnglobal_binding": [ { "__count": "#no"} ] }
```

authenticationlocalpolicy_vpnvserver_binding

Binding object showing the vpnvserver that can be bound to authenticationlocalpolicy.

Properties

(click to see Operations)

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
boundto	<String>	Read-write	The entity name to which policy is bound.
name	<String>	Read-write	Name of the local authentication policy.
priority	<Double>	Read-only	.
activepolicy	<Double>	Read-only	.
__count	<Double>	Read-only	count parameter

Operations

(click to see Properties)

GET| GET (ALL)| COUNT

Some options that you can use for each operations:

- **Getting warnings in response:**NITRO allows you to get warnings in an operation by specifying the "warning" query parameter as "yes". For example, to get warnings while connecting to the Citrix ADC appliance, the URL is as follows:

  http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/login?warning=yes

  If any, the warnings are displayed in the response payload with the HTTP code "209 X-NITRO-WARNING".

- **Authenticated access for individual NITRO operations:**NITRO allows you to logon to the Citrix ADC appliance to perform individual operations. You can use this option instead of creating a NITRO session (using the login object) and then using that session to perform all operations,

  To do this, you must specify the username and password in the request header of the NITRO request as follows:

  X-NITRO-USER:<username>

  X-NITRO-PASS:<password>

  Note: In such cases, make sure that the request header DOES not include the following:

  Cookie:NITRO_AUTH_TOKEN=<tokenvalue>

`*Note:`*

Mandatory parameters are marked in redand placeholder content is marked in <green>.

get

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/authenticationlocalpolicy_vpnvserver_binding/name_value<String>

Query-parameters:

filter

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/authenticationlocalpolicy_vpnvserver_binding/name_value<String>?**filter=property-name1:property-value1,property-name2:property-value2**

Use this query-parameter to get the filtered set of authenticationlocalpolicy_vpnvserver_binding resources configured on Citrix ADC. Filtering can be done on any of the properties of the resource.

pagination

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/authenticationlocalpolicy_vpnvserver_binding/name_value<String>?**pagesize=#no;pageno=#no**

Use this query-parameter to get the authenticationlocalpolicy_vpnvserver_binding resources in chunks.

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "authenticationlocalpolicy_vpnvserver_binding": [ {

"boundto":<String_value>,

"name":<String_value>,

"priority":<Double_value>,

"activepolicy":<Double_value>

}]}
```

get (all)

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/authenticationlocalpolicy_vpnvserver_binding

Query-parameters:

bulkbindings

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/authenticationlocalpolicy_vpnvserver_binding?**bulkbindings=yes**

NITRO allows you to fetch bindings in bulk.

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "authenticationlocalpolicy_vpnvserver_binding": [ {

"boundto":<String_value>,

"name":<String_value>,

"priority":<Double_value>,

"activepolicy":<Double_value>

}]}
```

count

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/authenticationlocalpolicy_vpnvserver_binding/name_value<String>?**count=yes**

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{"authenticationlocalpolicy_vpnvserver_binding": [ { "__count": "#no"} ] }
```

authenticationloginschema

Configuration for 0 resource.

Properties

(click to see Operations)

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
name	<String>	Read-write	Name for the new login schema. Login schema defines the way login form is rendered. It provides a way to customize

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
			the fields that are shown to the user. Must begin with an ASCII alphanumeric or underscore (_) character, and must contain only ASCII alphanumeric, underscore, hash (#), period (.), space, colon (:), at (@), equals (=), and hyphen (-) characters. Cannot be changed after an action is created. The following requirement applies only to the Citrix ADC CLI: If the name includes one or more spaces, enclose the name in double or single quotation marks (for example, "my action" or 'my action'). Minimum length = 1
authenticationschema	<String>	Read-write	Name of the file for reading authentication schema to be sent for Login Page UI. This file should contain xml definition of elements as per

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
			Citrix Forms Authentication Protocol to be able to render login form. If administrator does not want to prompt users for additional credentials but continue with previously obtained credentials, then "noschema" can be given as argument. Please note that this applies only to loginSchemas that are used with user-defined factors, and not the vserver factor. Minimum length = 1
userexpression	<String>	Read-write	Expression for username extraction during login. This can be any relevant advanced policy expression. Minimum length = 1
passwdexpression	<String>	Read-write	Expression for password extraction during login. This can be any relevant advanced policy expression. Minimum length = 1
usercredentialindex	<Double>	Read-write	The index at which user entered

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
			username should be stored in session. Minimum value = 1 Maximum value = 16
passwordcredentialindex	<Double>	Read-write	The index at which user entered password should be stored in session. Minimum value = 1 Maximum value = 16
authenticationstrength	<Double>	Read-write	Weight of the current authentication. Minimum value = 0 Maximum value = 65535
ssocredentials	<String>	Read-write	This option indicates whether current factor credentials are the default SSO (SingleSignOn) credentials. Default value: NO Possible values = YES, NO
__count	<Double>	Read-only	count parameter

Operations

(click to see Properties)

ADD| DELETE| UPDATE| UNSET| GET (ALL)| GET| COUNT

Some options that you can use for each operations:

- **Getting warnings in response:**NITRO allows you to get warnings in an operation by specifying the "warning" query parameter as "yes". For example, to get warnings while connecting to the Citrix ADC appliance, the URL is as follows:

  http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/login?warning=yes

  If any, the warnings are displayed in the response payload with the HTTP code "209 X-NITRO-WARNING".

- **Authenticated access for individual NITRO operations:**NITRO allows you to logon to the Citrix ADC appliance to perform individual operations. You can use this option instead of creating a NITRO session (using the login object) and then using that session to perform all operations,

  To do this, you must specify the username and password in the request header of the NITRO request as follows:

  X-NITRO-USER:<username>

  X-NITRO-PASS:<password>

  Note: In such cases, make sure that the request header DOES not include the following:

  Cookie:NITRO_AUTH_TOKEN=<tokenvalue>

`*Note:`*

Mandatory parameters are marked in redand placeholder content is marked in <green>.

add

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/authenticationloginschema

HTTP Method:POST

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Content-Type:application/json

Request Payload:

```
{"authenticationloginschema":{

<b>"name":<String_value>,

</b><b>"authenticationschema":<String_value>,

</b>"userexpression":<String_value>,

"passwdexpression":<String_value>,

"usercredentialindex":<Double_value>,

"passwordcredentialindex":<Double_value>,

"authenticationstrength":<Double_value>,

"ssocredentials":<String_value>

}}
```

Response:

HTTP Status Code on Success: 201 Created HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error

delete

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/authenticationloginschema/name_value<String>

HTTP Method:DELETE

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue>

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error

update

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/authenticationloginschema

HTTP Method:PUT

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Content-Type:application/json

Request Payload:

```
{"authenticationloginschema":{

<b>"name":<String_value>,

</b>"authenticationschema":<String_value>,

"userexpression":<String_value>,

"passwdexpression":<String_value>,

"usercredentialindex":<Double_value>,

"passwordcredentialindex":<Double_value>,

"authenticationstrength":<Double_value>,

"ssocredentials":<String_value>

}}
```

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error

unset

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/authenticationloginschema?**action=unset**

HTTP Method:POST

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Content-Type:application/json

Request Payload:

```
{"authenticationloginschema":{

<b>"name":<String_value>,

</b>"userexpression":true,

"passwdexpression":true,

"usercredentialindex":true,

"passwordcredentialindex":true,

"authenticationstrength":true,

"ssocredentials":true

}}
```

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error

get (all)

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/authenticationloginschema

Query-parameters:

attrs

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/authenticationloginschema?**attrs=property-name1,property-name2**

Use this query parameter to specify the resource details that you want to retrieve.

filter

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/authenticationloginschema?**filter=property-name1:property-val1,property-name2:property-val2**

Use this query-parameter to get the filtered set of authenticationloginschema resources configured on Citrix ADC. Filtering can be done on any of the properties of the resource.

view

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/authenticationloginschema?**view=summary**

Note: By default, the retrieved results are displayed in detail view (?view=detail).

pagination

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/authenticationloginschema?**pagesize=#no;pageno=#no**

Use this query-parameter to get the authenticationloginschema resources in chunks.

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "authenticationloginschema": [ {

"name":<String_value>,

"authenticationschema":<String_value>,

"userexpression":<String_value>,

"passwdexpression":<String_value>,

"usercredentialindex":<Double_value>,

"passwordcredentialindex":<Double_value>,

"authenticationstrength":<Double_value>,

"ssocredentials":<String_value>

}]}
```

get

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/authenticationloginschema/name_value<String>

Query-parameters:

attrs

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/authenticationloginschema/name_value<String>?**attrs=property-name1,property-name2**

Use this query parameter to specify the resource details that you want to retrieve.

view

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/authenticationloginschema/name_value<String>?**view=summary**

Note: By default, the retrieved results are displayed in detail view (?view=detail).

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "authenticationloginschema": [ {

"name":<String_value>,

"authenticationschema":<String_value>,

"userexpression":<String_value>,

"passwdexpression":<String_value>,

"usercredentialindex":<Double_value>,

"passwordcredentialindex":<Double_value>,

"authenticationstrength":<Double_value>,

"ssocredentials":<String_value>

}]}
```

count

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/authenticationloginschema?**count=yes**

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "authenticationloginschema": [ { "__count": "#no"} ] }
```

authenticationloginschemapolicy

Configuration for 0 resource.

Properties

(click to see Operations)

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
name	<String>	Read-write	Name for the LoginSchema policy. This is used for selecting parameters for user logon. Must begin with an ASCII alphanumeric or underscore (_) character, and must contain only ASCII alphanumeric, underscore, hash (#), period (.), space, colon (:), at (@),

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
			equals (=), and hyphen (-) characters. Cannot be changed after the policy is created. The following requirement applies only to the Citrix ADC CLI: If the name includes one or more spaces, enclose the name in double or single quotation marks (for example, "my policy" or 'my policy'). Minimum length = 1
rule	<String>	Read-write	Expression which is evaluated to choose a profile for authentication. The following requirements apply only to the Citrix ADC CLI: * If the expression includes one or more spaces, enclose the entire expression in double quotation marks.

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
			* If the expression itself includes double quotation marks, escape the quotations by using the \ character. * Alternatively, you can use single quotation marks to enclose the rule, in which case you do not have to escape the double quotation marks. Minimum length = 1
action	<String>	Read-write	Name of the profile to apply to requests or connections that match this policy. * NOOP - Do not take any specific action when this policy evaluates to true. This is useful to implicitly go to a different policy set. * RESET - Reset the client connection by closing it. The client program, such as a browser, will handle this and may inform the user. The client

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
			may then resend the request if desired. * DROP - Drop the request without sending a response to the user. Minimum length = 1
undefaction	<String>	Read-write	Action to perform if the result of policy evaluation is undefined (UNDEF). An UNDEF event indicates an internal error condition. Only the above built-in actions can be used.
comment	<String>	Read-write	Any comments to preserve information about this policy.
logaction	<String>	Read-write	Name of messagelog action to use when a request matches this policy.
newname	<String>	Read-write	New name for the LoginSchema policy. Must begin with a letter, number, or the underscore character (_), and must contain only letters, numbers, and the hyphen (-), period (.) hash (#),

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
			space ( ), at (@), equals (=), colon (:), and underscore characters. The following requirement applies only to the Citrix ADC CLI: If the name includes one or more spaces, enclose the name in double or single quotation marks (for example, "my loginschemapolicy policy" or 'my loginschemapolicy policy'). Minimum length = 1
hits	<Double>	Read-only	Number of hits.
undefhits	<Double>	Read-only	Number of Undef hits.
__count	<Double>	Read-only	count parameter

Operations

(click to see Properties)

ADD| DELETE| UPDATE| UNSET| RENAME| GET (ALL)| GET| COUNT

Some options that you can use for each operations:

- **Getting warnings in response:**NITRO allows you to get warnings in an operation by specifying the "warning" query parameter as "yes". For example, to get warnings while connecting to the Citrix ADC appliance, the URL is as follows:

  http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/login?warning=yes

  If any, the warnings are displayed in the response payload with the HTTP code "209 X-NITRO-WARNING".

- **Authenticated access for individual NITRO operations:**NITRO allows you to logon to the Citrix ADC appliance to perform individual operations. You can use this option instead of creating a NITRO session (using the login object) and then using that session to perform all operations,

  To do this, you must specify the username and password in the request header of the NITRO request as follows:

  X-NITRO-USER:<username>

  X-NITRO-PASS:<password>

  Note: In such cases, make sure that the request header DOES not include the following:

  Cookie:NITRO_AUTH_TOKEN=<tokenvalue>

`*Note:`*

Mandatory parameters are marked in redand placeholder content is marked in <green>.

add

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/authenticationloginschemapolicy

HTTP Method:POST

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Content-Type:application/json

Request Payload:

```
{"authenticationloginschemapolicy":{
```

```
<b>"name":<String_value>,

</b><b>"rule":<String_value>,

</b><b>"action":<String_value>,

</b>"undefaction":<String_value>,

"comment":<String_value>,

"logaction":<String_value>

}}
```

Response:

HTTP Status Code on Success: 201 Created HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error

delete

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/authenticationloginschemapolicy/name_value<String>

HTTP Method:DELETE

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue>

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error

update

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/authenticationloginschemapolicy

HTTP Method:PUT

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Content-Type:application/json

Request Payload:

```
{"authenticationloginschemapolicy":{

<b>"name":<String_value>,

</b>"rule":<String_value>,

"action":<String_value>,

"undefaction":<String_value>,

"comment":<String_value>,

"logaction":<String_value>

}}
```

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error

unset

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/authenticationloginschemapolicy?**action=unset**

HTTP Method:POST

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Content-Type:application/json

Request Payload:

```
{"authenticationloginschemapolicy":{

<b>"name":<String_value>,

</b>"undefaction":true,

"comment":true,

"logaction":true

}}
```

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error

rename

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/authenticationloginschemapolicy?**action=rename**

HTTP Method:POST

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Content-Type:application/json

Request Payload:

```
{"authenticationloginschemapolicy":{

<b>"name":<String_value>,

</b><b>"newname":<String_value>

</b>}}
```

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error

get (all)

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/authenticationloginschemapolicy

Query-parameters:

attrs

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/authenticationloginschemapolicy?**attrs=property-name1,property-name2**

Use this query parameter to specify the resource details that you want to retrieve.

filter

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/authenticationloginschemapolicy?**filter=property-name1:property-val1,property-name2:property-val2**

Use this query-parameter to get the filtered set of authenticationloginschemapolicy resources configured on Citrix ADC. Filtering can be done on any of the properties of the resource.

view

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/authenticationloginschemapolicy?**view=summary**

Note: By default, the retrieved results are displayed in detail view (?view=detail).

pagination

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/authenticationloginschemapolicy?**pagesize=#no;pageno=#no**

Use this query-parameter to get the authenticationloginschemapolicy resources in chunks.

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "authenticationloginschemapolicy": [ {

"name":<String_value>,

"rule":<String_value>,

"action":<String_value>,

"undefaction":<String_value>,

"comment":<String_value>,

"logaction":<String_value>,

"hits":<Double_value>,

"undefhits":<Double_value>

}]}
```

get

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/authenticationloginschemapolicy/name_value<String>

Query-parameters:

attrs

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/authenticationloginschemapolicy/name_value<String>?**attrs=property-name1,property-name2**

Use this query parameter to specify the resource details that you want to retrieve.

view

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/authenticationloginschemapolicy/name_value<String>?**view=summary**

Note: By default, the retrieved results are displayed in detail view (?view=detail).

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "authenticationloginschemapolicy": [ {

"name":<String_value>,

"rule":<String_value>,

"action":<String_value>,

"undefaction":<String_value>,

"comment":<String_value>,
```

```
"logaction":<String_value>,

"hits":<Double_value>,

"undefhits":<Double_value>

}]}
```

count

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/authenticationloginschemapolicy?**count=yes**

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "authenticationloginschemapolicy": [ { "__count": "#no"} ] }
```

authenticationloginschemapolicy_authenticationvserver_binding

Binding object showing the authenticationvserver that can be bound to authenticationloginschemapolicy.

Properties

(click to see Operations)

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
boundto	<String>	Read-write	The entity name to which policy is bound.
name	<String>	Read-write	Name of the LoginSchema policy for which to display detailed information. Minimum length = 1
priority	<Double>	Read-only	.
gotopriorityexpression	<String>	Read-only	Expression specifying the priority of the next policy which will get evaluated if the current policy rule evaluates to TRUE.
labeltype	<String>	Read-only	Type of policy label invocation. Possible values = reqvserver, resvserver, policylabel
activepolicy	<Double>	Read-only	.
labelname	<String>	Read-only	Name of the label to invoke if the current policy rule evaluates to TRUE.
__count	<Double>	Read-only	count parameter

Operations

(click to see Properties)

GET| GET (ALL)| COUNT

Some options that you can use for each operations:

- **Getting warnings in response:**NITRO allows you to get warnings in an operation by specifying the "warning" query parameter as "yes". For example, to get warnings while connecting to the Citrix ADC appliance, the URL is as follows:

  http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/login?warning=yes

  If any, the warnings are displayed in the response payload with the HTTP code "209 X-NITRO-WARNING".

- **Authenticated access for individual NITRO operations:**NITRO allows you to logon to the Citrix ADC appliance to perform individual operations. You can use this option instead of creating a NITRO session (using the login object) and then using that session to perform all operations,

  To do this, you must specify the username and password in the request header of the NITRO request as follows:

  X-NITRO-USER:<username>

  X-NITRO-PASS:<password>

  Note: In such cases, make sure that the request header DOES not include the following:

  Cookie:NITRO_AUTH_TOKEN=<tokenvalue>

`*Note:`*

Mandatory parameters are marked in redand placeholder content is marked in <green>.

get

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/authenticationloginschemapolicy_authenticationvserver_binding/name_value<String>

Query-parameters:

filter

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/authenticationloginschemapolicy_authenticationvserver_binding/name_value<String>?**filter=property-name1:property-value1,property-name2:property-value2**

Use this query-parameter to get the filtered set of authenticationloginschemapolicy_authenticationvserver_binding resources configured on Citrix ADC. Filtering can be done on any of the properties of the resource.

pagination

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/authenticationloginschemapolicy_authenticationvserver_binding/name_value<String>?**pagesize=#no;pageno=#no**

Use this query-parameter to get the authenticationloginschemapolicy_authenticationvserver_binding resources in chunks.

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "authenticationloginschemapolicy_authenticationvserver_binding": [ {

"boundto":<String_value>,

"name":<String_value>,

"priority":<Double_value>,

"gotopriorityexpression":<String_value>,

"labeltype":<String_value>,

"activepolicy":<Double_value>,
```

```
"labelname":<String_value>

}]}
```

get (all)

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/authenticationloginschemapolicy_authenticationvserver_binding

Query-parameters:

bulkbindings

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/authenticationloginschemapolicy_authenticationvserver_binding?**bulkbindings=yes**

NITRO allows you to fetch bindings in bulk.

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "authenticationloginschemapolicy_authenticationvserver_binding": [ {

"boundto":<String_value>,

"name":<String_value>,

"priority":<Double_value>,
```

```
"gotopriorityexpression":<String_value>,

"labeltype":<String_value>,

"activepolicy":<Double_value>,

"labelname":<String_value>

}]}
```

count

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/authenticationloginschemapolicy_authenticationvserver_binding/name_value<String>?**count=yes**

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{"authenticationloginschemapolicy_authenticationvserver_binding":
[ { "__count": "#no"} ] }
```

authenticationloginschemapolicy_binding

Binding object which returns the resources bound to authenticationloginschemapolicy.

Properties

(click to see Operations)

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
name	<String>	Read-write	Name of the LoginSchema policy for which to display detailed information. Minimum length = 1
authenticationloginschemapolicy_vpnvserver_binding	<authenticationloginschemapolicy_vpnvserver_binding[]>	Read-only	vpnvserver that can be bound to authenticationloginschemapolicy.
authenticationloginschemapolicy_authenticationvserver_binding	<authenticationloginschemapolicy_authenticationvserver_binding[]>	Read-only	authenticationvserver that can be bound to authenticationloginschemapolicy.

Operations

(click to see Properties)

GET| GET (ALL)

Some options that you can use for each operations:

- **Getting warnings in response:**NITRO allows you to get warnings in an operation by specifying the "warning" query parameter as "yes". For example, to get warnings while connecting to the Citrix ADC appliance, the URL is as follows:

  http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/login?warning=yes

  If any, the warnings are displayed in the response payload with the HTTP code "209 X-NITRO-WARNING".

- **Authenticated access for individual NITRO operations:**NITRO allows you to logon to the Citrix ADC appliance to perform individual operations. You can use this option instead of

creating a NITRO session (using the login object) and then using that session to perform all operations,

To do this, you must specify the username and password in the request header of the NITRO request as follows:

X-NITRO-USER:<username>

X-NITRO-PASS:<password>

Note: In such cases, make sure that the request header DOES not include the following:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue>

`*Note:`*

Mandatory parameters are marked in redand placeholder content is marked in <green>.

get

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/authenticationloginschemapolicy_binding/name_value<String>

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "authenticationloginschemapolicy_binding": [ {

"name":<String_value>,
```

```
"authenticationloginschemapolicy_authenticationvserver_binding":<authentic
ationloginschemapolicy_authenticationvserver_binding[]_value>,

"authenticationloginschemapolicy_vpnvserver_binding":<authenticationlogins
chemapolicy_vpnvserver_binding[]_value>

}]}
```

get (all)

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/authenticationloginschemapolicy_binding

Query-parameters:

bulkbindings

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/authenticationloginschemapolicy_binding?**bulkbindings=yes**

NITRO allows you to fetch bindings in bulk.

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "authenticationloginschemapolicy_binding": [ {

"name":<String_value>,
```

```
"authenticationloginschemapolicy_authenticationvserver_binding":<authentic
ationloginschemapolicy_authenticationvserver_binding[]_value>,

"authenticationloginschemapolicy_vpnvserver_binding":<authenticationlogins
chemapolicy_vpnvserver_binding[]_value>

}]}
```

authenticationloginschemapolicy_vpnvserver_binding

Binding object showing the vpnvserver that can be bound to authenticationloginschemapolicy.

Properties

(click to see Operations)

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
boundto	<String>	Read-write	The entity name to which policy is bound.
name	<String>	Read-write	Name of the LoginSchema policy for which to display detailed information. Minimum length = 1
priority	<Double>	Read-only	.
gotopriorityexpression	<String>	Read-only	Expression specifying the priority of the next policy which will get evaluated if the current policy rule evaluates to TRUE.
labeltype	<String>	Read-only	Type of policy label invocation.

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
			Possible values = reqvserver, resvserver, policylabel
activepolicy	<Double>	Read-only	.
labelname	<String>	Read-only	Name of the label to invoke if the current policy rule evaluates to TRUE.
__count	<Double>	Read-only	count parameter

Operations

(click to see Properties)

GET| GET (ALL)| COUNT

Some options that you can use for each operations:

- **Getting warnings in response:**NITRO allows you to get warnings in an operation by specifying the "warning" query parameter as "yes". For example, to get warnings while connecting to the Citrix ADC appliance, the URL is as follows:

  http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/login?warning=yes

  If any, the warnings are displayed in the response payload with the HTTP code "209 X-NITRO-WARNING".

- **Authenticated access for individual NITRO operations:**NITRO allows you to logon to the Citrix ADC appliance to perform individual operations. You can use this option instead of creating a NITRO session (using the login object) and then using that session to perform all operations,

  To do this, you must specify the username and password in the request header of the NITRO request as follows:

  X-NITRO-USER:<username>

  X-NITRO-PASS:<password>

Note: In such cases, make sure that the request header DOES not include the following:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue>

`*Note:`*

Mandatory parameters are marked in redand placeholder content is marked in <green>.

get

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/authenticationloginschemapolicy_vpnvserver_binding/name_value<String>

Query-parameters:

filter

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/authenticationloginschemapolicy_vpnvserver_binding/name_value<String>?**filter=property-name1:property-value1,property-name2:property-value2**

Use this query-parameter to get the filtered set of authenticationloginschemapolicy_vpnvserver_binding resources configured on Citrix ADC. Filtering can be done on any of the properties of the resource.

pagination

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/authenticationloginschemapolicy_vpnvserver_binding/name_value<String>?**pagesize=#no;pageno=#no**

Use this query-parameter to get the authenticationloginschemapolicy_vpnvserver_binding resources in chunks.

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "authenticationloginschemapolicy_vpnvserver_binding": [ {

"boundto":<String_value>,

"name":<String_value>,

"priority":<Double_value>,

"gotopriorityexpression":<String_value>,

"labeltype":<String_value>,

"activepolicy":<Double_value>,

"labelname":<String_value>

}]}
```

get (all)

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/authenticationloginschemapolicy_vpnvserver_binding

Query-parameters:

bulkbindings

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/authenticationloginschemapolicy_vpnvserver_binding?**bulkbindings=yes**

NITRO allows you to fetch bindings in bulk.

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "authenticationloginschemapolicy_vpnvserver_binding": [ {

"boundto":<String_value>,

"name":<String_value>,

"priority":<Double_value>,

"gotopriorityexpression":<String_value>,

"labeltype":<String_value>,

"activepolicy":<Double_value>,

"labelname":<String_value>

}]}
```

count

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/authenticationloginschemapolicy_vpnvserver_binding/name_value<String>?**count=yes**

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{"authenticationloginschemapolicy_vpnvserver_binding": [ { "__count":
"#no"} ] }
```

authenticationnegotiateaction

Configuration for Negotiate action resource.

Properties

(click to see Operations)

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
name	<String>	Read-write	Name for the AD KDC server profile (negotiate action). Must begin with a letter, number, or the underscore character (_), and must contain only letters, numbers, and the hyphen (-), period (.) pound (#), space ( ), at (@), equals (=), colon (:), and underscore characters. Cannot be changed after

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
			AD KDC server profile is created. The following requirement applies only to the Citrix ADC CLI: If the name includes one or more spaces, enclose the name in double or single quotation marks (for example, "my authentication action" or 'my authentication action'). Minimum length = 1
domain	<String>	Read-write	Domain name of the service principal that represnts Citrix ADC. Minimum length = 1
domainuser	<String>	Read-write	User name of the account that is mapped with Citrix ADC principal. This can be given along with domain and password when keytab file is not available. If username is given along with keytab file, then that keytab file will be searched

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
			for this user's credentials. Minimum length = 1
domainuserpasswd	<String>	Read-write	Password of the account that is mapped to the Citrix ADC principal. Minimum length = 1
ou	<String>	Read-write	Active Directory organizational units (OU) attribute. Minimum length = 1
defaultauthentication group	<String>	Read-write	This is the default group that is chosen when the authentication succeeds in addition to extracted groups.
keytab	<String>	Read-write	The path to the keytab file that is used to decrypt kerberos tickets presented to Citrix ADC. If keytab is not available, domain/ username/password can be specified in the negotiate action configuration. Minimum length = 1
ntlmpath	<String>	Read-write	The path to the site that is enabled for NTLM authentication, including FQDN of the server. This is

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
			used when clients fallback to NTLM. Minimum length = 1
kcdspn	<String>	Read-only	Host SPN extracted from keytab file.
__count	<Double>	Read-only	count parameter

Operations

(click to see Properties)

ADD| DELETE| UPDATE| UNSET| GET (ALL)| GET| COUNT

Some options that you can use for each operations:

- **Getting warnings in response:**NITRO allows you to get warnings in an operation by specifying the "warning" query parameter as "yes". For example, to get warnings while connecting to the Citrix ADC appliance, the URL is as follows:

  http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/login?warning=yes

  If any, the warnings are displayed in the response payload with the HTTP code "209 X-NITRO-WARNING".

- **Authenticated access for individual NITRO operations:**NITRO allows you to logon to the Citrix ADC appliance to perform individual operations. You can use this option instead of creating a NITRO session (using the login object) and then using that session to perform all operations,

  To do this, you must specify the username and password in the request header of the NITRO request as follows:

  X-NITRO-USER:<username>

  X-NITRO-PASS:<password>

  Note: In such cases, make sure that the request header DOES not include the following:

  Cookie:NITRO_AUTH_TOKEN=<tokenvalue>

`*Note:`*

Mandatory parameters are marked in redand placeholder content is marked in <green>.

add

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/authenticationnegotiateaction

HTTP Method:POST

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Content-Type:application/json

Request Payload:

```
{"authenticationnegotiateaction":{

<b>"name":<String_value>,

</b>"domain":<String_value>,

"domainuser":<String_value>,

"domainuserpasswd":<String_value>,

"ou":<String_value>,

"defaultauthenticationgroup":<String_value>,

"keytab":<String_value>,

"ntlmpath":<String_value>

}}
```

Response:

HTTP Status Code on Success: 201 Created HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error

delete

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/authenticationnegotiateaction/name_value<String>

HTTP Method:DELETE

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue>

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error

update

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/authenticationnegotiateaction

HTTP Method:PUT

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Content-Type:application/json

Request Payload:

```
{"authenticationnegotiateaction":{

<b>"name":<String_value>,

</b>"domain":<String_value>,

"domainuser":<String_value>,

"domainuserpasswd":<String_value>,

"ou":<String_value>,

"defaultauthenticationgroup":<String_value>,

"keytab":<String_value>,
```

```
"ntlmpath":<String_value>

}}
```

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error

unset

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/authenticationnegotiateaction?**action=unset**

HTTP Method:POST

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Content-Type:application/json

Request Payload:

```
{"authenticationnegotiateaction":{

<b>"name":<String_value>,

</b>"domain":true,

"domainuser":true,

"domainuserpasswd":true,

"ou":true,

"defaultauthenticationgroup":true,

"ntlmpath":true

}}
```

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error

get (all)

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/authenticationnegotiateaction

Query-parameters:

attrs

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/authenticationnegotiateaction?**attrs=property-name1,property-name2**

Use this query parameter to specify the resource details that you want to retrieve.

filter

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/authenticationnegotiateaction?**filter=property-name1:property-val1,property-name2:property-val2**

Use this query-parameter to get the filtered set of authenticationnegotiateaction resources configured on Citrix ADC. Filtering can be done on any of the properties of the resource.

view

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/authenticationnegotiateaction?**view=summary**

Note: By default, the retrieved results are displayed in detail view (?view=detail).

pagination

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/authenticationnegotiateaction?**pagesize=#no;pageno=#no**

Use this query-parameter to get the authenticationnegotiateaction resources in chunks.

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "authenticationnegotiateaction": [ {

"name":<String_value>,

"domain":<String_value>,

"domainuser":<String_value>,

"domainuserpasswd":<String_value>,

"ou":<String_value>,

"defaultauthenticationgroup":<String_value>,

"keytab":<String_value>,

"kcdspn":<String_value>,

"ntlmpath":<String_value>

}]}
```

get

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/authenticationnegotiateaction/name_value<String>

Query-parameters:

attrs

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/authenticationnegotiateaction/name_value<String>?**attrs=property-name1,property-name2**

Use this query parameter to specify the resource details that you want to retrieve.

view

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/authenticationnegotiateaction/name_value<String>?**view=summary**

Note: By default, the retrieved results are displayed in detail view (?view=detail).

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "authenticationnegotiateaction": [ {

"name":<String_value>,

"domain":<String_value>,

"domainuser":<String_value>,

"domainuserpasswd":<String_value>,

"ou":<String_value>,
```

```
"defaultauthenticationgroup":<String_value>,

"keytab":<String_value>,

"kcdspn":<String_value>,

"ntlmpath":<String_value>

}]}
```

count

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/authenticationnegotiateaction?**count=yes**

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "authenticationnegotiateaction": [ { "__count": "#no"} ] }
```

authenticationnegotiatepolicy

Configuration for Negotiate Policy resource.

Properties

(click to see Operations)

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
name	<String>	Read-write	Name for the negotiate authentication policy. Must begin with a letter, number, or the underscore character (_), and must contain only letters, numbers, and the hyphen (-), period (.) pound (#), space ( ), at (@), equals (=), colon (:), and underscore characters. Cannot be changed after AD KCD (negotiate) policy is created. The following requirement applies only to the Citrix ADC CLI: If the name includes one or more spaces, enclose the name in double or single quotation marks (for example, "my authentication policy" or 'my

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
			authentication policy'). Minimum length = 1
rule	<String>	Read-write	Name of the Citrix ADC named rule, or an expression, that the policy uses to determine whether to attempt to authenticate the user with the AD KCD server. Minimum length = 1
reqaction	<String>	Read-write	Name of the negotiate action to perform if the policy matches. Minimum length = 1
__count	<Double>	Read-only	count parameter

Operations

(click to see Properties)

ADD| DELETE| UPDATE| GET (ALL)| GET| COUNT

Some options that you can use for each operations:

- **Getting warnings in response:**NITRO allows you to get warnings in an operation by specifying the "warning" query parameter as "yes". For example, to get warnings while connecting to the Citrix ADC appliance, the URL is as follows:

  http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/login?warning=yes

  If any, the warnings are displayed in the response payload with the HTTP code "209 X-NITRO-WARNING".

- **Authenticated access for individual NITRO operations:**NITRO allows you to logon to the Citrix ADC appliance to perform individual operations. You can use this option instead of creating a NITRO session (using the login object) and then using that session to perform all operations,

  To do this, you must specify the username and password in the request header of the NITRO request as follows:

  X-NITRO-USER:<username>

  X-NITRO-PASS:<password>

  Note: In such cases, make sure that the request header DOES not include the following:

  Cookie:NITRO_AUTH_TOKEN=<tokenvalue>

`*Note:`*

Mandatory parameters are marked in redand placeholder content is marked in <green>.

add

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/authenticationnegotiatepolicy

HTTP Method:POST

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Content-Type:application/json

Request Payload:

```
{"authenticationnegotiatepolicy":{

<b>"name":<String_value>,

</b><b>"rule":<String_value>,

</b><b>"reqaction":<String_value>

</b>}}
```

Response:

HTTP Status Code on Success: 201 Created HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error

delete

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/authenticationnegotiatepolicy/name_value<String>

HTTP Method:DELETE

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue>

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error

update

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/authenticationnegotiatepolicy

HTTP Method:PUT

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Content-Type:application/json

Request Payload:

```
{"authenticationnegotiatepolicy":{

<b>"name":<String_value>,

</b>"rule":<String_value>,

"reqaction":<String_value>

}}
```

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error

get (all)

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/authenticationnegotiatepolicy

Query-parameters:

attrs

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/authenticationnegotiatepolicy?**attrs=property-name1,property-name2**

Use this query parameter to specify the resource details that you want to retrieve.

filter

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/authenticationnegotiatepolicy?**filter=property-name1:property-val1,property-name2:property-val2**

Use this query-parameter to get the filtered set of authenticationnegotiatepolicy resources configured on Citrix ADC. Filtering can be done on any of the properties of the resource.

view

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/authenticationnegotiatepolicy?**view=summary**

Note: By default, the retrieved results are displayed in detail view (?view=detail).

pagination

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/authenticationnegotiatepolicy?**pagesize=#no;pageno=#no**

Use this query-parameter to get the authenticationnegotiatepolicy resources in chunks.

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "authenticationnegotiatepolicy": [ {

"name":<String_value>,

"rule":<String_value>,

"reqaction":<String_value>

}]}
```

get

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/authenticationnegotiatepolicy/name_value<String>

Query-parameters:

attrs

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/authenticationnegotiatepolicy/name_value<String>?**attrs=property-name1,property-name2**

Use this query parameter to specify the resource details that you want to retrieve.

view

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/authenticationnegotiatepolicy/name_value<String>?**view=summary**

Note: By default, the retrieved results are displayed in detail view (?view=detail).

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "authenticationnegotiatepolicy": [ {

"name":<String_value>,

"rule":<String_value>,

"reqaction":<String_value>

}]}
```

count

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/authenticationnegotiatepolicy?**count=yes**

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "authenticationnegotiatepolicy": [ { "__count": "#no"} ] }
```

authenticationnegotiatepolicy_authenticationvserver_binding

Binding object showing the authenticationvserver that can be bound to authenticationnegotiatepolicy.

Properties

(click to see Operations)

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
boundto	<String>	Read-write	The entity name to which policy is bound.
name	<String>	Read-write	Name of the negotiate policy. Minimum length = 1
priority	<Double>	Read-only	.
activepolicy	<Double>	Read-only	.
__count	<Double>	Read-only	count parameter

Operations

(click to see Properties)

GET| GET (ALL)| COUNT

Some options that you can use for each operations:

- **Getting warnings in response:**NITRO allows you to get warnings in an operation by specifying the "warning" query parameter as "yes". For example, to get warnings while connecting to the Citrix ADC appliance, the URL is as follows:

  http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/login?warning=yes

  If any, the warnings are displayed in the response payload with the HTTP code "209 X-NITRO-WARNING".

- **Authenticated access for individual NITRO operations:**NITRO allows you to logon to the Citrix ADC appliance to perform individual operations. You can use this option instead of creating a NITRO session (using the login object) and then using that session to perform all operations,

  To do this, you must specify the username and password in the request header of the NITRO request as follows:

  X-NITRO-USER:<username>

  X-NITRO-PASS:<password>

  Note: In such cases, make sure that the request header DOES not include the following:

  Cookie:NITRO_AUTH_TOKEN=<tokenvalue>

`*Note:`*

Mandatory parameters are marked in redand placeholder content is marked in <green>.

get

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/authenticationnegotiatepolicy_authenticationvserver_binding/name_value<String>

Query-parameters:

filter

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/authenticationnegotiatepolicy_authenticationvserver_binding/name_value<String>?**filter=property-name1:property-value1,property-name2:property-value2**

Use this query-parameter to get the filtered set of authenticationnegotiatepolicy_authenticationvserver_binding resources configured on Citrix ADC. Filtering can be done on any of the properties of the resource.

pagination

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/authenticationnegotiatepolicy_authenticationvserver_binding/name_value<String>?**pagesize=#no;pageno=#no**

Use this query-parameter to get the authenticationnegotiatepolicy_authenticationvserver_binding resources in chunks.

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "authenticationnegotiatepolicy_authenticationvserver_binding": [ {

"boundto":<String_value>,

"name":<String_value>,

"priority":<Double_value>,

"activepolicy":<Double_value>

}]}
```

get (all)

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/authenticationnegotiatepolicy_authenticationvserver_binding

Query-parameters:

bulkbindings

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/authenticationnegotiatepolicy_authenticationvserver_binding?**bulkbindings=yes**

NITRO allows you to fetch bindings in bulk.

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "authenticationnegotiatepolicy_authenticationvserver_binding": [ {

"boundto":<String_value>,

"name":<String_value>,

"priority":<Double_value>,

"activepolicy":<Double_value>

}]}
```

count

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/authenticationnegotiatepolicy_authenticationvserver_binding/name_value<String>?**count=yes**

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{"authenticationnegotiatepolicy_authenticationvserver_binding":
[ { "__count": "#no"} ] }
```

authenticationnegotiatepolicy_binding

Binding object which returns the resources bound to authenticationnegotiatepolicy.

Properties

(click to see Operations)

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
name	<String>	Read-write	Name of the negotiate policy. Minimum length = 1

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
authenticationnegotiatepolicy_vpnvserver_binding	<authenticationnegotiatepolicy_vpnvserver_binding[]>	Read-only	vpnvserver that can be bound to authenticationnegotiatepolicy.
authenticationnegotiatepolicy_authenticationvserver_binding	<authenticationnegotiatepolicy_authenticationvserver_binding[]>	Read-only	authenticationvserver that can be bound to authenticationnegotiatepolicy.
authenticationnegotiatepolicy_vpnglobal_binding	<authenticationnegotiatepolicy_vpnglobal_binding[]>	Read-only	vpnglobal that can be bound to authenticationnegotiatepolicy.

Operations

(click to see Properties)

GET| GET (ALL)

Some options that you can use for each operations:

- **Getting warnings in response:**NITRO allows you to get warnings in an operation by specifying the "warning" query parameter as "yes". For example, to get warnings while connecting to the Citrix ADC appliance, the URL is as follows:

  http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/login?warning=yes

  If any, the warnings are displayed in the response payload with the HTTP code "209 X-NITRO-WARNING".

- **Authenticated access for individual NITRO operations:**NITRO allows you to logon to the Citrix ADC appliance to perform individual operations. You can use this option instead of creating a NITRO session (using the login object) and then using that session to perform all operations,

  To do this, you must specify the username and password in the request header of the NITRO request as follows:

  X-NITRO-USER:<username>

X-NITRO-PASS:<password>

Note: In such cases, make sure that the request header DOES not include the following:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue>

`*Note:`*

Mandatory parameters are marked in redand placeholder content is marked in <green>.

get

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/authenticationnegotiatepolicy_binding/name_value<String>

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "authenticationnegotiatepolicy_binding": [ {

"name":<String_value>,

"authenticationnegotiatepolicy_vpnglobal_binding":<authenticationnegotiate
policy_vpnglobal_binding[]_value>,

"authenticationnegotiatepolicy_authenticationvserver_binding":<authenticat
ionnegotiatepolicy_authenticationvserver_binding[]_value>,

"authenticationnegotiatepolicy_vpnvserver_binding":<authenticationnegotiat
epolicy_vpnvserver_binding[]_value>
```

```
}]}
```

get (all)

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/authenticationnegotiatepolicy_binding

Query-parameters:

bulkbindings

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/authenticationnegotiatepolicy_binding?**bulkbindings=yes**

NITRO allows you to fetch bindings in bulk.

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "authenticationnegotiatepolicy_binding": [ {

"name":<String_value>,

"authenticationnegotiatepolicy_vpnglobal_binding":<authenticationnegotiate
policy_vpnglobal_binding[]_value>,

"authenticationnegotiatepolicy_authenticationvserver_binding":<authenticat
ionnegotiatepolicy_authenticationvserver_binding[]_value>,
```

```
"authenticationnegotiatepolicy_vpnvserver_binding":<authenticationnegotiatepolicy_vpnvserver_binding[]_value>

}]}
```

authenticationnegotiatepolicy_vpnglobal_binding

Binding object showing the vpnglobal that can be bound to authenticationnegotiatepolicy.

Properties

(click to see Operations)

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
boundto	<String>	Read-write	The entity name to which policy is bound.
name	<String>	Read-write	Name of the negotiate policy. Minimum length = 1
priority	<Double>	Read-only	.
activepolicy	<Double>	Read-only	.
__count	<Double>	Read-only	count parameter

Operations

(click to see Properties)

GET| GET (ALL)| COUNT

Some options that you can use for each operations:

- **Getting warnings in response:**NITRO allows you to get warnings in an operation by specifying the "warning" query parameter as "yes". For example, to get warnings while connecting to the Citrix ADC appliance, the URL is as follows:

  http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/login?warning=yes

If any, the warnings are displayed in the response payload with the HTTP code "209 X-NITRO-WARNING".

- **Authenticated access for individual NITRO operations:**NITRO allows you to logon to the Citrix ADC appliance to perform individual operations. You can use this option instead of creating a NITRO session (using the login object) and then using that session to perform all operations,

  To do this, you must specify the username and password in the request header of the NITRO request as follows:

  X-NITRO-USER:<username>

  X-NITRO-PASS:<password>

  Note: In such cases, make sure that the request header DOES not include the following:

  Cookie:NITRO_AUTH_TOKEN=<tokenvalue>

`*Note:`*

Mandatory parameters are marked in redand placeholder content is marked in <green>.

get

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/authenticationnegotiatepolicy_vpnglobal_binding/name_value<String>

Query-parameters:

filter

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/authenticationnegotiatepolicy_vpnglobal_binding/name_value<String>?**filter=property-name1:property-value1,property-name2:property-value2**

Use this query-parameter to get the filtered set of authenticationnegotiatepolicy_vpnglobal_binding resources configured on Citrix ADC. Filtering can be done on any of the properties of the resource.

pagination

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/authenticationnegotiatepolicy_vpnglobal_binding/name_value<String>?**pagesize=#no;pageno=#no**

Use this query-parameter to get the authenticationnegotiatepolicy_vpnglobal_binding resources in chunks.

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "authenticationnegotiatepolicy_vpnglobal_binding": [ {

"boundto":<String_value>,

"name":<String_value>,

"priority":<Double_value>,

"activepolicy":<Double_value>

}]}
```

get (all)

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/authenticationnegotiatepolicy_vpnglobal_binding

Query-parameters:

bulkbindings

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/authenticationnegotiatepolicy_vpnglobal_binding?**bulkbindings=yes**

NITRO allows you to fetch bindings in bulk.

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "authenticationnegotiatepolicy_vpnglobal_binding": [ {

"boundto":<String_value>,

"name":<String_value>,

"priority":<Double_value>,

"activepolicy":<Double_value>

}]}
```

count

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/authenticationnegotiatepolicy_vpnglobal_binding/name_value<String>?**count=yes**

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{"authenticationnegotiatepolicy_vpnglobal_binding": [ { "__count":
"#no"} ] }
```

authenticationnegotiatepolicy_vpnvserver_binding

Binding object showing the vpnvserver that can be bound to authenticationnegotiatepolicy.

Properties

(click to see Operations)

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
boundto	<String>	Read-write	The entity name to which policy is bound.
name	<String>	Read-write	Name of the negotiate policy. Minimum length = 1
priority	<Double>	Read-only	.
activepolicy	<Double>	Read-only	.
__count	<Double>	Read-only	count parameter

Operations

(click to see Properties)

GET| GET (ALL)| COUNT

Some options that you can use for each operations:

- **Getting warnings in response:**NITRO allows you to get warnings in an operation by specifying the "warning" query parameter as "yes". For example, to get warnings while connecting to the Citrix ADC appliance, the URL is as follows:

  http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/login?warning=yes

  If any, the warnings are displayed in the response payload with the HTTP code "209 X-NITRO-WARNING".

- **Authenticated access for individual NITRO operations:**NITRO allows you to logon to the Citrix ADC appliance to perform individual operations. You can use this option instead of creating a NITRO session (using the login object) and then using that session to perform all operations,

  To do this, you must specify the username and password in the request header of the NITRO request as follows:

  X-NITRO-USER:<username>

  X-NITRO-PASS:<password>

  Note: In such cases, make sure that the request header DOES not include the following:

  Cookie:NITRO_AUTH_TOKEN=<tokenvalue>

`*Note:`*

Mandatory parameters are marked in redand placeholder content is marked in <green>.

get

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/authenticationnegotiatepolicy_vpnvserver_binding/name_value<String>

Query-parameters:

filter

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/authenticationnegotiatepolicy_vpnvserver_binding/name_value<String>?**filter=property-name1:property-value1,property-name2:property-value2**

Use this query-parameter to get the filtered set of authenticationnegotiatepolicy_vpnvserver_binding resources configured on Citrix ADC. Filtering can be done on any of the properties of the resource.

pagination

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/authenticationnegotiatepolicy_vpnvserver_binding/name_value<String>?**pagesize=#no;pageno=#no**

Use this query-parameter to get the authenticationnegotiatepolicy_vpnvserver_binding resources in chunks.

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "authenticationnegotiatepolicy_vpnvserver_binding": [ {

"boundto":<String_value>,

"name":<String_value>,

"priority":<Double_value>,
```

```
"activepolicy":<Double_value>

}]}
```

get (all)

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/authenticationnegotiatepolicy_vpnvserver_binding

Query-parameters:

bulkbindings

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/authenticationnegotiatepolicy_vpnvserver_binding?**bulkbindings=yes**

NITRO allows you to fetch bindings in bulk.

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "authenticationnegotiatepolicy_vpnvserver_binding": [ {

"boundto":<String_value>,

"name":<String_value>,

"priority":<Double_value>,
```

```
"activepolicy":<Double_value>

}]}
```

count

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/authenticationnegotiatepolicy_vpnvserver_binding/name_value<String>?**count=yes**

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{"authenticationnegotiatepolicy_vpnvserver_binding": [ { "__count":
"#no"} ] }
```

authenticationoauthaction

Configuration for OAuth authentication action resource.

Properties

(click to see Operations)

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
name	<String>	Read-write	Name for the OAuth Authentication action. Must begin with a letter, number, or the underscore character (_), and must contain only letters, numbers, and the hyphen (-), period (.) pound (#), space ( ), at (@), equals (=), colon (:), and underscore characters. Cannot be changed after the profile is created. The following requirement applies only to the Citrix ADC CLI: If the name includes one or more spaces, enclose the name in double or single quotation marks (for example, "my authentication action" or 'my authentication action'). Minimum length = 1

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
oauthtype	<String>	Read-write	Type of the OAuth implementation. Default value is generic implementation that is applicable for most deployments. Default value: GENERIC Possible values = GENERIC, INTUNE, ATHENA
authorizationendpoint	<String>	Read-write	Authorization endpoint/url to which unauthenticated user will be redirected. Citrix ADC redirects user to this endpoint by adding query parameters including clientid. If this parameter not specified then as default value we take Token Endpoint/URL value. Please note that Authorization Endpoint or Token Endpoint is mandatory for oauthAction.
tokenendpoint	<String>	Read-write	URL to which OAuth token will be posted to verify its authenticity. User obtains this token from Authorization

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
			server upon successful authentication. Citrix ADC will validate presented token by posting it to the URL configured.
idtokendecryptendpoint	<String>	Read-write	URL to which obtained idtoken will be posted to get a decrypted user identity. Encrypted idtoken will be obtained by posting OAuth token to token endpoint. In order to decrypt idtoken, Citrix ADC posts request to the URL configured.
clientid	<String>	Read-write	Unique identity of the client/user who is getting authenticated. Authorization server infers client configuration using this ID. Minimum length = 1
clientsecret	<String>	Read-write	Secret string established by user and authorization server. Minimum length = 1
defaultauthenticationgroup	<String>	Read-write	This is the default group that is chosen when the

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
			authentication succeeds in addition to extracted groups.
attribute1	<String>	Read-write	Name of the attribute to be extracted from OAuth Token and to be stored in the attribute1.
attribute2	<String>	Read-write	Name of the attribute to be extracted from OAuth Token and to be stored in the attribute2.
attribute3	<String>	Read-write	Name of the attribute to be extracted from OAuth Token and to be stored in the attribute3.
attribute4	<String>	Read-write	Name of the attribute to be extracted from OAuth Token and to be stored in the attribute4.
attribute5	<String>	Read-write	Name of the attribute to be extracted from OAuth Token and to be stored in the attribute5.
attribute6	<String>	Read-write	Name of the attribute to be extracted from OAuth Token and to be stored in the attribute6.

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
attribute7	<String>	Read-write	Name of the attribute to be extracted from OAuth Token and to be stored in the attribute7.
attribute8	<String>	Read-write	Name of the attribute to be extracted from OAuth Token and to be stored in the attribute8.
attribute9	<String>	Read-write	Name of the attribute to be extracted from OAuth Token and to be stored in the attribute9.
attribute10	<String>	Read-write	Name of the attribute to be extracted from OAuth Token and to be stored in the attribute10.
attribute11	<String>	Read-write	Name of the attribute to be extracted from OAuth Token and to be stored in the attribute11.
attribute12	<String>	Read-write	Name of the attribute to be extracted from OAuth Token and to be stored in the attribute12.
attribute13	<String>	Read-write	Name of the attribute to be extracted from OAuth Token and to be stored in the attribute13.

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
attribute14	<String>	Read-write	Name of the attribute to be extracted from OAuth Token and to be stored in the attribute14.
attribute15	<String>	Read-write	Name of the attribute to be extracted from OAuth Token and to be stored in the attribute15.
attribute16	<String>	Read-write	Name of the attribute to be extracted from OAuth Token and to be stored in the attribute16.
tenantid	<String>	Read-write	TenantID of the application. This is usually specific to providers such as Microsoft and usually refers to the deployment identifier.
graphendpoint	<String>	Read-write	URL of the Graph API service to learn Enterprise Mobility Services (EMS) endpoints.
refreshinterval	<Double>	Read-write	Interval at which services are monitored for necessary configuration. Default value: 1440
certendpoint	<String>	Read-write	URL of the endpoint that contains JWKs

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
			(Json Web Key) for JWT (Json Web Token) verification.
audience	<String>	Read-write	Audience for which token sent by Authorization server is applicable. This is typically entity name or url that represents the recipient.
usernamefield	<String>	Read-write	Attribute in the token from which username should be extracted. Minimum length = 1
skewtime	<Double>	Read-write	This option specifies the allowed clock skew in number of minutes that Citrix ADC allows on an incoming token. For example, if skewTime is 10, then token would be valid from (current time - 10) min to (current time + 10) min, ie 20min in all. Default value: 5
issuer	<String>	Read-write	Identity of the server whose tokens are to be accepted.
userinfourl	<String>	Read-write	URL to which OAuth access token will be posted to obtain user information.

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
certfilepath	<String>	Read-write	Path to the file that contains JWKs (Json Web Key) for JWT (Json Web Token) verification.
granttype	<String>	Read-write	Grant type support. value can be code or password. Default value: CODE Possible values = CODE, PASSWORD
authentication	<String>	Read-write	If authentication is disabled, password is not sent in the request. . Default value: ENABLED Possible values = ENABLED, DISABLED
oauthstatus	<String>	Read-only	Describes status information of oauth server. Default value: INIT Possible values = INIT, CERTFETCH, AADFORGRAPH, GRAPH, AADFORMDM, MDMINFO, COMPLETE

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
__count	<Double>	Read-only	count parameter

Operations

(click to see Properties)

ADD| DELETE| UPDATE| UNSET| GET (ALL)| GET| COUNT

Some options that you can use for each operations:

- **Getting warnings in response:**NITRO allows you to get warnings in an operation by specifying the "warning" query parameter as "yes". For example, to get warnings while connecting to the Citrix ADC appliance, the URL is as follows:

  http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/login?warning=yes

  If any, the warnings are displayed in the response payload with the HTTP code "209 X-NITRO-WARNING".

- **Authenticated access for individual NITRO operations:**NITRO allows you to logon to the Citrix ADC appliance to perform individual operations. You can use this option instead of creating a NITRO session (using the login object) and then using that session to perform all operations,

  To do this, you must specify the username and password in the request header of the NITRO request as follows:

  X-NITRO-USER:<username>

  X-NITRO-PASS:<password>

  Note: In such cases, make sure that the request header DOES not include the following:

  Cookie:NITRO_AUTH_TOKEN=<tokenvalue>

`*Note:`*

Mandatory parameters are marked in redand placeholder content is marked in <green>.

add

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/authenticationoauthaction

HTTP Method:POST

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Content-Type:application/json

Request Payload:

```
{"authenticationoauthaction":{

<b>"name":<String_value>,

</b>"oauthtype":<String_value>,

"authorizationendpoint":<String_value>,

"tokenendpoint":<String_value>,

"idtokendecryptendpoint":<String_value>,

<b>"clientid":<String_value>,

</b><b>"clientsecret":<String_value>,

</b>"defaultauthenticationgroup":<String_value>,

"attribute1":<String_value>,

"attribute2":<String_value>,

"attribute3":<String_value>,

"attribute4":<String_value>,

"attribute5":<String_value>,

"attribute6":<String_value>,

"attribute7":<String_value>,

"attribute8":<String_value>,

"attribute9":<String_value>,
```

```
"attribute10":<String_value>,

"attribute11":<String_value>,

"attribute12":<String_value>,

"attribute13":<String_value>,

"attribute14":<String_value>,

"attribute15":<String_value>,

"attribute16":<String_value>,

"tenantid":<String_value>,

"graphendpoint":<String_value>,

"refreshinterval":<Double_value>,

"certendpoint":<String_value>,

"audience":<String_value>,

"usernamefield":<String_value>,

"skewtime":<Double_value>,

"issuer":<String_value>,

"userinfourl":<String_value>,

"certfilepath":<String_value>,

"granttype":<String_value>,

"authentication":<String_value>

}}
```

Response:

HTTP Status Code on Success: 201 Created HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error

delete

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/authenticationoauthaction/name_value<String>

HTTP Method:DELETE

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue>

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error

update

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/authenticationoauthaction

HTTP Method:PUT

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Content-Type:application/json

Request Payload:

```
{"authenticationoauthaction":{

<b>"name":<String_value>,

</b>"oauthtype":<String_value>,

"authorizationendpoint":<String_value>,

"tokenendpoint":<String_value>,
```

```
"idtokendecryptendpoint":<String_value>,

"clientid":<String_value>,

"clientsecret":<String_value>,

"defaultauthenticationgroup":<String_value>,

"attribute1":<String_value>,

"attribute2":<String_value>,

"attribute3":<String_value>,

"attribute4":<String_value>,

"attribute5":<String_value>,

"attribute6":<String_value>,

"attribute7":<String_value>,

"attribute8":<String_value>,

"attribute9":<String_value>,

"attribute10":<String_value>,

"attribute11":<String_value>,

"attribute12":<String_value>,

"attribute13":<String_value>,

"attribute14":<String_value>,

"attribute15":<String_value>,

"attribute16":<String_value>,

"tenantid":<String_value>,

"graphendpoint":<String_value>,
```

```
"refreshinterval":<Double_value>,

"certendpoint":<String_value>,

"audience":<String_value>,

"usernamefield":<String_value>,

"skewtime":<Double_value>,

"issuer":<String_value>,

"userinfourl":<String_value>,

"certfilepath":<String_value>,

"granttype":<String_value>,

"authentication":<String_value>

}}
```

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error

unset

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/authenticationoauthaction?**action=unset**

HTTP Method:POST

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Content-Type:application/json

Request Payload:

```
{"authenticationoauthaction":{

<b>"name":<String_value>,

</b>"oauthtype":true,

"idtokendecryptendpoint":true,

"defaultauthenticationgroup":true,

"attribute1":true,

"attribute2":true,

"attribute3":true,

"attribute4":true,

"attribute5":true,

"attribute6":true,

"attribute7":true,

"attribute8":true,

"attribute9":true,

"attribute10":true,

"attribute11":true,

"attribute12":true,

"attribute13":true,

"attribute14":true,

"attribute15":true,

"attribute16":true,
```

```
"graphendpoint":true,

"refreshinterval":true,

"certendpoint":true,

"audience":true,

"usernamefield":true,

"skewtime":true,

"issuer":true,

"userinfourl":true,

"certfilepath":true,

"authentication":true

}}
```

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error

get (all)

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/authenticationoauthaction

Query-parameters:

attrs

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/authenticationoauthaction?**attrs=property-name1,property-name2**

Use this query parameter to specify the resource details that you want to retrieve.

filter

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/authenticationoauthaction?**filter=property-name1:property-val1,property-name2:property-val2**

Use this query-parameter to get the filtered set of authenticationoauthaction resources configured on Citrix ADC. Filtering can be done on any of the properties of the resource.

view

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/authenticationoauthaction?**view=summary**

Note: By default, the retrieved results are displayed in detail view (?view=detail).

pagination

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/authenticationoauthaction?**pagesize=#no;pageno=#no**

Use this query-parameter to get the authenticationoauthaction resources in chunks.

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "authenticationoauthaction": [ {

"name":<String_value>,

"authorizationendpoint":<String_value>,

"tokenendpoint":<String_value>,
```

```
"idtokendecryptendpoint":<String_value>,

"clientid":<String_value>,

"clientsecret":<String_value>,

"defaultauthenticationgroup":<String_value>,

"attribute1":<String_value>,

"attribute2":<String_value>,

"attribute3":<String_value>,

"attribute4":<String_value>,

"attribute5":<String_value>,

"attribute6":<String_value>,

"attribute7":<String_value>,

"attribute8":<String_value>,

"attribute9":<String_value>,

"attribute10":<String_value>,

"attribute11":<String_value>,

"attribute12":<String_value>,

"attribute13":<String_value>,

"attribute14":<String_value>,

"attribute15":<String_value>,

"attribute16":<String_value>,

"tenantid":<String_value>,

"graphendpoint":<String_value>,
```

```
"refreshinterval":<Double_value>,

"certendpoint":<String_value>,

"oauthtype":<String_value>,

"audience":<String_value>,

"usernamefield":<String_value>,

"skewtime":<Double_value>,

"issuer":<String_value>,

"oauthstatus":<String_value>,

"userinfourl":<String_value>,

"certfilepath":<String_value>,

"granttype":<String_value>,

"authentication":<String_value>

}]}
```

get

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/authenticationoauthaction/name_value<String>

Query-parameters:

attrs

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/authenticationoauthaction/name_value<String>?**attrs=property-name1,property-name2**

Use this query parameter to specify the resource details that you want to retrieve.

view

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/authenticationoauthaction/name_value<String>?**view=summary**

Note: By default, the retrieved results are displayed in detail view (?view=detail).

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "authenticationoauthaction": [ {

"name":<String_value>,

"authorizationendpoint":<String_value>,

"tokenendpoint":<String_value>,

"idtokendecryptendpoint":<String_value>,

"clientid":<String_value>,

"clientsecret":<String_value>,

"defaultauthenticationgroup":<String_value>,

"attribute1":<String_value>,

"attribute2":<String_value>,

"attribute3":<String_value>,
```

```
"attribute4":<String_value>,

"attribute5":<String_value>,

"attribute6":<String_value>,

"attribute7":<String_value>,

"attribute8":<String_value>,

"attribute9":<String_value>,

"attribute10":<String_value>,

"attribute11":<String_value>,

"attribute12":<String_value>,

"attribute13":<String_value>,

"attribute14":<String_value>,

"attribute15":<String_value>,

"attribute16":<String_value>,

"tenantid":<String_value>,

"graphendpoint":<String_value>,

"refreshinterval":<Double_value>,

"certendpoint":<String_value>,

"oauthtype":<String_value>,

"audience":<String_value>,

"usernamefield":<String_value>,

"skewtime":<Double_value>,

"issuer":<String_value>,
```

```
"oauthstatus":<String_value>,

"userinfourl":<String_value>,

"certfilepath":<String_value>,

"granttype":<String_value>,

"authentication":<String_value>

}]}
```

count

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/authenticationoauthaction?**count=yes**

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "authenticationoauthaction": [ { "__count": "#no"} ] }
```

authenticationoauthidppolicy

Configuration for AAA OAuth IdentityProvider (IdP) policy resource.

Properties

(click to see Operations)

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
name	<String>	Read-write	Name for the OAuth Identity Provider (IdP) authentication policy. This is used for configuring Citrix ADC as OAuth Identity Provider. Must begin with an ASCII alphanumeric or underscore (_) character, and must contain only ASCII alphanumeric, underscore, hash (#), period (.), space, colon (:), at (@), equals (=), and hyphen (-) characters. The following requirement applies only to the Citrix ADC CLI: If the name includes one or more spaces, enclose the name in double or single quotation marks (for example, "my policy" or 'my policy'). Minimum length = 1

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
rule	<String>	Read-write	Expression that the policy uses to determine whether to respond to the specified request. Minimum length = 1
action	<String>	Read-write	Name of the profile to apply to requests or connections that match this policy. Minimum length = 1
undefaction	<String>	Read-write	Action to perform if the result of policy evaluation is undefined (UNDEF). An UNDEF event indicates an internal error condition. Only DROP/RESET actions can be used.
comment	<String>	Read-write	Any comments to preserve information about this policy.
logaction	<String>	Read-write	Name of messagelog action to use when a request matches this policy.
newname	<String>	Read-write	New name for the OAuth IdentityProvider policy. Must begin with a letter, number, or the underscore character (_), and

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
			must contain only letters, numbers, and the hyphen (-), period (.) hash (#), space ( ), at (@), equals (=), colon (:), and underscore characters. The following requirement applies only to the Citrix ADC CLI: If the name includes one or more spaces, enclose the name in double or single quotation marks (for example, "my oauthidppolicy policy" or 'my oauthidppolicy policy'). Minimum length = 1
gotopriorityexpression	<String>	Read-only	Expression specifying the priority of the next policy which will get evaluated if the current policy rule evaluates to TRUE.
hits	<Double>	Read-only	Number of hits.
__count	<Double>	Read-only	count parameter

Operations

(click to see Properties)

ADD| DELETE| UPDATE| UNSET| RENAME| GET (ALL)| GET| COUNT

Some options that you can use for each operations:

- **Getting warnings in response:**NITRO allows you to get warnings in an operation by specifying the "warning" query parameter as "yes". For example, to get warnings while connecting to the Citrix ADC appliance, the URL is as follows:

  http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/login?warning=yes

  If any, the warnings are displayed in the response payload with the HTTP code "209 X-NITRO-WARNING".

- **Authenticated access for individual NITRO operations:**NITRO allows you to logon to the Citrix ADC appliance to perform individual operations. You can use this option instead of creating a NITRO session (using the login object) and then using that session to perform all operations,

  To do this, you must specify the username and password in the request header of the NITRO request as follows:

  X-NITRO-USER:<username>

  X-NITRO-PASS:<password>

  Note: In such cases, make sure that the request header DOES not include the following:

  Cookie:NITRO_AUTH_TOKEN=<tokenvalue>

`*Note:`*

Mandatory parameters are marked in redand placeholder content is marked in <green>.

add

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/authenticationoauthidppolicy

HTTP Method:POST

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Content-Type:application/json

Request Payload:

```
{"authenticationoauthidppolicy":{

<b>"name":<String_value>,

</b><b>"rule":<String_value>,

</b><b>"action":<String_value>,

</b>"undefaction":<String_value>,

"comment":<String_value>,

"logaction":<String_value>

}}
```

Response:

HTTP Status Code on Success: 201 Created HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error

delete

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/authenticationoauthidppolicy/name_value<String>

HTTP Method:DELETE

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue>

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error

update

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/authenticationoauthidppolicy

HTTP Method:PUT

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Content-Type:application/json

Request Payload:

```
{"authenticationoauthidppolicy":{

<b>"name":<String_value>,

</b>"rule":<String_value>,

"action":<String_value>,

"undefaction":<String_value>,

"comment":<String_value>,

"logaction":<String_value>

}}
```

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error

unset

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/authenticationoauthidppolicy?**action=unset**

HTTP Method:POST

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Content-Type:application/json

Request Payload:

```
{"authenticationoauthidppolicy":{

<b>"name":<String_value>,

</b>"undefaction":true,

"comment":true,

"logaction":true

}}
```

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error

rename

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/authenticationoauthidppolicy?**action=rename**

HTTP Method:POST

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Content-Type:application/json

Request Payload:

```
{"authenticationoauthidppolicy":{

<b>"name":<String_value>,
```

```
</b><b>"newname":<String_value>

</b>}}
```

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error

get (all)

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/authenticationoauthidppolicy

Query-parameters:

attrs

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/authenticationoauthidppolicy?**attrs=property-name1,property-name2**

Use this query parameter to specify the resource details that you want to retrieve.

filter

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/authenticationoauthidppolicy?**filter=property-name1:property-val1,property-name2:property-val2**

Use this query-parameter to get the filtered set of authenticationoauthidppolicy resources configured on Citrix ADC. Filtering can be done on any of the properties of the resource.

view

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/authenticationoauthidppolicy?**view=summary**

Note: By default, the retrieved results are displayed in detail view (?view=detail).

pagination

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/authenticationoauthidppolicy?**pagesize=#no;pageno=#no**

Use this query-parameter to get the authenticationoauthidppolicy resources in chunks.

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "authenticationoauthidppolicy": [ {

"name":<String_value>,

"rule":<String_value>,

"action":<String_value>,

"undefaction":<String_value>,

"comment":<String_value>,

"logaction":<String_value>,

"gotopriorityexpression":<String_value>,

"hits":<Double_value>

}]}
```

get

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/authenticationoauthidppolicy/name_value<String>

Query-parameters:

attrs

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/authenticationoauthidppolicy/name_value<String>?**attrs=property-name1,property-name2**

Use this query parameter to specify the resource details that you want to retrieve.

view

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/authenticationoauthidppolicy/name_value<String>?**view=summary**

Note: By default, the retrieved results are displayed in detail view (?view=detail).

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "authenticationoauthidppolicy": [ {

"name":<String_value>,

"rule":<String_value>,

"action":<String_value>,

"undefaction":<String_value>,

"comment":<String_value>,
```

```
"logaction":<String_value>,

"gotopriorityexpression":<String_value>,

"hits":<Double_value>

}]}
```

count

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/authenticationoauthidppolicy?**count=yes**

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "authenticationoauthidppolicy": [ { "__count": "#no"} ] }
```

authenticationoauthidppolicy_authenticationvserver_binding

Binding object showing the authenticationvserver that can be bound to authenticationoauthidppolicy.

Properties

(click to see Operations)

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
name	<String>	Read-write	Name of the OAuth IdentityProvider (IdP) policy for which to display detailed information. Minimum length = 1
boundto	<String>	Read-write	The entity name to which policy is bound.
priority	<Double>	Read-only	.
activepolicy	<Double>	Read-only	.
__count	<Double>	Read-only	count parameter

Operations

(click to see Properties)

GET| GET (ALL)| COUNT

Some options that you can use for each operations:

- **Getting warnings in response:**NITRO allows you to get warnings in an operation by specifying the "warning" query parameter as "yes". For example, to get warnings while connecting to the Citrix ADC appliance, the URL is as follows:

  http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/login?warning=yes

  If any, the warnings are displayed in the response payload with the HTTP code "209 X-NITRO-WARNING".

- **Authenticated access for individual NITRO operations:**NITRO allows you to logon to the Citrix ADC appliance to perform individual operations. You can use this option instead of creating a NITRO session (using the login object) and then using that session to perform all operations,

  To do this, you must specify the username and password in the request header of the NITRO request as follows:

X-NITRO-USER:<username>

X-NITRO-PASS:<password>

Note: In such cases, make sure that the request header DOES not include the following:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue>

`*Note:`*

Mandatory parameters are marked in redand placeholder content is marked in <green>.

get

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/authenticationoauthidppolicy_authenticationvserver_binding/name_value<String>

Query-parameters:

filter

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/authenticationoauthidppolicy_authenticationvserver_binding/name_value<String>?**filter=property-name1:property-value1,property-name2:property-value2**

Use this query-parameter to get the filtered set of authenticationoauthidppolicy_authenticationvserver_binding resources configured on Citrix ADC. Filtering can be done on any of the properties of the resource.

pagination

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/authenticationoauthidppolicy_authenticationvserver_binding/name_value<String>?**pagesize=#no;pageno=#no**

Use this query-parameter to get the authenticationoauthidppolicy_authenticationvserver_binding resources in chunks.

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "authenticationoauthidppolicy_authenticationvserver_binding": [ {

"name":<String_value>,

"boundto":<String_value>,

"priority":<Double_value>,

"activepolicy":<Double_value>

}]}
```

get (all)

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/authenticationoauthidppolicy_authenticationvserver_binding

Query-parameters:

bulkbindings

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/authenticationoauthidppolicy_authenticationvserver_binding?**bulkbindings=yes**

NITRO allows you to fetch bindings in bulk.

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "authenticationoauthidppolicy_authenticationvserver_binding": [ {

"name":<String_value>,

"boundto":<String_value>,

"priority":<Double_value>,

"activepolicy":<Double_value>

}]}
```

count

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/authenticationoauthidppolicy_authenticationvserver_binding/name_value<String>?**count=yes**

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{"authenticationoauthidppolicy_authenticationvserver_binding":
[ { "__count": "#no"} ] }
```

authenticationoauthidppolicy_binding

Binding object which returns the resources bound to authenticationoauthidppolicy.

Properties

(click to see Operations)

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
name	<String>	Read-write	Name of the OAuth IdentityProvider (IdP) policy for which to display detailed information. Minimum length = 1
authenticationoauthidppolicy_vpnvserver_binding	<authenticationoauthidppolicy_vpnvserver_binding[]>	Read-only	vpnvserver that can be bound to authenticationoauthidppolicy.
authenticationoauthidppolicy_authenticationvserver_binding	<authenticationoauthidppolicy_authenticationvserver_binding[]>	Read-only	authenticationvserver that can be bound to authenticationoauthidppolicy.

Operations

(click to see Properties)

GET| GET (ALL)

Some options that you can use for each operations:

- **Getting warnings in response:**NITRO allows you to get warnings in an operation by specifying the "warning" query parameter as "yes". For example, to get warnings while connecting to the Citrix ADC appliance, the URL is as follows:

  http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/login?warning=yes

  If any, the warnings are displayed in the response payload with the HTTP code "209 X-NITRO-WARNING".

- **Authenticated access for individual NITRO operations:**NITRO allows you to logon to the Citrix ADC appliance to perform individual operations. You can use this option instead of creating a NITRO session (using the login object) and then using that session to perform all operations,

  To do this, you must specify the username and password in the request header of the NITRO request as follows:

  X-NITRO-USER:<username>

  X-NITRO-PASS:<password>

  Note: In such cases, make sure that the request header DOES not include the following:

  Cookie:NITRO_AUTH_TOKEN=<tokenvalue>

`*Note:`*

Mandatory parameters are marked in redand placeholder content is marked in <green>.

get

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/authenticationoauthidppolicy_binding/name_value<String>

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "authenticationoauthidppolicy_binding": [ {

"name":<String_value>,

"authenticationoauthidppolicy_vpnvserver_binding":<authenticationoauthidpp
olicy_vpnvserver_binding[]_value>,

"authenticationoauthidppolicy_authenticationvserver_binding":<authenticati
onoauthidppolicy_authenticationvserver_binding[]_value>

}]}
```

get (all)

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/authenticationoauthidppolicy_binding

Query-parameters:

bulkbindings

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/authenticationoauthidppolicy_binding?**bulkbindings=yes**

NITRO allows you to fetch bindings in bulk.

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "authenticationoauthidppolicy_binding": [ {

"name":<String_value>,

"authenticationoauthidppolicy_vpnvserver_binding":<authenticationoauthidpp
olicy_vpnvserver_binding[]_value>,

"authenticationoauthidppolicy_authenticationvserver_binding":<authenticati
onoauthidppolicy_authenticationvserver_binding[]_value>

}]}
```

authenticationoauthidppolicy_vpnvserver_binding

Binding object showing the vpnvserver that can be bound to authenticationoauthidppolicy.

Properties

(click to see Operations)

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
name	<String>	Read-write	Name of the OAuth IdentityProvider (IdP) policy for which to display detailed information. Minimum length = 1
boundto	<String>	Read-write	The entity name to which policy is bound.
priority	<Double>	Read-only	.
activepolicy	<Double>	Read-only	.
__count	<Double>	Read-only	count parameter

Operations

(click to see Properties)

GET| GET (ALL)| COUNT

Some options that you can use for each operations:

- **Getting warnings in response:**NITRO allows you to get warnings in an operation by specifying the "warning" query parameter as "yes". For example, to get warnings while connecting to the Citrix ADC appliance, the URL is as follows:

  http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/login?warning=yes

  If any, the warnings are displayed in the response payload with the HTTP code "209 X-NITRO-WARNING".

- **Authenticated access for individual NITRO operations:**NITRO allows you to logon to the Citrix ADC appliance to perform individual operations. You can use this option instead of creating a NITRO session (using the login object) and then using that session to perform all operations,

  To do this, you must specify the username and password in the request header of the NITRO request as follows:

  X-NITRO-USER:<username>

  X-NITRO-PASS:<password>

  Note: In such cases, make sure that the request header DOES not include the following:

  Cookie:NITRO_AUTH_TOKEN=<tokenvalue>

`*Note:`*

Mandatory parameters are marked in redand placeholder content is marked in <green>.

get

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/authenticationoauthidppolicy_vpnvserver_binding/name_value<String>

Query-parameters:

filter

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/authenticationoauthidppolicy_vpnvserver_binding/name_value<String>?**filter=property-name1:property-value1,property-name2:property-value2**

Use this query-parameter to get the filtered set of authenticationoauthidppolicy_vpnvserver_binding resources configured on Citrix ADC. Filtering can be done on any of the properties of the resource.

pagination

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/authenticationoauthidppolicy_vpnvserver_binding/name_value<String>?**pagesize=#no;pageno=#no**

Use this query-parameter to get the authenticationoauthidppolicy_vpnvserver_binding resources in chunks.

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "authenticationoauthidppolicy_vpnvserver_binding": [ {

"name":<String_value>,

"boundto":<String_value>,

"priority":<Double_value>,
```

```
"activepolicy":<Double_value>

}]}
```

get (all)

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/authenticationoauthidppolicy_vpnvserver_binding

Query-parameters:

bulkbindings

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/authenticationoauthidppolicy_vpnvserver_binding?**bulkbindings=yes**

NITRO allows you to fetch bindings in bulk.

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "authenticationoauthidppolicy_vpnvserver_binding": [ {

"name":<String_value>,

"boundto":<String_value>,

"priority":<Double_value>,
```

```
"activepolicy":<Double_value>

}]}
```

count

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/authenticationoauthidppolicy_vpnvserver_binding/name_value<String>?**count=yes**

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{"authenticationoauthidppolicy_vpnvserver_binding": [ { "__count":
"#no"} ] }
```

authenticationoauthidpprofile

Configuration for OAuth Identity Provider (IdP) profile resource.

Properties

(click to see Operations)

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
name	<String>	Read-write	Name for the new OAuth Identity Provider (IdP) single sign-on profile. Must begin with an ASCII alphanumeric or underscore (_) character, and must contain only ASCII alphanumeric, underscore, hash (#), period (.), space, colon (:), at (@), equals (=), and hyphen (-) characters. Cannot be changed after an action is created. The following requirement applies only to the Citrix ADC CLI: If the name includes one or more spaces, enclose the name in double or single quotation marks (for example, "my action" or 'my action'). Minimum length = 1
clientid	<String>	Read-write	Unique identity of the relying party requesting for authentication. Minimum length = 1

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
clientsecret	<String>	Read-write	Unique secret string to authorize relying party at authorization server. Minimum length = 1
redirecturl	<String>	Read-write	URL endpoint on relying party to which the OAuth token is to be sent. Minimum length = 1
issuer	<String>	Read-write	The name to be used in requests sent from Citrix ADC to IdP to uniquely identify Citrix ADC. Minimum length = 1
configservice	<String>	Read-write	Name of the entity that is used to obtain configuration for the current authentication request. It is used only in Citrix Cloud. Minimum length = 1
audience	<String>	Read-write	Audience for which token is being sent by Citrix ADC IdP. This is typically entity name or url that represents the recipient.
skewtime	<Double>	Read-write	This option specifies the duration for which the token sent by Citrix ADC IdP is valid. For example, if

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
			skewTime is 10, then token would be valid from (current time - 10) min to (current time + 10) min, ie 20min in all. Default value: 5
defaultauthentication group	<String>	Read-write	This is the group that is added to user sessions that match current IdP policy. It can be used in policies to identify relying party trust.
relyingpartymetadata url	<String>	Read-write	This is the endpoint at which Citrix ADC IdP can get details about Relying Party (RP) being configured. Metadata response should include endpoints for jwks_uri for RP public key(s).
refreshinterval	<Double>	Read-write	Interval at which Relying Party metadata is refreshed. Default value: 50
encrypttoken	<String>	Read-write	Option to encrypt token when Citrix ADC IDP sends one. Default value: OFF Possible values = ON, OFF

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
signatureservice	<String>	Read-write	Name of the service in cloud used to sign the data. This is applicable only if signature if offloaded to cloud. Minimum length = 1
signaturealg	<String>	Read-write	Algorithm to be used to sign OpenID tokens. Default value: RS256 Possible values = RS256, RS512
attributes	<String>	Read-write	Name-Value pairs of attributes to be inserted in idtoken. Configuration format is name=value_expr@@@name2=value2_expr@@@. '@@@' is used as delimiter between Name-Value pairs. name is a literal string whose value is 127 characters and does not contain '=' character. Value is advanced policy expression terminated by @@@ delimiter.

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
			Last value need not contain the delimiter.
sendpassword	<String>	Read-write	Option to send encrypted password in idtoken. Default value: OFF Possible values = ON, OFF
oauthstatus	<String>	Read-only	Describes status information of oauth idp metadata fetch process. Default value: INIT Possible values = INIT, CERTFETCH, AADFORGRAPH, GRAPH, AADFORMDM, MDMINFO, COMPLETE
__count	<Double>	Read-only	count parameter

Operations

(click to see Properties)

ADD| DELETE| UPDATE| UNSET| GET (ALL)| GET| COUNT

Some options that you can use for each operations:

- **Getting warnings in response:**NITRO allows you to get warnings in an operation by specifying the "warning" query parameter as "yes". For example, to get warnings while connecting to the Citrix ADC appliance, the URL is as follows:

  http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/login?warning=yes

If any, the warnings are displayed in the response payload with the HTTP code "209 X-NITRO-WARNING".

- **Authenticated access for individual NITRO operations:**NITRO allows you to logon to the Citrix ADC appliance to perform individual operations. You can use this option instead of creating a NITRO session (using the login object) and then using that session to perform all operations,

  To do this, you must specify the username and password in the request header of the NITRO request as follows:

  X-NITRO-USER:<username>

  X-NITRO-PASS:<password>

  Note: In such cases, make sure that the request header DOES not include the following:

  Cookie:NITRO_AUTH_TOKEN=<tokenvalue>

`*Note:`*

Mandatory parameters are marked in redand placeholder content is marked in <green>.

add

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/authenticationoauthidpprofile

HTTP Method:POST

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Content-Type:application/json

Request Payload:

```
{"authenticationoauthidpprofile":{

<b>"name":<String_value>,

</b>"clientid":<String_value>,

"clientsecret":<String_value>,
```

```
"redirecturl":<String_value>,

"issuer":<String_value>,

"configservice":<String_value>,

"audience":<String_value>,

"skewtime":<Double_value>,

"defaultauthenticationgroup":<String_value>,

"relyingpartymetadataurl":<String_value>,

"refreshinterval":<Double_value>,

"encrypttoken":<String_value>,

"signatureservice":<String_value>,

"signaturealg":<String_value>,

"attributes":<String_value>,

"sendpassword":<String_value>

}}
```

Response:

HTTP Status Code on Success: 201 Created HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error

delete

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/authenticationoauthidpprofile/name_value<String>

HTTP Method:DELETE

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue>

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error

update

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/authenticationoauthidpprofile

HTTP Method:PUT

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Content-Type:application/json

Request Payload:

```
{"authenticationoauthidpprofile":{

<b>"name":<String_value>,

</b>"clientid":<String_value>,

"clientsecret":<String_value>,

"redirecturl":<String_value>,

"issuer":<String_value>,

"configservice":<String_value>,

"audience":<String_value>,

"skewtime":<Double_value>,

"defaultauthenticationgroup":<String_value>,

"relyingpartymetadataurl":<String_value>,

"refreshinterval":<Double_value>,
```

```
"encrypttoken":<String_value>,

"signatureservice":<String_value>,

"signaturealg":<String_value>,

"attributes":<String_value>,

"sendpassword":<String_value>

}}
```

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error

unset

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/authenticationoauthidpprofile?**action=unset**

HTTP Method:POST

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Content-Type:application/json

Request Payload:

```
{"authenticationoauthidpprofile":{

<b>"name":<String_value>,

</b>"issuer":true,

"configservice":true,

"audience":true,
```

```
"skewtime":true,

"defaultauthenticationgroup":true,

"relyingpartymetadataurl":true,

"refreshinterval":true,

"encrypttoken":true,

"signatureservice":true,

"signaturealg":true,

"attributes":true,

"sendpassword":true

}}
```

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error

get (all)

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/authenticationoauthidpprofile

Query-parameters:

attrs

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/authenticationoauthidpprofile?**attrs=property-name1,property-name2**

Use this query parameter to specify the resource details that you want to retrieve.

filter

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/authenticationoauthidpprofile?**filter=property-name1:property-val1,property-name2:property-val2**

Use this query-parameter to get the filtered set of authenticationoauthidpprofile resources configured on Citrix ADC. Filtering can be done on any of the properties of the resource.

view

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/authenticationoauthidpprofile?**view=summary**

Note: By default, the retrieved results are displayed in detail view (?view=detail).

pagination

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/authenticationoauthidpprofile?**pagesize=#no;pageno=#no**

Use this query-parameter to get the authenticationoauthidpprofile resources in chunks.

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "authenticationoauthidpprofile": [ {

"name":<String_value>,

"clientid":<String_value>,
```

```
"clientsecret":<String_value>,

"redirecturl":<String_value>,

"issuer":<String_value>,

"configservice":<String_value>,

"audience":<String_value>,

"skewtime":<Double_value>,

"defaultauthenticationgroup":<String_value>,

"relyingpartymetadataurl":<String_value>,

"refreshinterval":<Double_value>,

"encrypttoken":<String_value>,

"signatureservice":<String_value>,

"signaturealg":<String_value>,

"oauthstatus":<String_value>,

"attributes":<String_value>,

"sendpassword":<String_value>

}]}
```

get

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/authenticationoauthidpprofile/name_value<String>

Query-parameters:

attrs

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/authenticationoauthidpprofile/name_value<String>?**attrs=property-name1,property-name2**

Use this query parameter to specify the resource details that you want to retrieve.

view

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/authenticationoauthidpprofile/name_value<String>?**view=summary**

Note: By default, the retrieved results are displayed in detail view (?view=detail).

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "authenticationoauthidpprofile": [ {

"name":<String_value>,

"clientid":<String_value>,

"clientsecret":<String_value>,

"redirecturl":<String_value>,

"issuer":<String_value>,

"configservice":<String_value>,
```

```
"audience":<String_value>,

"skewtime":<Double_value>,

"defaultauthenticationgroup":<String_value>,

"relyingpartymetadataurl":<String_value>,

"refreshinterval":<Double_value>,

"encrypttoken":<String_value>,

"signatureservice":<String_value>,

"signaturealg":<String_value>,

"oauthstatus":<String_value>,

"attributes":<String_value>,

"sendpassword":<String_value>

}]}
```

count

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/authenticationoauthidpprofile?**count=yes**

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "authenticationoauthidpprofile": [ { "__count": "#no"} ] }
```

authenticationpolicy_systemglobal_binding

Binding object showing the systemglobal that can be bound to authenticationpolicy.

Properties

(click to see Operations)

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
boundto	<String>	Read-write	The entity name to which policy is bound.
name	<String>	Read-write	Name of the advance authentication policy. Minimum length = 1
priority	<Double>	Read-only	.
nextfactor	<String>	Read-only	On success invoke label.
gotopriorityexpression	<String>	Read-only	Expression specifying the priority of the next policy which will get evaluated if the current policy rule evaluates to TRUE.
activepolicy	<Double>	Read-only	.
__count	<Double>	Read-only	count parameter

Operations

(click to see Properties)

GET| GET (ALL)| COUNT

Some options that you can use for each operations:

- **Getting warnings in response:**NITRO allows you to get warnings in an operation by specifying the "warning" query parameter as "yes". For example, to get warnings while connecting to the Citrix ADC appliance, the URL is as follows:

  http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/login?warning=yes

  If any, the warnings are displayed in the response payload with the HTTP code "209 X-NITRO-WARNING".

- **Authenticated access for individual NITRO operations:**NITRO allows you to logon to the Citrix ADC appliance to perform individual operations. You can use this option instead of creating a NITRO session (using the login object) and then using that session to perform all operations,

  To do this, you must specify the username and password in the request header of the NITRO request as follows:

  X-NITRO-USER:<username>

  X-NITRO-PASS:<password>

  Note: In such cases, make sure that the request header DOES not include the following:

  Cookie:NITRO_AUTH_TOKEN=<tokenvalue>

`*Note:`*

Mandatory parameters are marked in redand placeholder content is marked in <green>.

get

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/authenticationpolicy_systemglobal_binding/name_value<String>

Query-parameters:

filter

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/authenticationpolicy_systemglobal_binding/name_value<String>?**filter=property-name1:property-value1,property-name2:property-value2**

Use this query-parameter to get the filtered set of authenticationpolicy_systemglobal_binding resources configured on Citrix ADC. Filtering can be done on any of the properties of the resource.

pagination

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/authenticationpolicy_systemglobal_binding/name_value<String>?**pagesize=#no;pageno=#no**

Use this query-parameter to get the authenticationpolicy_systemglobal_binding resources in chunks.

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "authenticationpolicy_systemglobal_binding": [ {

"boundto":<String_value>,

"name":<String_value>,

"priority":<Double_value>,

"nextfactor":<String_value>,

"gotopriorityexpression":<String_value>,
```

```
"activepolicy":<Double_value>

}]}
```

get (all)

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/authenticationpolicy_systemglobal_binding

Query-parameters:

bulkbindings

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/authenticationpolicy_systemglobal_binding?**bulkbindings=yes**

NITRO allows you to fetch bindings in bulk.

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "authenticationpolicy_systemglobal_binding": [ {

"boundto":<String_value>,

"name":<String_value>,

"priority":<Double_value>,
```

```
"nextfactor":<String_value>,

"gotopriorityexpression":<String_value>,

"activepolicy":<Double_value>

}]}
```

count

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/authenticationpolicy_systemglobal_binding/name_value<String>?**count=yes**

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{"authenticationpolicy_systemglobal_binding": [ { "__count": "#no"} ] }
```

authenticationvserver_authenticationoauthidppolicy_binding

Binding object showing the authenticationoauthidppolicy that can be bound to authenticationvserver.

Properties

(click to see Operations)

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
priority	<Double>	Read-write	The priority, if any, of the vpn vserver policy.
name	<String>	Read-write	Name of the authentication virtual server to which to bind the policy. Minimum length = 1
nextfactor	<String>	Read-write	Applicable only while binding advance authentication policy as classic authentication policy does not support nFactor.
gotopriorityexpression	<String>	Read-write	Applicable only to advance authentication policy. Expression or other value specifying the next policy to be evaluated if the current policy evaluates to TRUE. Specify one of the following values: * NEXT - Evaluate the policy with the next higher priority number. * END - End policy evaluation. * USE_INVOCATION_RESULT - Applicable if this policy invokes another policy label. If the final goto in the invoked policy label has a value of END,

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
			the evaluation stops. If the final goto is anything other than END, the current policy label performs a NEXT. * An expression that evaluates to a number. If you specify an expression, the number to which it evaluates determines the next policy to evaluate, as follows: * If the expression evaluates to a higher numbered priority, the policy with that priority is evaluated next. * If the expression evaluates to the priority of the current policy, the policy with the next higher numbered priority is evaluated next. * If the expression evaluates to a priority number that is numerically higher than the highest numbered priority, policy evaluation ends. An UNDEF event is triggered if: * The expression is invalid. * The expression evaluates to a

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
			priority number that is numerically lower than the current policy's priority. * The expression evaluates to a priority number that is between the current policy's priority number (say, 30) and the highest priority number (say, 100), but does not match any configured priority number (for example, the expression evaluates to the number 85). This example assumes that the priority number increments by 10 for every successive policy, and therefore a priority number of 85 does not exist in the policy label.
secondary	<Boolean>	Read-write	Applicable only while bindind classic authentication policy as advance authentication policy use nFactor.
policy	<String>	Read-write	The name of the policy, if any, bound to the authentication vserver.

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
groupextraction	<Boolean>	Read-write	Applicable only while bindind classic authentication policy as advance authentication policy use nFactor.
acttype	<Double>	Read-only	.
__count	<Double>	Read-only	count parameter

Operations

(click to see Properties)

ADD:| DELETE:| GET| GET (ALL)| COUNT

Some options that you can use for each operations:

- **Getting warnings in response:**NITRO allows you to get warnings in an operation by specifying the "warning" query parameter as "yes". For example, to get warnings while connecting to the Citrix ADC appliance, the URL is as follows:

  http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/login?warning=yes

  If any, the warnings are displayed in the response payload with the HTTP code "209 X-NITRO-WARNING".

- **Authenticated access for individual NITRO operations:**NITRO allows you to logon to the Citrix ADC appliance to perform individual operations. You can use this option instead of creating a NITRO session (using the login object) and then using that session to perform all operations,

  To do this, you must specify the username and password in the request header of the NITRO request as follows:

  X-NITRO-USER:<username>

  X-NITRO-PASS:<password>

  Note: In such cases, make sure that the request header DOES not include the following:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue>

`*Note:` *

Mandatory parameters are marked in redand placeholder content is marked in <green>.

add

URL:http://<Citrix-ADC-IP-address(NSIP)/nitro/v1/config/authenticationvserver_authenticationoauthidppolicy_binding

HTTP Method:PUT

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Content-Type:application/json

Request Payload:

```
{
"authenticationvserver_authenticationoauthidppolicy_binding":{
<b>"name":<String_value>,
</b>"policy":<String_value>,
"priority":<Double_value>,
"secondary":<Boolean_value>,
"groupextraction":<Boolean_value>,
"nextfactor":<String_value>,
"gotopriorityexpression":<String_value>
}}
```

Response:

HTTP Status Code on Success: 201 Created HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error

delete

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/authenticationvserver_authenticationoauthidppolicy_binding/name_value<String>

Query-parameters:

args

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/authenticationvserver_authenticationoauthidppolicy_binding/name_value<String>?**args=policy:<String_value>,secondary:<Boolean_value>,groupextraction:<Boolean_value>**

HTTP Method:DELETE

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue>

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error

get

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/authenticationvserver_authenticationoauthidppolicy_binding/name_value<String>

Query-parameters:

filter

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/authenticationvserver_authenticationoauthidppolicy_binding/name_value<String>?**filter=property-name1:property-value1,property-name2:property-value2**

Use this query-parameter to get the filtered set of authenticationvserver_authenticationoauthidppolicy_binding resources configured on Citrix ADC. Filtering can be done on any of the properties of the resource.

pagination

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/authenticationvserver_authenticationoauthidppolicy_binding/name_value<String>?**pagesize=#no;pageno=#no**

Use this query-parameter to get the authenticationvserver_authenticationoauthidppolicy_binding resources in chunks.

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "authenticationvserver_authenticationoauthidppolicy_binding": [ {

"priority":<Double_value>,

"name":<String_value>,

"nextfactor":<String_value>,

"gotopriorityexpression":<String_value>,

"secondary":<Boolean_value>,

"policy":<String_value>,

"groupextraction":<Boolean_value>,

"acttype":<Double_value>
```

```
}]}
```

get (all)

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/authenticationvserver_authenticationoauthidppolicy_binding

Query-parameters:

bulkbindings

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/authenticationvserver_authenticationoauthidppolicy_binding?**bulkbindings=yes**

NITRO allows you to fetch bindings in bulk.

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "authenticationvserver_authenticationoauthidppolicy_binding": [ {

"priority":<Double_value>,

"name":<String_value>,

"nextfactor":<String_value>,

"gotopriorityexpression":<String_value>,
```

```
"secondary":<Boolean_value>,

"policy":<String_value>,

"groupextraction":<Boolean_value>,

"acttype":<Double_value>

}]}
```

count

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/authenticationvserver_authenticationoauthidppolicy_binding/name_value<String>?**count=yes**

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{"authenticationvserver_authenticationoauthidppolicy_binding":
[ { "__count": "#no"} ] }
```

authenticationpolicy

Configuration for Authentication Policy resource.

Properties

(click to see Operations)

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
name	<String>	Read-write	Name for the advance AUTHENTICATION policy. Must begin with a letter, number, or the underscore character (_), and must contain only letters, numbers, and the hyphen (-), period (.) pound (#), space ( ), at (@), equals (=), colon (:), and underscore characters. Cannot be changed after AUTHENTICATION policy is created. The following requirement applies only to the Citrix ADC CLI: If the name includes one or more spaces, enclose the name in double or single quotation marks (for example, "my authentication policy" or 'my

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
			authentication policy'). Minimum length = 1
rule	<String>	Read-write	Name of the Citrix ADC named rule, or an expression, that the policy uses to determine whether to attempt to authenticate the user with the AUTHENTICATION server.
action	<String>	Read-write	Name of the authentication action to be performed if the policy matches.
undefaction	<String>	Read-write	Action to perform if the result of policy evaluation is undefined (UNDEF). An UNDEF event indicates an internal error condition. Only the above built-in actions can be used.
comment	<String>	Read-write	Any comments to preserve information about this policy.
logaction	<String>	Read-write	Name of messagelog action to use when a request matches this policy.

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
newname	<String>	Read-write	New name for the authentication policy. Must begin with a letter, number, or the underscore character (_), and must contain only letters, numbers, and the hyphen (-), period (.) hash (#), space ( ), at (@), equals (=), colon (:), and underscore characters. The following requirement applies only to the Citrix ADC CLI: If the name includes one or more spaces, enclose the name in double or single quotation marks (for example, "my authentication policy" or 'my authentication policy'). Minimum length = 1
hits	<Double>	Read-only	Number of hits.
description	<String>	Read-only	Description of the policy.
policysubtype	<String>	Read-only	.

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
			Possible values = LOCAL, RADIUS, LDAP, TACACS, CERT, NEGOTIATE, SAML, PROFILE, DFA, SAMLIDP, WEBAUTH, OAUTH, NO_AUTHN, EPA, SFAUTH, AZUREVAULT, EMAIL, CAPTCHA
__count	<Double>	Read-only	count parameter

Operations

(click to see Properties)

ADD| DELETE| UPDATE| UNSET| RENAME| GET (ALL)| GET| COUNT

Some options that you can use for each operations:

- **Getting warnings in response:**NITRO allows you to get warnings in an operation by specifying the "warning" query parameter as "yes". For example, to get warnings while connecting to the Citrix ADC appliance, the URL is as follows:

  http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/login?warning=yes

  If any, the warnings are displayed in the response payload with the HTTP code "209 X-NITRO-WARNING".

- **Authenticated access for individual NITRO operations:**NITRO allows you to logon to the Citrix ADC appliance to perform individual operations. You can use this option instead of creating a NITRO session (using the login object) and then using that session to perform all operations,

  To do this, you must specify the username and password in the request header of the NITRO request as follows:

X-NITRO-USER:<username>

X-NITRO-PASS:<password>

Note: In such cases, make sure that the request header DOES not include the following:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue>

`*Note:`*

Mandatory parameters are marked in redand placeholder content is marked in <green>.

add

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/authenticationpolicy

HTTP Method:POST

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Content-Type:application/json

Request Payload:

```
{"authenticationpolicy":{

<b>"name":<String_value>,

</b><b>"rule":<String_value>,

</b><b>"action":<String_value>,

</b>"undefaction":<String_value>,

"comment":<String_value>,

"logaction":<String_value>

}}
```

Response:

HTTP Status Code on Success: 201 Created HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error

delete

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/authenticationpolicy/name_value<String>

HTTP Method:DELETE

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue>

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error

update

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/authenticationpolicy

HTTP Method:PUT

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Content-Type:application/json

Request Payload:

```
{"authenticationpolicy":{

<b>"name":<String_value>,

</b>"rule":<String_value>,

"action":<String_value>,

"undefaction":<String_value>,
```

```
"comment":<String_value>,

"logaction":<String_value>

}}
```

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error

unset

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/authenticationpolicy?**action=unset**

HTTP Method:POST

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Content-Type:application/json

Request Payload:

```
{"authenticationpolicy":{

<b>"name":<String_value>,

</b>"undefaction":true,

"comment":true,

"logaction":true

}}
```

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error

rename

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/authenticationpolicy?**action=rename**

HTTP Method:POST

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Content-Type:application/json

Request Payload:

```
{"authenticationpolicy":{

<b>"name":<String_value>,

</b><b>"newname":<String_value>

</b>}}
```

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error

get (all)

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/authenticationpolicy

Query-parameters:

attrs

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/authenticationpolicy?**attrs=property-name1,property-name2**

Use this query parameter to specify the resource details that you want to retrieve.

filter

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/authenticationpolicy?**filter=property-name1:property-val1,property-name2:property-val2**

Use this query-parameter to get the filtered set of authenticationpolicy resources configured on Citrix ADC. Filtering can be done on any of the properties of the resource.

view

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/authenticationpolicy?**view=summary**

Note: By default, the retrieved results are displayed in detail view (?view=detail).

pagination

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/authenticationpolicy?**pagesize=#no;pageno=#no**

Use this query-parameter to get the authenticationpolicy resources in chunks.

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "authenticationpolicy": [ {

"name":<String_value>,

"rule":<String_value>,
```

```
"action":<String_value>,

"undefaction":<String_value>,

"comment":<String_value>,

"logaction":<String_value>,

"hits":<Double_value>,

"description":<String_value>,

"policysubtype":<String_value>

}]}
```

get

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/authenticationpolicy/name_value<String>

Query-parameters:

attrs

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/authenticationpolicy/name_value<String>?**attrs=property-name1,property-name2**

Use this query parameter to specify the resource details that you want to retrieve.

view

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/authenticationpolicy/name_value<String>?**view=summary**

Note: By default, the retrieved results are displayed in detail view (?view=detail).

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "authenticationpolicy": [ {

"name":<String_value>,

"rule":<String_value>,

"action":<String_value>,

"undefaction":<String_value>,

"comment":<String_value>,

"logaction":<String_value>,

"hits":<Double_value>,

"description":<String_value>,

"policysubtype":<String_value>

}]}
```

count

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/authenticationpolicy?**count=yes**

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "authenticationpolicy": [ { "__count": "#no"} ] }
```

authenticationpolicylabel

Configuration for authentication policy label resource.

Properties

(click to see Operations)

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
labelname	<String>	Read-write	Name for the new authentication policy label. Must begin with a letter, number, or the underscore character (_), and must contain only letters, numbers, and the hyphen (-), period (.) pound (#), space ( ), at (@), equals (=), colon (:), and underscore characters.

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
			The following requirement applies only to the Citrix ADC CLI: If the name includes one or more spaces, enclose the name in double or single quotation marks (for example, "my authentication policy label" or 'authentication policy label').
type	<String>	Read-write	Type of feature (aaatm or rba) against which to match the policies bound to this policy label. Default value: AAATM_REQ Possible values = AAATM_REQ, RBA_REQ
comment	<String>	Read-write	Any comments to preserve information about this authentication policy label.
loginschema	<String>	Read-write	Login schema associated with authentication policy label. Login schema

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
			defines the UI rendering by providing customization option of the fields. If user intervention is not needed for a given factor such as group extraction, a loginSchema whose authentication schema is "noschema" should be used.
newname	<String>	Read-write	The new name of the auth policy label. Minimum length = 1
numpol	<Double>	Read-only	Number of polices bound to label.
hits	<Double>	Read-only	Number of times policy label was invoked.
policyname	<String>	Read-only	Name of the authentication policy to bind to the policy label.
priority	<Double>	Read-only	Specifies the priority of the policy.
gotopriorityexpression	<String>	Read-only	Expression specifying the priority of the next policy which will get evaluated if the current policy rule evaluates to TRUE.

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
flowtype	<Double>	Read-only	Flowtype of the bound authentication policy.
description	<String>	Read-only	Description of the policylabel.
__count	<Double>	Read-only	count parameter

Operations

(click to see Properties)

ADD| DELETE| RENAME| GET (ALL)| GET| COUNT

Some options that you can use for each operations:

- **Getting warnings in response:**NITRO allows you to get warnings in an operation by specifying the "warning" query parameter as "yes". For example, to get warnings while connecting to the Citrix ADC appliance, the URL is as follows:

  http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/login?warning=yes

  If any, the warnings are displayed in the response payload with the HTTP code "209 X-NITRO-WARNING".

- **Authenticated access for individual NITRO operations:**NITRO allows you to logon to the Citrix ADC appliance to perform individual operations. You can use this option instead of creating a NITRO session (using the login object) and then using that session to perform all operations,

  To do this, you must specify the username and password in the request header of the NITRO request as follows:

  X-NITRO-USER:<username>

  X-NITRO-PASS:<password>

  Note: In such cases, make sure that the request header DOES not include the following:

  Cookie:NITRO_AUTH_TOKEN=<tokenvalue>

`*Note:`*

Mandatory parameters are marked in redand placeholder content is marked in <green>.

add

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/authenticationpolicylabel

HTTP Method:POST

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Content-Type:application/json

Request Payload:

```
{"authenticationpolicylabel":{

<b>"labelname":<String_value>,

</b>"type":<String_value>,

"comment":<String_value>,

"loginschema":<String_value>

}}
```

Response:

HTTP Status Code on Success: 201 Created HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error

delete

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/authenticationpolicylabel/labelname_value<String>

HTTP Method:DELETE

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue>

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error

rename

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/authenticationpolicylabel?**action=rename**

HTTP Method:POST

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Content-Type:application/json

Request Payload:

```
{"authenticationpolicylabel":{

<b>"labelname":<String_value>,

</b><b>"newname":<String_value>

</b>}}
```

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error

get (all)

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/authenticationpolicylabel

Query-parameters:

attrs

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/authenticationpolicylabel?**attrs=property-name1,property-name2**

Use this query parameter to specify the resource details that you want to retrieve.

filter

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/authenticationpolicylabel?**filter=property-name1:property-val1,property-name2:property-val2**

Use this query-parameter to get the filtered set of authenticationpolicylabel resources configured on Citrix ADC. Filtering can be done on any of the properties of the resource.

view

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/authenticationpolicylabel?**view=summary**

Note: By default, the retrieved results are displayed in detail view (?view=detail).

pagination

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/authenticationpolicylabel?**pagesize=#no;pageno=#no**

Use this query-parameter to get the authenticationpolicylabel resources in chunks.

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "authenticationpolicylabel": [ {
```

```
"labelname":<String_value>,

"numpol":<Double_value>,

"hits":<Double_value>,

"policyname":<String_value>,

"priority":<Double_value>,

"gotopriorityexpression":<String_value>,

"flowtype":<Double_value>,

"description":<String_value>,

"comment":<String_value>,

"loginschema":<String_value>,

"type":<String_value>

}]}
```

get

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/authenticationpolicylabel/labelname_value<String>

Query-parameters:

attrs

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/authenticationpolicylabel/labelname_value<String>?**attrs=property-name1,property-name2**

Use this query parameter to specify the resource details that you want to retrieve.

view

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/authenticationpolicylabel/labelname_value<String>?**view=summary**

Note: By default, the retrieved results are displayed in detail view (?view=detail).

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "authenticationpolicylabel": [ {

"labelname":<String_value>,

"numpol":<Double_value>,

"hits":<Double_value>,

"policyname":<String_value>,

"priority":<Double_value>,

"gotopriorityexpression":<String_value>,

"flowtype":<Double_value>,

"description":<String_value>,

"comment":<String_value>,

"loginschema":<String_value>,

"type":<String_value>
```

```
}]}
```

count

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/authenticationpolicylabel?**count=yes**

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "authenticationpolicylabel": [ { "__count": "#no"} ] }
```

authenticationpolicylabel_authenticationpolicy_binding

Binding object showing the authenticationpolicy that can be bound to authenticationpolicylabel.

Properties

(click to see Operations)

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
priority	<Double>	Read-write	Specifies the priority of the policy.
nextfactor	<String>	Read-write	On success invoke label.

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
gotopriorityexpression	<String>	Read-write	Expression specifying the priority of the next policy which will get evaluated if the current policy rule evaluates to TRUE.
policyname	<String>	Read-write	Name of the authentication policy to bind to the policy label.
labelname	<String>	Read-write	Name of the authentication policy label to which to bind the policy.
__count	<Double>	Read-write	count parameter

Operations

(click to see Properties)

ADD:| DELETE:| GET| GET (ALL)| COUNT

Some options that you can use for each operations:

- **Getting warnings in response:**NITRO allows you to get warnings in an operation by specifying the "warning" query parameter as "yes". For example, to get warnings while connecting to the Citrix ADC appliance, the URL is as follows:

  http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/login?warning=yes

  If any, the warnings are displayed in the response payload with the HTTP code "209 X-NITRO-WARNING".

- **Authenticated access for individual NITRO operations:**NITRO allows you to logon to the Citrix ADC appliance to perform individual operations. You can use this option instead of creating a NITRO session (using the login object) and then using that session to perform all operations,

To do this, you must specify the username and password in the request header of the NITRO request as follows:

X-NITRO-USER:<username>

X-NITRO-PASS:<password>

Note: In such cases, make sure that the request header DOES not include the following:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue>

`*Note:`*

Mandatory parameters are marked in redand placeholder content is marked in <green>.

add

URL:http://<Citrix-ADC-IP-address(NSIP)/nitro/v1/config/authenticationpolicylabel_authenticationpolicy_binding

HTTP Method:PUT

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Content-Type:application/json

Request Payload:

```
{

"authenticationpolicylabel_authenticationpolicy_binding":{

<b>"labelname":<String_value>,

</b><b>"policyname":<String_value>,

</b><b>"priority":<Double_value>,

</b>"gotopriorityexpression":<String_value>,

"nextfactor":<String_value>

}}
```

Response:

HTTP Status Code on Success: 201 Created HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error

delete

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/authenticationpolicylabel_authenticationpolicy_binding/labelname_value<String>

Query-parameters:

args

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/authenticationpolicylabel_authenticationpolicy_binding/labelname_value<String>?**args=policyname:<String_value>,priority:<Double_value>**

HTTP Method:DELETE

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue>

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error

get

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/authenticationpolicylabel_authenticationpolicy_binding/labelname_value<String>

Query-parameters:

filter

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/authenticationpolicylabel_authenticationpolicy_binding/labelname_value<String>?**filter=property-name1:property-value1,property-name2:property-value2**

Use this query-parameter to get the filtered set of authenticationpolicylabel_authenticationpolicy_binding resources configured on Citrix ADC. Filtering can be done on any of the properties of the resource.

pagination

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/authenticationpolicylabel_authenticationpolicy_binding/labelname_value<String>?**pagesize=#no;pageno=#no**

Use this query-parameter to get the authenticationpolicylabel_authenticationpolicy_binding resources in chunks.

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "authenticationpolicylabel_authenticationpolicy_binding": [ {

"priority":<Double_value>,

"nextfactor":<String_value>,

"gotopriorityexpression":<String_value>,

"policyname":<String_value>,

"labelname":<String_value>

}]}
```

get (all)

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/authenticationpolicylabel_authenticationpolicy_binding

Query-parameters:

bulkbindings

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/authenticationpolicylabel_authenticationpolicy_binding?**bulkbindings=yes**

NITRO allows you to fetch bindings in bulk.

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "authenticationpolicylabel_authenticationpolicy_binding": [ {

"priority":<Double_value>,

"nextfactor":<String_value>,

"gotopriorityexpression":<String_value>,

"policyname":<String_value>,

"labelname":<String_value>
```

```
}]}
```

count

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/authenticationpolicylabel_authenticationpolicy_binding/labelname_value<String>?**count=yes**

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{"authenticationpolicylabel_authenticationpolicy_binding": [ { "__count":
"#no"} ] }
```

authenticationpolicylabel_binding

Binding object which returns the resources bound to authenticationpolicylabel.

Properties

(click to see Operations)

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
labelname	<String>	Read-write	Name of the authorization policy label.

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
authenticationpolicylabel_authenticationpolicy_binding	<authenticationpolicylabel_authenticationpolicy_binding[]>	Read-only	authenticationpolicy that can be bound to authenticationpolicylabel.

Operations

(click to see Properties)

GET| GET (ALL)

Some options that you can use for each operations:

- **Getting warnings in response:**NITRO allows you to get warnings in an operation by specifying the "warning" query parameter as "yes". For example, to get warnings while connecting to the Citrix ADC appliance, the URL is as follows:

  http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/login?warning=yes

  If any, the warnings are displayed in the response payload with the HTTP code "209 X-NITRO-WARNING".

- **Authenticated access for individual NITRO operations:**NITRO allows you to logon to the Citrix ADC appliance to perform individual operations. You can use this option instead of creating a NITRO session (using the login object) and then using that session to perform all operations,

  To do this, you must specify the username and password in the request header of the NITRO request as follows:

  X-NITRO-USER:<username>

  X-NITRO-PASS:<password>

  Note: In such cases, make sure that the request header DOES not include the following:

  Cookie:NITRO_AUTH_TOKEN=<tokenvalue>

`*Note:`*

Mandatory parameters are marked in redand placeholder content is marked in <green>.

get

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/authenticationpolicylabel_binding/labelname_value<String>

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "authenticationpolicylabel_binding": [ {

"labelname":<String_value>,

"authenticationpolicylabel_authenticationpolicy_binding":<authenticationpolicylabel_authenticationpolicy_binding[]_value>

}]}
```

get (all)

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/authenticationpolicylabel_binding

Query-parameters:

bulkbindings

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/authenticationpolicylabel_binding?**bulkbindings=yes**

NITRO allows you to fetch bindings in bulk.

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "authenticationpolicylabel_binding": [ {

"labelname":<String_value>,

"authenticationpolicylabel_authenticationpolicy_binding":<authenticationpo
licylabel_authenticationpolicy_binding[]_value>

}]}
```

authenticationpolicy_authenticationpolicylabel_binding

Binding object showing the authenticationpolicylabel that can be bound to authenticationpolicy.

Properties

(click to see Operations)

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
boundto	<String>	Read-write	The entity name to which policy is bound.

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
name	<String>	Read-write	Name of the advance authentication policy. Minimum length = 1
priority	<Double>	Read-only	.
nextfactor	<String>	Read-only	On success invoke label.
gotopriorityexpression	<String>	Read-only	Expression specifying the priority of the next policy which will get evaluated if the current policy rule evaluates to TRUE.
activepolicy	<Double>	Read-only	.
__count	<Double>	Read-only	count parameter

Operations

(click to see Properties)

GET| GET (ALL)| COUNT

Some options that you can use for each operations:

- **Getting warnings in response:**NITRO allows you to get warnings in an operation by specifying the "warning" query parameter as "yes". For example, to get warnings while connecting to the Citrix ADC appliance, the URL is as follows:

  http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/login?warning=yes

  If any, the warnings are displayed in the response payload with the HTTP code "209 X-NITRO-WARNING".
- **Authenticated access for individual NITRO operations:**NITRO allows you to logon to the Citrix ADC appliance to perform individual operations. You can use this option instead of

creating a NITRO session (using the login object) and then using that session to perform all operations,

To do this, you must specify the username and password in the request header of the NITRO request as follows:

X-NITRO-USER:<username>

X-NITRO-PASS:<password>

Note: In such cases, make sure that the request header DOES not include the following:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue>

`*Note:`*

Mandatory parameters are marked in redand placeholder content is marked in <green>.

get

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/authenticationpolicy_authenticationpolicylabel_binding/name_value<String>

Query-parameters:

filter

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/authenticationpolicy_authenticationpolicylabel_binding/name_value<String>?**filter=property-name1:property-value1,property-name2:property-value2**

Use this query-parameter to get the filtered set of authenticationpolicy_authenticationpolicylabel_binding resources configured on Citrix ADC. Filtering can be done on any of the properties of the resource.

pagination

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/authenticationpolicy_authenticationpolicylabel_binding/name_value<String>?**pagesize=#no;pageno=#no**

Use this query-parameter to get the authenticationpolicy_authenticationpolicylabel_binding resources in chunks.

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "authenticationpolicy_authenticationpolicylabel_binding": [ {

"boundto":<String_value>,

"name":<String_value>,

"priority":<Double_value>,

"nextfactor":<String_value>,

"gotopriorityexpression":<String_value>,

"activepolicy":<Double_value>

}]}
```

get (all)

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/authenticationpolicy_authenticationpolicylabel_binding

Query-parameters:

bulkbindings

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/authenticationpolicy_authenticationpolicylabel_binding?**bulkbindings=yes**

NITRO allows you to fetch bindings in bulk.

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "authenticationpolicy_authenticationpolicylabel_binding": [ {

"boundto":<String_value>,

"name":<String_value>,

"priority":<Double_value>,

"nextfactor":<String_value>,

"gotopriorityexpression":<String_value>,

"activepolicy":<Double_value>

}]}
```

count

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/authenticationpolicy_authenticationpolicylabel_binding/name_value<String>?**count=yes**

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{"authenticationpolicy_authenticationpolicylabel_binding": [ { "__count":
"#no"} ] }
```

authenticationpolicy_authenticationvserver_binding

Binding object showing the authenticationvserver that can be bound to authenticationpolicy.

Properties

(click to see Operations)

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
boundto	<String>	Read-write	The entity name to which policy is bound.
name	<String>	Read-write	Name of the advance authentication policy. Minimum length = 1
priority	<Double>	Read-only	.
nextfactor	<String>	Read-only	On success invoke label.

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
gotopriorityexpression	<String>	Read-only	Expression specifying the priority of the next policy which will get evaluated if the current policy rule evaluates to TRUE.
activepolicy	<Double>	Read-only	.
__count	<Double>	Read-only	count parameter

Operations

(click to see Properties)

GET| GET (ALL)| COUNT

Some options that you can use for each operations:

- **Getting warnings in response:**NITRO allows you to get warnings in an operation by specifying the "warning" query parameter as "yes". For example, to get warnings while connecting to the Citrix ADC appliance, the URL is as follows:

  http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/login?warning=yes

  If any, the warnings are displayed in the response payload with the HTTP code "209 X-NITRO-WARNING".

- **Authenticated access for individual NITRO operations:**NITRO allows you to logon to the Citrix ADC appliance to perform individual operations. You can use this option instead of creating a NITRO session (using the login object) and then using that session to perform all operations,

  To do this, you must specify the username and password in the request header of the NITRO request as follows:

  X-NITRO-USER:<username>

  X-NITRO-PASS:<password>

  Note: In such cases, make sure that the request header DOES not include the following:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue>

`*Note:`*

Mandatory parameters are marked in redand placeholder content is marked in <green>.

get

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/authenticationpolicy_authenticationvserver_binding/name_value<String>

Query-parameters:

filter

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/authenticationpolicy_authenticationvserver_binding/name_value<String>?**filter=property-name1:property-value1,property-name2:property-value2**

Use this query-parameter to get the filtered set of authenticationpolicy_authenticationvserver_binding resources configured on Citrix ADC. Filtering can be done on any of the properties of the resource.

pagination

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/authenticationpolicy_authenticationvserver_binding/name_value<String>?**pagesize=#no;pageno=#no**

Use this query-parameter to get the authenticationpolicy_authenticationvserver_binding resources in chunks.

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "authenticationpolicy_authenticationvserver_binding": [ {

"boundto":<String_value>,

"name":<String_value>,

"priority":<Double_value>,

"nextfactor":<String_value>,

"gotopriorityexpression":<String_value>,

"activepolicy":<Double_value>

}]}
```

get (all)

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/authenticationpolicy_authenticationvserver_binding

Query-parameters:

bulkbindings

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/authenticationpolicy_authenticationvserver_binding?**bulkbindings=yes**

NITRO allows you to fetch bindings in bulk.

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "authenticationpolicy_authenticationvserver_binding": [ {

"boundto":<String_value>,

"name":<String_value>,

"priority":<Double_value>,

"nextfactor":<String_value>,

"gotopriorityexpression":<String_value>,

"activepolicy":<Double_value>

}]}
```

count

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/authenticationpolicy_authenticationvserver_binding/name_value<String>?**count=yes**

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{"authenticationpolicy_authenticationvserver_binding": [ { "__count":
"#no"} ] }
```

authenticationpolicy_binding

Binding object which returns the resources bound to authenticationpolicy.

Properties

(click to see Operations)

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
name	<String>	Read-write	Name of the advance authentication policy. Minimum length = 1
authenticationpolicy_authenticationpolicylabel_binding	<authenticationpolicy_authenticationpolicylabel_binding[]>	Read-only	authenticationpolicylabel that can be bound to authenticationpolicy.
authenticationpolicy_systemglobal_binding	<authenticationpolicy_systemglobal_binding[]>	Read-only	systemglobal that can be bound to authenticationpolicy.
authenticationpolicy_authenticationvserver_binding	<authenticationpolicy_authenticationvserver_binding[]>	Read-only	authenticationvserver that can be bound to authenticationpolicy.

Operations

(click to see Properties)

GET| GET (ALL)

Some options that you can use for each operations:

- **Getting warnings in response:**NITRO allows you to get warnings in an operation by specifying the "warning" query parameter as "yes". For example, to get warnings while connecting to the Citrix ADC appliance, the URL is as follows:

  http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/login?warning=yes

  If any, the warnings are displayed in the response payload with the HTTP code "209 X-NITRO-WARNING".

- **Authenticated access for individual NITRO operations:**NITRO allows you to logon to the Citrix ADC appliance to perform individual operations. You can use this option instead of creating a NITRO session (using the login object) and then using that session to perform all operations,

  To do this, you must specify the username and password in the request header of the NITRO request as follows:

  X-NITRO-USER:<username>

  X-NITRO-PASS:<password>

  Note: In such cases, make sure that the request header DOES not include the following:

  Cookie:NITRO_AUTH_TOKEN=<tokenvalue>

`*Note:`*

Mandatory parameters are marked in redand placeholder content is marked in <green>.

get

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/authenticationpolicy_binding/name_value<String>

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "authenticationpolicy_binding": [ {

"name":<String_value>,

"authenticationpolicy_systemglobal_binding":<authenticationpolicy_systemglobal_binding[]_value>,

"authenticationpolicy_authenticationvserver_binding":<authenticationpolicy_authenticationvserver_binding[]_value>,

"authenticationpolicy_authenticationpolicylabel_binding":<authenticationpolicy_authenticationpolicylabel_binding[]_value>

}]}
```

get (all)

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/authenticationpolicy_binding

Query-parameters:

bulkbindings

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/authenticationpolicy_binding?**bulkbindings=yes**

NITRO allows you to fetch bindings in bulk.

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "authenticationpolicy_binding": [ {

"name":<String_value>,

"authenticationpolicy_systemglobal_binding":<authenticationpolicy_systemglobal_binding[]_value>,

"authenticationpolicy_authenticationvserver_binding":<authenticationpolicy_authenticationvserver_binding[]_value>,

"authenticationpolicy_authenticationpolicylabel_binding":<authenticationpolicy_authenticationpolicylabel_binding[]_value>

}]}
```

authenticationradiusaction

Configuration for RADIUS action resource.

Properties

(click to see Operations)

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
name	<String>	Read-write	Name for the RADIUS action. Must begin with a letter, number, or the underscore

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
			character (_), and must contain only letters, numbers, and the hyphen (-), period (.) pound (#), space ( ), at (@), equals (=), colon (:), and underscore characters. Cannot be changed after the RADIUS action is added. Minimum length = 1
serverip	<String>	Read-write	IP address assigned to the RADIUS server. Minimum length = 1
servername	<String>	Read-write	RADIUS server name as a FQDN. Mutually exclusive with RADIUS IP address. Minimum length = 1
serverport	<Integer>	Read-write	Port number on which the RADIUS server listens for connections. Minimum value = 1
authtimeout	<Double>	Read-write	Number of seconds the Citrix ADC waits for a response from the RADIUS server. Default value: 3

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
			Minimum value = 1
radkey	<String>	Read-write	Key shared between the RADIUS server and the Citrix ADC. Required to allow the Citrix ADC to communicate with the RADIUS server. Minimum length = 1
radnasip	<String>	Read-write	If enabled, the Citrix ADC IP address (NSIP) is sent to the RADIUS server as the Network Access Server IP (NASIP) address. The RADIUS protocol defines the meaning and use of the NASIP address. Possible values = ENABLED, DISABLED
radnasid	<String>	Read-write	If configured, this string is sent to the RADIUS server as the Network Access Server ID (NASID).
radvendorid	<Double>	Read-write	RADIUS vendor ID attribute, used for RADIUS group extraction. Minimum value = 1

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
radattributetype	<Double>	Read-write	RADIUS attribute type, used for RADIUS group extraction. Minimum value = 1
radgroupsprefix	<String>	Read-write	RADIUS groups prefix string. This groups prefix precedes the group names within a RADIUS attribute for RADIUS group extraction.
radgroupseparator	<String>	Read-write	RADIUS group separator string The group separator delimits group names within a RADIUS attribute for RADIUS group extraction.
passencoding	<String>	Read-write	Encoding type for passwords in RADIUS packets that the Citrix ADC sends to the RADIUS server. Default value: pap Possible values = pap, chap, mschapv1, mschapv2

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
ipvendorid	<Double>	Read-write	Vendor ID of the intranet IP attribute in the RADIUS response. NOTE: A value of 0 indicates that the attribute is not vendor encoded.
ipattributetype	<Double>	Read-write	Remote IP address attribute type in a RADIUS response. Minimum value = 1
accounting	<String>	Read-write	Whether the RADIUS server is currently accepting accounting messages. Possible values = ON, OFF
pwdvendorid	<Double>	Read-write	Vendor ID of the attribute, in the RADIUS response, used to extract the user password. Minimum value = 1
pwdattributetype	<Double>	Read-write	Vendor-specific password attribute type in a RADIUS response. Minimum value = 1
defaultauthentication group	<String>	Read-write	This is the default group that is chosen when the authentication

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
			succeeds in addition to extracted groups.
callingstationid	<String>	Read-write	Send Calling-Station-ID of the client to the RADIUS server. IP Address of the client is sent as its Calling-Station-ID. Default value: DISABLED Possible values = ENABLED, DISABLED
authservretry	<Double>	Read-write	Number of retry by the Citrix ADC before getting response from the RADIUS server. Default value: 3 Minimum value = 1 Maximum value = 10
authentication	<String>	Read-write	Configure the RADIUS server state to accept or refuse authentication messages. Default value: ON Possible values = ON, OFF
tunnelendpointclientip	<String>	Read-write	Send Tunnel Endpoint Client IP

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
			address to the RADIUS server. Default value: DISABLED Possible values = ENABLED, DISABLED
ipaddress	<String>	Read-only	IP address.
success	<Double>	Read-only	.
failure	<Double>	Read-only	.
__count	<Double>	Read-only	count parameter

Operations

(click to see Properties)

ADD| DELETE| UPDATE| UNSET| GET (ALL)| GET| COUNT

Some options that you can use for each operations:

- **Getting warnings in response:**NITRO allows you to get warnings in an operation by specifying the "warning" query parameter as "yes". For example, to get warnings while connecting to the Citrix ADC appliance, the URL is as follows:

  http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/login?warning=yes

  If any, the warnings are displayed in the response payload with the HTTP code "209 X-NITRO-WARNING".

- **Authenticated access for individual NITRO operations:**NITRO allows you to logon to the Citrix ADC appliance to perform individual operations. You can use this option instead of creating a NITRO session (using the login object) and then using that session to perform all operations,

  To do this, you must specify the username and password in the request header of the NITRO request as follows:

X-NITRO-USER:<username>

X-NITRO-PASS:<password>

Note: In such cases, make sure that the request header DOES not include the following:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue>

`*Note:`*

Mandatory parameters are marked in redand placeholder content is marked in <green>.

add

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/authenticationradiusaction

HTTP Method:POST

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Content-Type:application/json

Request Payload:

```
{"authenticationradiusaction":{

<b>"name":<String_value>,

</b>"serverip":<String_value>,

"servername":<String_value>,

"serverport":<Integer_value>,

"authtimeout":<Double_value>,

<b>"radkey":<String_value>,

</b>"radnasip":<String_value>,

"radnasid":<String_value>,

"radvendorid":<Double_value>,
```

```
"radattributetype":<Double_value>,

"radgroupsprefix":<String_value>,

"radgroupseparator":<String_value>,

"passencoding":<String_value>,

"ipvendorid":<Double_value>,

"ipattributetype":<Double_value>,

"accounting":<String_value>,

"pwdvendorid":<Double_value>,

"pwdattributetype":<Double_value>,

"defaultauthenticationgroup":<String_value>,

"callingstationid":<String_value>,

"authservretry":<Double_value>,

"authentication":<String_value>,

"tunnelendpointclientip":<String_value>

}}
```

Response:

HTTP Status Code on Success: 201 Created HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error

delete

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/authenticationradiusaction/name_value<String>

HTTP Method:DELETE

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue>

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error

update

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/authenticationradiusaction

HTTP Method:PUT

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Content-Type:application/json

Request Payload:

```
{"authenticationradiusaction":{

<b>"name":<String_value>,

</b>"serverip":<String_value>,

"servername":<String_value>,

"serverport":<Integer_value>,

"authtimeout":<Double_value>,

"radkey":<String_value>,

"radnasip":<String_value>,

"radnasid":<String_value>,

"radvendorid":<Double_value>,

"radattributetype":<Double_value>,
```

```
"radgroupsprefix":<String_value>,

"radgroupseparator":<String_value>,

"passencoding":<String_value>,

"ipvendorid":<Double_value>,

"ipattributetype":<Double_value>,

"accounting":<String_value>,

"pwdvendorid":<Double_value>,

"pwdattributetype":<Double_value>,

"defaultauthenticationgroup":<String_value>,

"callingstationid":<String_value>,

"authservretry":<Double_value>,

"authentication":<String_value>,

"tunnelendpointclientip":<String_value>

}}
```

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error

unset

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/authenticationradiusaction?**action=unset**

HTTP Method:POST

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Content-Type:application/json

Request Payload:

```
{"authenticationradiusaction":{

<b>"name":<String_value>,

</b>"serverport":true,

"authtimeout":true,

"radnasip":true,

"radnasid":true,

"radvendorid":true,

"radattributetype":true,

"radgroupsprefix":true,

"radgroupseparator":true,

"passencoding":true,

"ipvendorid":true,

"ipattributetype":true,

"accounting":true,

"pwdvendorid":true,

"pwdattributetype":true,

"defaultauthenticationgroup":true,

"callingstationid":true,

"authservretry":true,
```

```
"authentication":true,

"tunnelendpointclientip":true

}}
```

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error

get (all)

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/authenticationradiusaction

Query-parameters:

attrs

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/authenticationradiusaction?**attrs=property-name1,property-name2**

Use this query parameter to specify the resource details that you want to retrieve.

filter

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/authenticationradiusaction?**filter=property-name1:property-val1,property-name2:property-val2**

Use this query-parameter to get the filtered set of authenticationradiusaction resources configured on Citrix ADC. Filtering can be done on any of the properties of the resource.

view

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/authenticationradiusaction?**view=summary**

Note: By default, the retrieved results are displayed in detail view (?view=detail).

pagination

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/authenticationradiusaction?**pagesize=#no;pageno=#no**

Use this query-parameter to get the authenticationradiusaction resources in chunks.

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "authenticationradiusaction": [ {

"name":<String_value>,

"serverip":<String_value>,

"servername":<String_value>,

"serverport":<Integer_value>,

"authtimeout":<Double_value>,

"radkey":<String_value>,

"radnasip":<String_value>,

"ipaddress":<String_value>,

"radnasid":<String_value>,

"radvendorid":<Double_value>,

"radattributetype":<Double_value>,
```

```
"radgroupsprefix":<String_value>,

"radgroupseparator":<String_value>,

"passencoding":<String_value>,

"ipvendorid":<Double_value>,

"ipattributetype":<Double_value>,

"accounting":<String_value>,

"success":<Double_value>,

"failure":<Double_value>,

"pwdvendorid":<Double_value>,

"pwdattributetype":<Double_value>,

"defaultauthenticationgroup":<String_value>,

"callingstationid":<String_value>,

"authservretry":<Double_value>,

"authentication":<String_value>,

"tunnelendpointclientip":<String_value>

}]}
```

get

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/authenticationradiusaction/name_value<String>

Query-parameters:

attrs

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/authenticationradiusaction/name_value<String>?**attrs=property-name1,property-name2**

Use this query parameter to specify the resource details that you want to retrieve.

view

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/authenticationradiusaction/name_value<String>?**view=summary**

Note: By default, the retrieved results are displayed in detail view (?view=detail).

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "authenticationradiusaction": [ {

"name":<String_value>,

"serverip":<String_value>,

"servername":<String_value>,

"serverport":<Integer_value>,

"authtimeout":<Double_value>,

"radkey":<String_value>,
```

```
"radnasip":<String_value>,

"ipaddress":<String_value>,

"radnasid":<String_value>,

"radvendorid":<Double_value>,

"radattributetype":<Double_value>,

"radgroupsprefix":<String_value>,

"radgroupseparator":<String_value>,

"passencoding":<String_value>,

"ipvendorid":<Double_value>,

"ipattributetype":<Double_value>,

"accounting":<String_value>,

"success":<Double_value>,

"failure":<Double_value>,

"pwdvendorid":<Double_value>,

"pwdattributetype":<Double_value>,

"defaultauthenticationgroup":<String_value>,

"callingstationid":<String_value>,

"authservretry":<Double_value>,

"authentication":<String_value>,

"tunnelendpointclientip":<String_value>

}]}
```

count

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/authenticationradiusaction?**count=yes**

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "authenticationradiusaction": [ { "__count": "#no"} ] }
```

authenticationradiuspolicy

Configuration for RADIUS policy resource.

Properties

(click to see Operations)

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
name	<String>	Read-write	Name for the RADIUS authentication policy. Must begin with a letter, number, or the underscore character (_), and

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
			must contain only letters, numbers, and the hyphen (-), period (.) pound (#), space ( ), at (@), equals (=), colon (:), and underscore characters. Cannot be changed after RADIUS policy is created. The following requirement applies only to the Citrix ADC CLI: If the name includes one or more spaces, enclose the name in double or single quotation marks (for example, "my authentication policy" or 'my authentication policy'). Minimum length = 1
rule	<String>	Read-write	Name of the Citrix ADC named rule, or an expression, that the policy uses to determine whether to attempt to authenticate the user

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
			with the RADIUS server.
reqaction	<String>	Read-write	Name of the RADIUS action to perform if the policy matches. Minimum length = 1
__count	<Double>	Read-only	count parameter

Operations

(click to see Properties)

ADD| DELETE| UPDATE| UNSET| GET (ALL)| GET| COUNT

Some options that you can use for each operations:

- **Getting warnings in response:**NITRO allows you to get warnings in an operation by specifying the "warning" query parameter as "yes". For example, to get warnings while connecting to the Citrix ADC appliance, the URL is as follows:

  http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/login?warning=yes

  If any, the warnings are displayed in the response payload with the HTTP code "209 X-NITRO-WARNING".

- **Authenticated access for individual NITRO operations:**NITRO allows you to logon to the Citrix ADC appliance to perform individual operations. You can use this option instead of creating a NITRO session (using the login object) and then using that session to perform all operations,

  To do this, you must specify the username and password in the request header of the NITRO request as follows:

  X-NITRO-USER:<username>

  X-NITRO-PASS:<password>

  Note: In such cases, make sure that the request header DOES not include the following:

  Cookie:NITRO_AUTH_TOKEN=<tokenvalue>

`*Note:` *

Mandatory parameters are marked in redand placeholder content is marked in <green>.

add

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/authenticationradiuspolicy

HTTP Method:POST

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Content-Type:application/json

Request Payload:

```
{"authenticationradiuspolicy":{

<b>"name":<String_value>,

</b><b>"rule":<String_value>,

</b>"reqaction":<String_value>

}}
```

Response:

HTTP Status Code on Success: 201 Created HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error

delete

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/authenticationradiuspolicy/name_value<String>

HTTP Method:DELETE

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue>

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error

update

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/authenticationradiuspolicy

HTTP Method:PUT

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Content-Type:application/json

Request Payload:

```
{"authenticationradiuspolicy":{

<b>"name":<String_value>,

</b>"rule":<String_value>,

"reqaction":<String_value>

}}
```

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error

unset

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/authenticationradiuspolicy?**action=unset**

HTTP Method:POST

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Content-Type:application/json

Request Payload:

```
{"authenticationradiuspolicy":{

<b>"name":<String_value>,

</b>"rule":true,

"reqaction":true

}}
```

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error

get (all)

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/authenticationradiuspolicy

Query-parameters:

attrs

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/authenticationradiuspolicy?**attrs=property-name1,property-name2**

Use this query parameter to specify the resource details that you want to retrieve.

filter

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/authenticationradiuspolicy?**filter=property-name1:property-val1,property-name2:property-val2**

Use this query-parameter to get the filtered set of authenticationradiuspolicy resources configured on Citrix ADC. Filtering can be done on any of the properties of the resource.

view

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/authenticationradiuspolicy?**view=summary**

Note: By default, the retrieved results are displayed in detail view (?view=detail).

pagination

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/authenticationradiuspolicy?**pagesize=#no;pageno=#no**

Use this query-parameter to get the authenticationradiuspolicy resources in chunks.

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "authenticationradiuspolicy": [ {

"name":<String_value>,

"rule":<String_value>,

"reqaction":<String_value>

}]}
```

get

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/authenticationradiuspolicy/name_value<String>

Query-parameters:

attrs

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/authenticationradiuspolicy/name_value<String>?**attrs=property-name1,property-name2**

Use this query parameter to specify the resource details that you want to retrieve.

view

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/authenticationradiuspolicy/name_value<String>?**view=summary**

Note: By default, the retrieved results are displayed in detail view (?view=detail).

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "authenticationradiuspolicy": [ {

"name":<String_value>,

"rule":<String_value>,
```

```
"reqaction":<String_value>

}]}
```

count

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/authenticationradiuspolicy?**count=yes**

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "authenticationradiuspolicy": [ { "__count": "#no"} ] }
```

authenticationradiuspolicy_authenticationvserver_binding

Binding object showing the authenticationvserver that can be bound to authenticationradiuspolicy.

Properties

(click to see Operations)

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
boundto	<String>	Read-write	The entity name to which policy is bound.

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
name	<String>	Read-write	Name of the RADIUS authentication policy. Minimum length = 1
priority	<Double>	Read-only	.
activepolicy	<Double>	Read-only	.
__count	<Double>	Read-only	count parameter

Operations

(click to see Properties)

GET| GET (ALL)| COUNT

Some options that you can use for each operations:

- **Getting warnings in response:**NITRO allows you to get warnings in an operation by specifying the "warning" query parameter as "yes". For example, to get warnings while connecting to the Citrix ADC appliance, the URL is as follows:

  http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/login?warning=yes

  If any, the warnings are displayed in the response payload with the HTTP code "209 X-NITRO-WARNING".

- **Authenticated access for individual NITRO operations:**NITRO allows you to logon to the Citrix ADC appliance to perform individual operations. You can use this option instead of creating a NITRO session (using the login object) and then using that session to perform all operations,

  To do this, you must specify the username and password in the request header of the NITRO request as follows:

  X-NITRO-USER:<username>

  X-NITRO-PASS:<password>

  Note: In such cases, make sure that the request header DOES not include the following:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue>

`*Note:`*

Mandatory parameters are marked in redand placeholder content is marked in <green>.

get

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/authenticationradiuspolicy_authenticationvserver_binding/name_value<String>

Query-parameters:

filter

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/authenticationradiuspolicy_authenticationvserver_binding/name_value<String>?**filter=property-name1:property-value1,property-name2:property-value2**

Use this query-parameter to get the filtered set of authenticationradiuspolicy_authenticationvserver_binding resources configured on Citrix ADC. Filtering can be done on any of the properties of the resource.

pagination

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/authenticationradiuspolicy_authenticationvserver_binding/name_value<String>?**pagesize=#no;pageno=#no**

Use this query-parameter to get the authenticationradiuspolicy_authenticationvserver_binding resources in chunks.

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "authenticationradiuspolicy_authenticationvserver_binding": [ {

"boundto":<String_value>,

"name":<String_value>,

"priority":<Double_value>,

"activepolicy":<Double_value>

}]}
```

get (all)

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/authenticationradiuspolicy_authenticationvserver_binding

Query-parameters:

bulkbindings

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/authenticationradiuspolicy_authenticationvserver_binding?**bulkbindings=yes**

NITRO allows you to fetch bindings in bulk.

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "authenticationradiuspolicy_authenticationvserver_binding": [ {

"boundto":<String_value>,

"name":<String_value>,

"priority":<Double_value>,

"activepolicy":<Double_value>

}]}
```

count

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/authenticationradiuspolicy_authenticationvserver_binding/name_value<String>?**count=yes**

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{"authenticationradiuspolicy_authenticationvserver_binding":
[ { "__count": "#no"} ] }
```

authenticationradiuspolicy_binding

Binding object which returns the resources bound to authenticationradiuspolicy.

Properties

(click to see Operations)

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
name	<String>	Read-write	Name of the RADIUS authentication policy. Minimum length = 1
authenticationradiuspolicy_vpnvserver_binding	<authenticationradiuspolicy_vpnvserver_binding[]>	Read-only	vpnvserver that can be bound to authenticationradiuspolicy.
authenticationradiuspolicy_systemglobal_binding	<authenticationradiuspolicy_systemglobal_binding[]>	Read-only	systemglobal that can be bound to authenticationradiuspolicy.
authenticationradiuspolicy_authenticationvserver_binding	<authenticationradiuspolicy_authenticationvserver_binding[]>	Read-only	authenticationvserver that can be bound to authenticationradiuspolicy.
authenticationradiuspolicy_vpnglobal_binding	<authenticationradiuspolicy_vpnglobal_binding[]>	Read-only	vpnglobal that can be bound to authenticationradiuspolicy.

Operations

(click to see Properties)

GET| GET (ALL)

Some options that you can use for each operations:

- **Getting warnings in response:**NITRO allows you to get warnings in an operation by specifying the "warning" query parameter as "yes". For example, to get warnings while connecting to the Citrix ADC appliance, the URL is as follows:

  http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/login?warning=yes

  If any, the warnings are displayed in the response payload with the HTTP code "209 X-NITRO-WARNING".

- **Authenticated access for individual NITRO operations:**NITRO allows you to logon to the Citrix ADC appliance to perform individual operations. You can use this option instead of creating a NITRO session (using the login object) and then using that session to perform all operations,

  To do this, you must specify the username and password in the request header of the NITRO request as follows:

  X-NITRO-USER:<username>

  X-NITRO-PASS:<password>

  Note: In such cases, make sure that the request header DOES not include the following:

  Cookie:NITRO_AUTH_TOKEN=<tokenvalue>

`*Note:`*

Mandatory parameters are marked in redand placeholder content is marked in <green>.

get

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/authenticationradiuspolicy_binding/name_value<String>

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "authenticationradiuspolicy_binding": [ {

"name":<String_value>,

"authenticationradiuspolicy_systemglobal_binding":<authenticationradiuspol
icy_systemglobal_binding[]_value>,

"authenticationradiuspolicy_authenticationvserver_binding":<authentication
radiuspolicy_authenticationvserver_binding[]_value>,

"authenticationradiuspolicy_vpnglobal_binding":<authenticationradiuspolicy
_vpnglobal_binding[]_value>,

"authenticationradiuspolicy_vpnvserver_binding":<authenticationradiuspolic
y_vpnvserver_binding[]_value>

}]}
```

get (all)

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/authenticationradiuspolicy_binding

Query-parameters:

bulkbindings

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/authenticationradiuspolicy_binding?**bulkbindings=yes**

NITRO allows you to fetch bindings in bulk.

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "authenticationradiuspolicy_binding": [ {

"name":<String_value>,

"authenticationradiuspolicy_systemglobal_binding":<authenticationradiuspol
icy_systemglobal_binding[]_value>,

"authenticationradiuspolicy_authenticationvserver_binding":<authentication
radiuspolicy_authenticationvserver_binding[]_value>,

"authenticationradiuspolicy_vpnglobal_binding":<authenticationradiuspolicy
_vpnglobal_binding[]_value>,

"authenticationradiuspolicy_vpnvserver_binding":<authenticationradiuspolic
y_vpnvserver_binding[]_value>

}]}
```

authenticationradiuspolicy_systemglobal_binding

Binding object showing the systemglobal that can be bound to authenticationradiuspolicy.

Properties

(click to see Operations)

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
boundto	<String>	Read-write	The entity name to which policy is bound.
name	<String>	Read-write	Name of the RADIUS authentication policy. Minimum length = 1
priority	<Double>	Read-only	.
activepolicy	<Double>	Read-only	.
__count	<Double>	Read-only	count parameter

Operations

(click to see Properties)

GET| GET (ALL)| COUNT

Some options that you can use for each operations:

- **Getting warnings in response:**NITRO allows you to get warnings in an operation by specifying the "warning" query parameter as "yes". For example, to get warnings while connecting to the Citrix ADC appliance, the URL is as follows:

  http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/login?warning=yes

  If any, the warnings are displayed in the response payload with the HTTP code "209 X-NITRO-WARNING".

- **Authenticated access for individual NITRO operations:**NITRO allows you to logon to the Citrix ADC appliance to perform individual operations. You can use this option instead of creating a NITRO session (using the login object) and then using that session to perform all operations,

  To do this, you must specify the username and password in the request header of the NITRO request as follows:

  X-NITRO-USER:<username>

X-NITRO-PASS:<password>

Note: In such cases, make sure that the request header DOES not include the following:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue>

`*Note:`*

Mandatory parameters are marked in redand placeholder content is marked in <green>.

get

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/authenticationradiuspolicy_systemglobal_binding/name_value<String>

Query-parameters:

filter

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/authenticationradiuspolicy_systemglobal_binding/name_value<String>?**filter=property-name1:property-value1,property-name2:property-value2**

Use this query-parameter to get the filtered set of authenticationradiuspolicy_systemglobal_binding resources configured on Citrix ADC. Filtering can be done on any of the properties of the resource.

pagination

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/authenticationradiuspolicy_systemglobal_binding/name_value<String>?**pagesize=#no;pageno=#no**

Use this query-parameter to get the authenticationradiuspolicy_systemglobal_binding resources in chunks.

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "authenticationradiuspolicy_systemglobal_binding": [ {

"boundto":<String_value>,

"name":<String_value>,

"priority":<Double_value>,

"activepolicy":<Double_value>

}]}
```

get (all)

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/authenticationradiuspolicy_systemglobal_binding

Query-parameters:

bulkbindings

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/authenticationradiuspolicy_systemglobal_binding?**bulkbindings=yes**

NITRO allows you to fetch bindings in bulk.

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "authenticationradiuspolicy_systemglobal_binding": [ {

"boundto":<String_value>,

"name":<String_value>,

"priority":<Double_value>,

"activepolicy":<Double_value>

}]}
```

count

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/authenticationradiuspolicy_systemglobal_binding/name_value<String>?**count=yes**

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{"authenticationradiuspolicy_systemglobal_binding": [ { "__count":
"#no"} ] }
```

authenticationradiuspolicy_vpnglobal_binding

Binding object showing the vpnglobal that can be bound to authenticationradiuspolicy.

Properties

(click to see Operations)

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
boundto	<String>	Read-write	The entity name to which policy is bound.
name	<String>	Read-write	Name of the RADIUS authentication policy. Minimum length = 1
priority	<Double>	Read-only	.
activepolicy	<Double>	Read-only	.
__count	<Double>	Read-only	count parameter

Operations

(click to see Properties)

GET| GET (ALL)| COUNT

Some options that you can use for each operations:

- **Getting warnings in response:**NITRO allows you to get warnings in an operation by specifying the "warning" query parameter as "yes". For example, to get warnings while connecting to the Citrix ADC appliance, the URL is as follows:

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/login?warning=yes

If any, the warnings are displayed in the response payload with the HTTP code "209 X-NITRO-WARNING".

- **Authenticated access for individual NITRO operations:**NITRO allows you to logon to the Citrix ADC appliance to perform individual operations. You can use this option instead of creating a NITRO session (using the login object) and then using that session to perform all operations,

  To do this, you must specify the username and password in the request header of the NITRO request as follows:

  X-NITRO-USER:<username>

  X-NITRO-PASS:<password>

  Note: In such cases, make sure that the request header DOES not include the following:

  Cookie:NITRO_AUTH_TOKEN=<tokenvalue>

`*Note:`*

Mandatory parameters are marked in redand placeholder content is marked in <green>.

get

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/authenticationradiuspolicy_vpnglobal_binding/name_value<String>

Query-parameters:

filter

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/authenticationradiuspolicy_vpnglobal_binding/name_value<String>?**filter=property-name1:property-value1,property-name2:property-value2**

Use this query-parameter to get the filtered set of authenticationradiuspolicy_vpnglobal_binding resources configured on Citrix ADC. Filtering can be done on any of the properties of the resource.

pagination

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/authenticationradiuspolicy_vpnglobal_binding/name_value<String>?**pagesize=#no;pageno=#no**

Use this query-parameter to get the authenticationradiuspolicy_vpnglobal_binding resources in chunks.

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "authenticationradiuspolicy_vpnglobal_binding": [ {

"boundto":<String_value>,

"name":<String_value>,

"priority":<Double_value>,

"activepolicy":<Double_value>

}]}
```

get (all)

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/authenticationradiuspolicy_vpnglobal_binding

Query-parameters:

bulkbindings

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/authenticationradiuspolicy_vpnglobal_binding?**bulkbindings=yes**

NITRO allows you to fetch bindings in bulk.

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "authenticationradiuspolicy_vpnglobal_binding": [ {

"boundto":<String_value>,

"name":<String_value>,

"priority":<Double_value>,

"activepolicy":<Double_value>

}]}
```

count

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/authenticationradiuspolicy_vpnglobal_binding/name_value<String>?**count=yes**

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{"authenticationradiuspolicy_vpnglobal_binding": [ { "__count": "#no"} ] }
```

authenticationradiuspolicy_vpnvserver_binding

Binding object showing the vpnvserver that can be bound to authenticationradiuspolicy.

Properties

(click to see Operations)

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
boundto	<String>	Read-write	The entity name to which policy is bound.
name	<String>	Read-write	Name of the RADIUS authentication policy. Minimum length = 1
priority	<Double>	Read-only	.
activepolicy	<Double>	Read-only	.
__count	<Double>	Read-only	count parameter

Operations

(click to see Properties)

GET| GET (ALL)| COUNT

Some options that you can use for each operations:

- **Getting warnings in response:**NITRO allows you to get warnings in an operation by specifying the "warning" query parameter as "yes". For example, to get warnings while connecting to the Citrix ADC appliance, the URL is as follows:

  http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/login?warning=yes

  If any, the warnings are displayed in the response payload with the HTTP code "209 X-NITRO-WARNING".

- **Authenticated access for individual NITRO operations:**NITRO allows you to logon to the Citrix ADC appliance to perform individual operations. You can use this option instead of creating a NITRO session (using the login object) and then using that session to perform all operations,

  To do this, you must specify the username and password in the request header of the NITRO request as follows:

  X-NITRO-USER:<username>

  X-NITRO-PASS:<password>

  Note: In such cases, make sure that the request header DOES not include the following:

  Cookie:NITRO_AUTH_TOKEN=<tokenvalue>

`*Note:`*

Mandatory parameters are marked in redand placeholder content is marked in <green>.

get

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/authenticationradiuspolicy_vpnvserver_binding/name_value<String>

Query-parameters:

filter

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/authenticationradiuspolicy_vpnvserver_binding/name_value<String>?**filter=property-name1:property-value1,property-name2:property-value2**

Use this query-parameter to get the filtered set of authenticationradiuspolicy_vpnvserver_binding resources configured on Citrix ADC. Filtering can be done on any of the properties of the resource.

pagination

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/authenticationradiuspolicy_vpnvserver_binding/name_value<String>?**pagesize=#no;pageno=#no**

Use this query-parameter to get the authenticationradiuspolicy_vpnvserver_binding resources in chunks.

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "authenticationradiuspolicy_vpnvserver_binding": [ {

"boundto":<String_value>,

"name":<String_value>,

"priority":<Double_value>,

"activepolicy":<Double_value>

}]}
```

get (all)

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/authenticationradiuspolicy_vpnvserver_binding

Query-parameters:

bulkbindings

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/authenticationradiuspolicy_vpnvserver_binding?**bulkbindings=yes**

NITRO allows you to fetch bindings in bulk.

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "authenticationradiuspolicy_vpnvserver_binding": [ {

"boundto":<String_value>,

"name":<String_value>,

"priority":<Double_value>,

"activepolicy":<Double_value>

}]}
```

count

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/authenticationradiuspolicy_vpnvserver_binding/name_value<String>?**count=yes**

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{"authenticationradiuspolicy_vpnvserver_binding": [ { "__count":
"#no"} ] }
```

authenticationsamlaction

Configuration for AAA Saml action resource.

Properties

(click to see Operations)

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
name	<String>	Read-write	Name for the SAML server profile (action).

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
			Must begin with a letter, number, or the underscore character (_), and must contain only letters, numbers, and the hyphen (-), period (.) pound (#), space ( ), at (@), equals (=), colon (:), and underscore characters. Cannot be changed after SAML profile is created. The following requirement applies only to the Citrix ADC CLI: If the name includes one or more spaces, enclose the name in double or single quotation marks (for example, "my authentication action" or 'my authentication action'). Minimum length = 1
samlidpcertname	<String>	Read-write	Name of the SSL certificate used to verify responses

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
			from SAML Identity Provider (IdP). Minimum length = 1
samlsigningcertname	<String>	Read-write	Name of the SSL certificate to sign requests from ServiceProvider (SP) to Identity Provider (IdP). Minimum length = 1
samlredirecturl	<String>	Read-write	URL to which users are redirected for authentication. Minimum length = 1
samlacsindex	<Double>	Read-write	Index/ID of the metadata entry corresponding to this configuration. Default value: 255 Minimum value = 0 Maximum value = 255
samluserfield	<String>	Read-write	SAML user ID, as given in the SAML assertion. Minimum length = 1
samlrejectunsignedassertion	<String>	Read-write	Reject unsigned SAML assertions. ON option results in rejection of Assertion that is received without signature. STRICT option ensures that both

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
			Response and Assertion are signed. OFF allows unsigned Assertions. Default value: ON Possible values = ON, OFF, STRICT
samlissuername	<String>	Read-write	The name to be used in requests sent from Citrix ADC to IdP to uniquely identify Citrix ADC. Minimum length = 1
samltwofactor	<String>	Read-write	Option to enable second factor after SAML. Default value: OFF Possible values = ON, OFF
defaultauthentication group	<String>	Read-write	This is the default group that is chosen when the authentication succeeds in addition to extracted groups.
attribute1	<String>	Read-write	Name of the attribute in SAML Assertion whose value needs to be extracted and stored as attribute1. Maximum length of the extracted attribute is 239 bytes.

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
attribute2	<String>	Read-write	Name of the attribute in SAML Assertion whose value needs to be extracted and stored as attribute2. Maximum length of the extracted attribute is 239 bytes.
attribute3	<String>	Read-write	Name of the attribute in SAML Assertion whose value needs to be extracted and stored as attribute3. Maximum length of the extracted attribute is 239 bytes.
attribute4	<String>	Read-write	Name of the attribute in SAML Assertion whose value needs to be extracted and stored as attribute4. Maximum length of the extracted attribute is 239 bytes.
attribute5	<String>	Read-write	Name of the attribute in SAML Assertion whose value needs to be extracted and stored as attribute5. Maximum length of the extracted attribute is 239 bytes.

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
attribute6	<String>	Read-write	Name of the attribute in SAML Assertion whose value needs to be extracted and stored as attribute6. Maximum length of the extracted attribute is 239 bytes.
attribute7	<String>	Read-write	Name of the attribute in SAML Assertion whose value needs to be extracted and stored as attribute7. Maximum length of the extracted attribute is 239 bytes.
attribute8	<String>	Read-write	Name of the attribute in SAML Assertion whose value needs to be extracted and stored as attribute8. Maximum length of the extracted attribute is 239 bytes.
attribute9	<String>	Read-write	Name of the attribute in SAML Assertion whose value needs to be extracted and stored as attribute9. Maximum length of the extracted attribute is 239 bytes.

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
attribute10	<String>	Read-write	Name of the attribute in SAML Assertion whose value needs to be extracted and stored as attribute10. Maximum length of the extracted attribute is 239 bytes.
attribute11	<String>	Read-write	Name of the attribute in SAML Assertion whose value needs to be extracted and stored as attribute11. Maximum length of the extracted attribute is 239 bytes.
attribute12	<String>	Read-write	Name of the attribute in SAML Assertion whose value needs to be extracted and stored as attribute12. Maximum length of the extracted attribute is 239 bytes.
attribute13	<String>	Read-write	Name of the attribute in SAML Assertion whose value needs to be extracted and stored as attribute13. Maximum length of the extracted attribute is 239 bytes.

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
attribute14	<String>	Read-write	Name of the attribute in SAML Assertion whose value needs to be extracted and stored as attribute14. Maximum length of the extracted attribute is 239 bytes.
attribute15	<String>	Read-write	Name of the attribute in SAML Assertion whose value needs to be extracted and stored as attribute15. Maximum length of the extracted attribute is 239 bytes.
attribute16	<String>	Read-write	Name of the attribute in SAML Assertion whose value needs to be extracted and stored as attribute16. Maximum length of the extracted attribute is 239 bytes.
attributes	<String>	Read-write	List of attribute names separated by ',' which needs to be extracted. Note that preceeding and trailing spaces will be removed.

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
			Attribute name can be 127 bytes and total length of this string should not cross 2047 bytes. These attributes have multi-value support separated by ',' and stored as key-value pair in AAA session.
signaturealg	<String>	Read-write	Algorithm to be used to sign/verify SAML transactions. Default value: RSA-SHA256 Possible values = RSA-SHA1, RSA-SHA256
digestmethod	<String>	Read-write	Algorithm to be used to compute/verify digest for SAML transactions. Default value: SHA256 Possible values = SHA1, SHA256
requestedauthncontext	<String>	Read-write	This element specifies the authentication context requirements of authentication

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
			statements returned in the response. Default value: exact Possible values = exact, minimum, maximum, better
authnctxclassref	<String[]>	Read-write	This element specifies the authentication class types that are requested from IdP (IdentityProvider). InternetProtocol: This is applicable when a principal is authenticated through the use of a provided IP address. InternetProtocolPassword: This is applicable when a principal is authenticated through the use of a provided IP address, in addition to a username/ password. Kerberos: This is applicable when the principal has authenticated using a password to a

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
			local authentication authority, in order to acquire a Kerberos ticket. MobileOneFactorUnregistered: This indicates authentication of the mobile device without requiring explicit end-user interaction. MobileTwoFactorUnregistered: This indicates two-factor based authentication during mobile customer registration process, such as secure device and user PIN. MobileOneFactorContract: Reflects mobile contract customer registration procedures and a single factor authentication.

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
			MobileTwoFactorContract: Reflects mobile contract customer registration procedures and a two-factor based authentication. Password: This class is applicable when a principal authenticates using password over unprotected http session. PasswordProtectedTransport: This class is applicable when a principal authenticates to an authentication authority through the presentation of a password over a protected session. PreviousSession: This class is applicable when a principal had authenticated to an authentication authority at some point in the past using any

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
			authentication context. X509: This indicates that the principal authenticated by means of a digital signature where the key was validated as part of an X.509 Public Key Infrastructure. PGP: This indicates that the principal authenticated by means of a digital signature where the key was validated as part of a PGP Public Key Infrastructure. SPKI: This indicates that the principal authenticated by means of a digital signature where the key was validated via an SPKI Infrastructure. XMLDSig: This indicates that the principal authenticated by means of a digital

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
			signature according to the processing rules specified in the XML Digital Signature specification. Smartcard: This indicates that the principal has authenticated using smartcard. SmartcardPKI: This class is applicable when a principal authenticates to an authentication authority through a two-factor authentication mechanism using a smartcard with enclosed private key and a PIN. SoftwarePKI: This class is applicable when a principal uses an X.509 certificate stored in software to authenticate to the authentication authority.

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
			Telephony: This class is used to indicate that the principal authenticated via the provision of a fixed-line telephone number, transported via a telephony protocol such as ADSL. NomadTelephony: Indicates that the principal is "roaming" and authenticates via the means of the line number, a user suffix, and a password element. PersonalTelephony: This class is used to indicate that the principal authenticated via the provision of a fixed-line telephone. AuthenticatedTelephony: Indicates that the principal authenticated via the means of the line number, a user

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
			suffix, and a password element. SecureRemotePassword: This class is applicable when the authentication was performed by means of Secure Remote Password. TLSClient: This class indicates that the principal authenticated by means of a client certificate, secured with the SSL/TLS transport. TimeSyncToken: This is applicable when a principal authenticates through a time synchronization token. Unspecified: This indicates that the authentication was performed by unspecified means. Windows: This indicates that Windows integrated authentication is

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
			utilized for authentication. Possible values = InternetProtocol, InternetProtocolPassword, Kerberos, MobileOneFactorUnregistered, MobileTwoFactorUnregistered, MobileOneFactorContract, MobileTwoFactorContract, Password, PasswordProtectedTransport, PreviousSession, X509, PGP, SPKI, XMLDSig, Smartcard, SmartcardPKI, SoftwarePKI, Telephony, NomadTelephony, PersonalTelephony, AuthenticatedTelephony, SecureRemotePassword, TLSClient, TimeSyncToken, Unspecified, Windows
samlbinding	<String>	Read-write	This element specifies the

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
			transport mechanism of saml messages. Default value: POST Possible values = REDIRECT, POST, ARTIFACT
attributeconsumingserviceindex	<Double>	Read-write	Index/ID of the attribute specification at Identity Provider (IdP). IdP will locate attributes requested by SP using this index and send those attributes in Assertion. Default value: 255 Minimum value = 0 Maximum value = 255
sendthumbprint	<String>	Read-write	Option to send thumbprint instead of x509 certificate in SAML request. Default value: OFF Possible values = ON, OFF
enforceusername	<String>	Read-write	Option to choose whether the username that is extracted from SAML assertion can be edited in login page

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
			while doing second factor. Default value: ON Possible values = ON, OFF
logouturl	<String>	Read-write	SingleLogout URL on IdP to which logoutRequest will be sent on Citrix ADC session cleanup.
artifactresolutionserviceurl	<String>	Read-write	URL of the Artifact Resolution Service on IdP to which Citrix ADC will post artifact to get actual SAML token.
skewtime	<Double>	Read-write	This option specifies the allowed clock skew in number of minutes that Citrix ADC ServiceProvider allows on an incoming assertion. For example, if skewTime is 10, then assertion would be valid from (current time - 10) min to (current time + 10) min, ie 20min in all. Default value: 5
logoutbinding	<String>	Read-write	This element specifies the transport mechanism

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
			of saml logout messages. Default value: POST Possible values = REDIRECT, POST
forceauthn	<String>	Read-write	Option that forces authentication at the Identity Provider (IdP) that receives Citrix ADC's request. Default value: OFF Possible values = ON, OFF
groupnamefield	<String>	Read-write	Name of the tag in assertion that contains user groups.
audience	<String>	Read-write	Audience for which assertion sent by IdP is applicable. This is typically entity name or url that represents ServiceProvider.
metadataurl	<String>	Read-write	This URL is used for obtaining saml metadata.
metadatarefreshinterval	<Double>	Read-write	Interval in minutes for fetching metadata from specified metadata URL. Default value: 36000

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
storesamlresponse	<String>	Read-write	Option to store entire SAML Response through the life of user session. Default value: OFF Possible values = ON, OFF
metadataimportstatus	<String>	Read-only	Describes metadata import status. Default value: SUCCESS Possible values = INIT, SUCCESS, FAIL
__count	<Double>	Read-only	count parameter

Operations

(click to see Properties)

ADD| DELETE| UPDATE| UNSET| GET (ALL)| GET| COUNT

Some options that you can use for each operations:

- **Getting warnings in response:**NITRO allows you to get warnings in an operation by specifying the "warning" query parameter as "yes". For example, to get warnings while connecting to the Citrix ADC appliance, the URL is as follows:

  http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/login?warning=yes

  If any, the warnings are displayed in the response payload with the HTTP code "209 X-NITRO-WARNING".

- **Authenticated access for individual NITRO operations:**NITRO allows you to logon to the Citrix ADC appliance to perform individual operations. You can use this option instead of

creating a NITRO session (using the login object) and then using that session to perform all operations,

To do this, you must specify the username and password in the request header of the NITRO request as follows:

X-NITRO-USER:<username>

X-NITRO-PASS:<password>

Note: In such cases, make sure that the request header DOES not include the following:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue>

`*Note:`*

Mandatory parameters are marked in redand placeholder content is marked in <green>.

add

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/authenticationsamlaction

HTTP Method:POST

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Content-Type:application/json

Request Payload:

```
{"authenticationsamlaction":{

<b>"name":<String_value>,

</b>"samlidpcertname":<String_value>,

"samlsigningcertname":<String_value>,

"samlredirecturl":<String_value>,

"samlacsindex":<Double_value>,

"samluserfield":<String_value>,
```

```
"samlrejectunsignedassertion":<String_value>,

"samlissuername":<String_value>,

"samltwofactor":<String_value>,

"defaultauthenticationgroup":<String_value>,

"attribute1":<String_value>,

"attribute2":<String_value>,

"attribute3":<String_value>,

"attribute4":<String_value>,

"attribute5":<String_value>,

"attribute6":<String_value>,

"attribute7":<String_value>,

"attribute8":<String_value>,

"attribute9":<String_value>,

"attribute10":<String_value>,

"attribute11":<String_value>,

"attribute12":<String_value>,

"attribute13":<String_value>,

"attribute14":<String_value>,

"attribute15":<String_value>,

"attribute16":<String_value>,

"attributes":<String_value>,

"signaturealg":<String_value>,
```

```
"digestmethod":<String_value>,

"requestedauthncontext":<String_value>,

"authnctxclassref":<String[]_value>,

"samlbinding":<String_value>,

"attributeconsumingserviceindex":<Double_value>,

"sendthumbprint":<String_value>,

"enforceusername":<String_value>,

"logouturl":<String_value>,

"artifactresolutionserviceurl":<String_value>,

"skewtime":<Double_value>,

"logoutbinding":<String_value>,

"forceauthn":<String_value>,

"groupnamefield":<String_value>,

"audience":<String_value>,

"metadataurl":<String_value>,

"metadatarefreshinterval":<Double_value>,

"storesamlresponse":<String_value>

}}
```

Response:

HTTP Status Code on Success: 201 Created HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error

delete

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/authenticationsamlaction/name_value<String>

HTTP Method:DELETE

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue>

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error

update

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/authenticationsamlaction

HTTP Method:PUT

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Content-Type:application/json

Request Payload:

```
{"authenticationsamlaction":{

<b>"name":<String_value>,

</b>"samlidpcertname":<String_value>,

"samlsigningcertname":<String_value>,

"samlredirecturl":<String_value>,

"samlacsindex":<Double_value>,

"samluserfield":<String_value>,

"samlrejectunsignedassertion":<String_value>,
```

```
"samlissuername":<String_value>,

"samltwofactor":<String_value>,

"defaultauthenticationgroup":<String_value>,

"attribute1":<String_value>,

"attribute2":<String_value>,

"attribute3":<String_value>,

"attribute4":<String_value>,

"attribute5":<String_value>,

"attribute6":<String_value>,

"attribute7":<String_value>,

"attribute8":<String_value>,

"attribute9":<String_value>,

"attribute10":<String_value>,

"attribute11":<String_value>,

"attribute12":<String_value>,

"attribute13":<String_value>,

"attribute14":<String_value>,

"attribute15":<String_value>,

"attribute16":<String_value>,

"attributes":<String_value>,

"signaturealg":<String_value>,

"digestmethod":<String_value>,
```

```
"requestedauthncontext":<String_value>,

"authnctxclassref":<String[]_value>,

"samlbinding":<String_value>,

"attributeconsumingserviceindex":<Double_value>,

"sendthumbprint":<String_value>,

"enforceusername":<String_value>,

"logouturl":<String_value>,

"artifactresolutionserviceurl":<String_value>,

"skewtime":<Double_value>,

"logoutbinding":<String_value>,

"forceauthn":<String_value>,

"groupnamefield":<String_value>,

"audience":<String_value>,

"metadataurl":<String_value>,

"metadatarefreshinterval":<Double_value>,

"storesamlresponse":<String_value>

}}
```

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error

unset

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/authenticationsamlaction?**action=unset**

HTTP Method:POST

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Content-Type:application/json

Request Payload:

```
{"authenticationsamlaction":{

<b>"name":<String_value>,

</b>"samlsigningcertname":true,

"samlredirecturl":true,

"samlacsindex":true,

"samluserfield":true,

"samlrejectunsignedassertion":true,

"samlissuername":true,

"samltwofactor":true,

"defaultauthenticationgroup":true,

"attribute1":true,

"attribute2":true,

"attribute3":true,

"attribute4":true,

"attribute5":true,
```

```
"attribute6":true,

"attribute7":true,

"attribute8":true,

"attribute9":true,

"attribute10":true,

"attribute11":true,

"attribute12":true,

"attribute13":true,

"attribute14":true,

"attribute15":true,

"attribute16":true,

"attributes":true,

"signaturealg":true,

"digestmethod":true,

"requestedauthncontext":true,

"authnctxclassref":true,

"samlbinding":true,

"attributeconsumingserviceindex":true,

"sendthumbprint":true,

"enforceusername":true,

"logouturl":true,

"artifactresolutionserviceurl":true,
```

```
"skewtime":true,

"logoutbinding":true,

"forceauthn":true,

"groupnamefield":true,

"audience":true,

"metadataurl":true,

"metadatarefreshinterval":true,

"storesamlresponse":true

}}
```

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error

get (all)

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/authenticationsamlaction

Query-parameters:

attrs

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/authenticationsamlaction?**attrs=property-name1,property-name2**

Use this query parameter to specify the resource details that you want to retrieve.

filter

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/authenticationsamlaction?**filter=property-name1:property-val1,property-name2:property-val2**

Use this query-parameter to get the filtered set of authenticationsamlaction resources configured on Citrix ADC. Filtering can be done on any of the properties of the resource.

view

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/authenticationsamlaction?**view=summary**

Note: By default, the retrieved results are displayed in detail view (?view=detail).

pagination

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/authenticationsamlaction?**pagesize=#no;pageno=#no**

Use this query-parameter to get the authenticationsamlaction resources in chunks.

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "authenticationsamlaction": [ {

"name":<String_value>,

"samlidpcertname":<String_value>,

"samlsigningcertname":<String_value>,

"samlredirecturl":<String_value>,

"samlacsindex":<Double_value>,
```

```
"samluserfield":<String_value>,

"samlrejectunsignedassertion":<String_value>,

"samlissuername":<String_value>,

"samltwofactor":<String_value>,

"defaultauthenticationgroup":<String_value>,

"attribute1":<String_value>,

"attribute2":<String_value>,

"attribute3":<String_value>,

"attribute4":<String_value>,

"attribute5":<String_value>,

"attribute6":<String_value>,

"attribute7":<String_value>,

"attribute8":<String_value>,

"attribute9":<String_value>,

"attribute10":<String_value>,

"attribute11":<String_value>,

"attribute12":<String_value>,

"attribute13":<String_value>,

"attribute14":<String_value>,

"attribute15":<String_value>,

"attribute16":<String_value>,

"attributes":<String_value>,
```

```
"signaturealg":<String_value>,

"digestmethod":<String_value>,

"requestedauthncontext":<String_value>,

"authnctxclassref":<String[]_value>,

"samlbinding":<String_value>,

"attributeconsumingserviceindex":<Double_value>,

"sendthumbprint":<String_value>,

"enforceusername":<String_value>,

"logouturl":<String_value>,

"artifactresolutionserviceurl":<String_value>,

"skewtime":<Double_value>,

"logoutbinding":<String_value>,

"forceauthn":<String_value>,

"groupnamefield":<String_value>,

"audience":<String_value>,

"metadataurl":<String_value>,

"metadatarefreshinterval":<Double_value>,

"metadataimportstatus":<String_value>,

"storesamlresponse":<String_value>

}]}
```

get

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/authenticationsamlaction/name_value<String>

Query-parameters:

attrs

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/authenticationsamlaction/name_value<String>?**attrs=property-name1,property-name2**

Use this query parameter to specify the resource details that you want to retrieve.

view

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/authenticationsamlaction/name_value<String>?**view=summary**

Note: By default, the retrieved results are displayed in detail view (?view=detail).

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "authenticationsamlaction": [ {

"name":<String_value>,

"samlidpcertname":<String_value>,
```

```
"samlsigningcertname":<String_value>,

"samlredirecturl":<String_value>,

"samlacsindex":<Double_value>,

"samluserfield":<String_value>,

"samlrejectunsignedassertion":<String_value>,

"samlissuername":<String_value>,

"samltwofactor":<String_value>,

"defaultauthenticationgroup":<String_value>,

"attribute1":<String_value>,

"attribute2":<String_value>,

"attribute3":<String_value>,

"attribute4":<String_value>,

"attribute5":<String_value>,

"attribute6":<String_value>,

"attribute7":<String_value>,

"attribute8":<String_value>,

"attribute9":<String_value>,

"attribute10":<String_value>,

"attribute11":<String_value>,

"attribute12":<String_value>,

"attribute13":<String_value>,

"attribute14":<String_value>,
```

```
"attribute15":<String_value>,

"attribute16":<String_value>,

"attributes":<String_value>,

"signaturealg":<String_value>,

"digestmethod":<String_value>,

"requestedauthncontext":<String_value>,

"authnctxclassref":<String[]_value>,

"samlbinding":<String_value>,

"attributeconsumingserviceindex":<Double_value>,

"sendthumbprint":<String_value>,

"enforceusername":<String_value>,

"logouturl":<String_value>,

"artifactresolutionserviceurl":<String_value>,

"skewtime":<Double_value>,

"logoutbinding":<String_value>,

"forceauthn":<String_value>,

"groupnamefield":<String_value>,

"audience":<String_value>,

"metadataurl":<String_value>,

"metadatarefreshinterval":<Double_value>,

"metadataimportstatus":<String_value>,

"storesamlresponse":<String_value>
```

```
}]}
```

count

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/authenticationsamlaction?**count=yes**

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "authenticationsamlaction": [ { "__count": "#no"} ] }
```

authenticationsamlidppolicy

Configuration for AAA Saml IdentityProvider (IdP) policy resource.

Properties

(click to see Operations)

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
name	<String>	Read-write	Name for the SAML Identity Provider (IdP) authentication policy. This is used for configuring Citrix

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
			ADC as SAML Identity Provider. Must begin with an ASCII alphanumeric or underscore (_) character, and must contain only ASCII alphanumeric, underscore, hash (#), period (.), space, colon (:), at (@), equals (=), and hyphen (-) characters. Cannot be changed after the policy is created. The following requirement applies only to the Citrix ADC CLI: If the name includes one or more spaces, enclose the name in double or single quotation marks (for example, "my policy" or 'my policy'). Minimum length = 1
rule	<String>	Read-write	Expression which is evaluated to choose a profile for authentication. The following requirements apply

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
			only to the Citrix ADC CLI: * If the expression includes one or more spaces, enclose the entire expression in double quotation marks. * If the expression itself includes double quotation marks, escape the quotations by using the \ character. * Alternatively, you can use single quotation marks to enclose the rule, in which case you do not have to escape the double quotation marks. Minimum length = 1
action	<String>	Read-write	Name of the profile to apply to requests or connections that match this policy. Minimum length = 1
undefaction	<String>	Read-write	Action to perform if the result of policy evaluation is undefined (UNDEF).

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
			An UNDEF event indicates an internal error condition. Only the above built-in actions can be used.
comment	<String>	Read-write	Any comments to preserve information about this policy.
logaction	<String>	Read-write	Name of messagelog action to use when a request matches this policy.
newname	<String>	Read-write	New name for the SAML IdentityProvider policy. Must begin with a letter, number, or the underscore character (_), and must contain only letters, numbers, and the hyphen (-), period (.) hash (#), space ( ), at (@), equals (=), colon (:), and underscore characters. The following requirement applies only to the Citrix ADC CLI:

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
			If the name includes one or more spaces, enclose the name in double or single quotation marks (for example, "my samlidppolicy policy" or 'my samlidppolicy policy'). Minimum length = 1
gotopriorityexpression	<String>	Read-only	Expression specifying the priority of the next policy which will get evaluated if the current policy rule evaluates to TRUE.
hits	<Double>	Read-only	Number of hits.
__count	<Double>	Read-only	count parameter

Operations

(click to see Properties)

ADD| DELETE| UPDATE| UNSET| RENAME| GET (ALL)| GET| COUNT

Some options that you can use for each operations:

- **Getting warnings in response:**NITRO allows you to get warnings in an operation by specifying the "warning" query parameter as "yes". For example, to get warnings while connecting to the Citrix ADC appliance, the URL is as follows:

  http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/login?warning=yes

  If any, the warnings are displayed in the response payload with the HTTP code "209 X-NITRO-WARNING".

- **Authenticated access for individual NITRO operations:**NITRO allows you to logon to the Citrix ADC appliance to perform individual operations. You can use this option instead of creating a NITRO session (using the login object) and then using that session to perform all operations,

  To do this, you must specify the username and password in the request header of the NITRO request as follows:

  X-NITRO-USER:<username>

  X-NITRO-PASS:<password>

  Note: In such cases, make sure that the request header DOES not include the following:

  Cookie:NITRO_AUTH_TOKEN=<tokenvalue>

`*Note:`*

Mandatory parameters are marked in redand placeholder content is marked in <green>.

add

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/authenticationsamlidppolicy

HTTP Method:POST

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Content-Type:application/json

Request Payload:

```
{"authenticationsamlidppolicy":{

<b>"name":<String_value>,

</b><b>"rule":<String_value>,

</b><b>"action":<String_value>,

</b>"undefaction":<String_value>,

"comment":<String_value>,
```

```
"logaction":<String_value>

}}
```

Response:

HTTP Status Code on Success: 201 Created HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error

delete

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/authenticationsamlidppolicy/name_value<String>

HTTP Method:DELETE

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue>

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error

update

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/authenticationsamlidppolicy

HTTP Method:PUT

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Content-Type:application/json

Request Payload:

```
{"authenticationsamlidppolicy":{
```

```
<b>"name":<String_value>,

</b>"rule":<String_value>,

"action":<String_value>,

"undefaction":<String_value>,

"comment":<String_value>,

"logaction":<String_value>

}}
```

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error

unset

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/authenticationsamlidppolicy?**action=unset**

HTTP Method:POST

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Content-Type:application/json

Request Payload:

```
{"authenticationsamlidppolicy":{

<b>"name":<String_value>,

</b>"undefaction":true,

"comment":true,
```

```
"logaction":true

}}
```

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error

rename

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/authenticationsamlidppolicy?**action=rename**

HTTP Method:POST

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Content-Type:application/json

Request Payload:

```
{"authenticationsamlidppolicy":{

<b>"name":<String_value>,

</b><b>"newname":<String_value>

</b>}}
```

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error

get (all)

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/authenticationsamlidppolicy

Query-parameters:

attrs

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/authenticationsamlidppolicy?**attrs=property-name1,property-name2**

Use this query parameter to specify the resource details that you want to retrieve.

filter

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/authenticationsamlidppolicy?**filter=property-name1:property-val1,property-name2:property-val2**

Use this query-parameter to get the filtered set of authenticationsamlidppolicy resources configured on Citrix ADC. Filtering can be done on any of the properties of the resource.

view

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/authenticationsamlidppolicy?**view=summary**

Note: By default, the retrieved results are displayed in detail view (?view=detail).

pagination

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/authenticationsamlidppolicy?**pagesize=#no;pageno=#no**

Use this query-parameter to get the authenticationsamlidppolicy resources in chunks.

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "authenticationsamlidppolicy": [ {

"name":<String_value>,

"rule":<String_value>,

"action":<String_value>,

"undefaction":<String_value>,

"comment":<String_value>,

"logaction":<String_value>,

"gotopriorityexpression":<String_value>,

"hits":<Double_value>

}]}
```

get

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/authenticationsamlidppolicy/name_value<String>

Query-parameters:

attrs

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/authenticationsamlidppolicy/name_value<String>?**attrs=property-name1,property-name2**

Use this query parameter to specify the resource details that you want to retrieve.

view

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/authenticationsamlidppolicy/name_value<String>?**view=summary**

Note: By default, the retrieved results are displayed in detail view (?view=detail).

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "authenticationsamlidppolicy": [ {

"name":<String_value>,

"rule":<String_value>,

"action":<String_value>,

"undefaction":<String_value>,

"comment":<String_value>,

"logaction":<String_value>,

"gotopriorityexpression":<String_value>,

"hits":<Double_value>

}]}
```

count

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/authenticationsamlidppolicy?**count=yes**

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "authenticationsamlidppolicy": [ { "__count": "#no"} ] }
```

authenticationsamlidppolicy_authenticationvserver_binding

Binding object showing the authenticationvserver that can be bound to authenticationsamlidppolicy.

Properties

(click to see Operations)

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
boundto	<String>	Read-write	The entity name to which policy is bound.
name	<String>	Read-write	Name of the SAML IdentityProvider (IdP) policy for which to display detailed information.

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
			Minimum length = 1
priority	<Double>	Read-only	.
activepolicy	<Double>	Read-only	.
__count	<Double>	Read-only	count parameter

Operations

(click to see Properties)

GET| GET (ALL)| COUNT

Some options that you can use for each operations:

- **Getting warnings in response:**NITRO allows you to get warnings in an operation by specifying the "warning" query parameter as "yes". For example, to get warnings while connecting to the Citrix ADC appliance, the URL is as follows:

  http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/login?warning=yes

  If any, the warnings are displayed in the response payload with the HTTP code "209 X-NITRO-WARNING".

- **Authenticated access for individual NITRO operations:**NITRO allows you to logon to the Citrix ADC appliance to perform individual operations. You can use this option instead of creating a NITRO session (using the login object) and then using that session to perform all operations,

  To do this, you must specify the username and password in the request header of the NITRO request as follows:

  X-NITRO-USER:<username>

  X-NITRO-PASS:<password>

  Note: In such cases, make sure that the request header DOES not include the following:

  Cookie:NITRO_AUTH_TOKEN=<tokenvalue>

`*Note:` *

Mandatory parameters are marked in redand placeholder content is marked in <green>.

get

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/authenticationsamlidppolicy_authenticationvserver_binding/name_value<String>

Query-parameters:

filter

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/authenticationsamlidppolicy_authenticationvserver_binding/name_value<String>?**filter=property-name1:property-value1,property-name2:property-value2**

Use this query-parameter to get the filtered set of authenticationsamlidppolicy_authenticationvserver_binding resources configured on Citrix ADC. Filtering can be done on any of the properties of the resource.

pagination

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/authenticationsamlidppolicy_authenticationvserver_binding/name_value<String>?**pagesize=#no;pageno=#no**

Use this query-parameter to get the authenticationsamlidppolicy_authenticationvserver_binding resources in chunks.

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "authenticationsamlidppolicy_authenticationvserver_binding": [ {

"boundto":<String_value>,

"name":<String_value>,

"priority":<Double_value>,

"activepolicy":<Double_value>

}]}
```

get (all)

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/authenticationsamlidppolicy_authenticationvserver_binding

Query-parameters:

bulkbindings

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/authenticationsamlidppolicy_authenticationvserver_binding?**bulkbindings=yes**

NITRO allows you to fetch bindings in bulk.

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "authenticationsamlidppolicy_authenticationvserver_binding": [ {

"boundto":<String_value>,

"name":<String_value>,

"priority":<Double_value>,

"activepolicy":<Double_value>

}]}
```

count

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/authenticationsamlidppolicy_authenticationvserver_binding/name_value<String>?**count=yes**

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{"authenticationsamlidppolicy_authenticationvserver_binding":
[ { "__count": "#no"} ] }
```

authenticationsamlidppolicy_binding

Binding object which returns the resources bound to authenticationsamlidppolicy.

Properties

(click to see Operations)

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
name	<String>	Read-write	Name of the SAML IdentityProvider (IdP) policy for which to display detailed information. Minimum length = 1
authenticationsamlidppolicy_vpnvserver_binding	<authenticationsamlidppolicy_vpnvserver_binding[]>	Read-only	vpnvserver that can be bound to authenticationsamlidppolicy.
authenticationsamlidppolicy_authenticationvserver_binding	<authenticationsamlidppolicy_authenticationvserver_binding[]>	Read-only	authenticationvserver that can be bound to authenticationsamlidppolicy.

Operations

(click to see Properties)

GET| GET (ALL)

Some options that you can use for each operations:

- **Getting warnings in response:**NITRO allows you to get warnings in an operation by specifying the "warning" query parameter as "yes". For example, to get warnings while connecting to the Citrix ADC appliance, the URL is as follows:

  http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/login?warning=yes

  If any, the warnings are displayed in the response payload with the HTTP code "209 X-NITRO-WARNING".

- **Authenticated access for individual NITRO operations:**NITRO allows you to logon to the Citrix ADC appliance to perform individual operations. You can use this option instead of creating a NITRO session (using the login object) and then using that session to perform all operations,

  To do this, you must specify the username and password in the request header of the NITRO request as follows:

  X-NITRO-USER:<username>

  X-NITRO-PASS:<password>

  Note: In such cases, make sure that the request header DOES not include the following:

  Cookie:NITRO_AUTH_TOKEN=<tokenvalue>

`*Note:`*

Mandatory parameters are marked in redand placeholder content is marked in <green>.

get

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/authenticationsamlidppolicy_binding/name_value<String>

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "authenticationsamlidppolicy_binding": [ {
```

```
"name":<String_value>,

"authenticationsamlidppolicy_authenticationvserver_binding":<authenticatio
nsamlidppolicy_authenticationvserver_binding[]_value>,

"authenticationsamlidppolicy_vpnvserver_binding":<authenticationsamlidppol
icy_vpnvserver_binding[]_value>

}]}
```

get (all)

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/authenticationsamlidppolicy_binding

Query-parameters:

bulkbindings

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/authenticationsamlidppolicy_binding?**bulkbindings=yes**

NITRO allows you to fetch bindings in bulk.

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "authenticationsamlidppolicy_binding": [ {
```

```
"name":<String_value>,

"authenticationsamlidppolicy_authenticationvserver_binding":<authenticatio
nsamlidppolicy_authenticationvserver_binding[]_value>,

"authenticationsamlidppolicy_vpnvserver_binding":<authenticationsamlidppol
icy_vpnvserver_binding[]_value>

}]}
```

authenticationsamlidppolicy_vpnvserver_binding

Binding object showing the vpnvserver that can be bound to authenticationsamlidppolicy.

Properties

(click to see Operations)

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
boundto	<String>	Read-write	The entity name to which policy is bound.
name	<String>	Read-write	Name of the SAML IdentityProvider (IdP) policy for which to display detailed information. Minimum length = 1
priority	<Double>	Read-only	.
activepolicy	<Double>	Read-only	.
__count	<Double>	Read-only	count parameter

Operations

(click to see Properties)

GET| GET (ALL)| COUNT

Some options that you can use for each operations:

- **Getting warnings in response:**NITRO allows you to get warnings in an operation by specifying the "warning" query parameter as "yes". For example, to get warnings while connecting to the Citrix ADC appliance, the URL is as follows:

  http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/login?warning=yes

  If any, the warnings are displayed in the response payload with the HTTP code "209 X-NITRO-WARNING".

- **Authenticated access for individual NITRO operations:**NITRO allows you to logon to the Citrix ADC appliance to perform individual operations. You can use this option instead of creating a NITRO session (using the login object) and then using that session to perform all operations,

  To do this, you must specify the username and password in the request header of the NITRO request as follows:

  X-NITRO-USER:<username>

  X-NITRO-PASS:<password>

  Note: In such cases, make sure that the request header DOES not include the following:

  Cookie:NITRO_AUTH_TOKEN=<tokenvalue>

`*Note:`*

Mandatory parameters are marked in redand placeholder content is marked in <green>.

get

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/authenticationsamlidppolicy_vpnvserver_binding/name_value<String>

Query-parameters:

filter

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/authenticationsamlidppolicy_vpnvserver_binding/name_value<String>?**filter=property-name1:property-value1,property-name2:property-value2**

Use this query-parameter to get the filtered set of authenticationsamlidppolicy_vpnvserver_binding resources configured on Citrix ADC. Filtering can be done on any of the properties of the resource.

pagination

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/authenticationsamlidppolicy_vpnvserver_binding/name_value<String>?**pagesize=#no;pageno=#no**

Use this query-parameter to get the authenticationsamlidppolicy_vpnvserver_binding resources in chunks.

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "authenticationsamlidppolicy_vpnvserver_binding": [ {

"boundto":<String_value>,

"name":<String_value>,

"priority":<Double_value>,

"activepolicy":<Double_value>

}]}
```

get (all)

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/authenticationsamlidppolicy_vpnvserver_binding

Query-parameters:

bulkbindings

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/authenticationsamlidppolicy_vpnvserver_binding?**bulkbindings=yes**

NITRO allows you to fetch bindings in bulk.

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "authenticationsamlidppolicy_vpnvserver_binding": [ {

"boundto":<String_value>,

"name":<String_value>,

"priority":<Double_value>,

"activepolicy":<Double_value>

}]}
```

count

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/authenticationsamlidppolicy_vpnvserver_binding/name_value<String>?**count=yes**

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{"authenticationsamlidppolicy_vpnvserver_binding": [ { "__count":
"#no"} ] }
```

authenticationsamlidpprofile

Configuration for AAA Saml IdentityProvider (IdP) profile resource.

Properties

(click to see Operations)

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
name	<String>	Read-write	Name for the new saml single sign-on profile. Must begin with an ASCII alphanumeric or underscore (_) character, and must

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
			contain only ASCII alphanumeric, underscore, hash (#), period (.), space, colon (:), at (@), equals (=), and hyphen (-) characters. Cannot be changed after an action is created. The following requirement applies only to the Citrix ADC CLI: If the name includes one or more spaces, enclose the name in double or single quotation marks (for example, "my action" or 'my action'). Minimum length = 1
samlspcertname	<String>	Read-write	Name of the SSL certificate of SAML Relying Party. This certificate is used to verify signature of the incoming AuthnRequest from a Relying Party or Service Provider. Minimum length = 1
samlidpcertname	<String>	Read-write	Name of the certificate used to sign the

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
			SAMLResposne that is sent to Relying Party or Service Provider after successful authentication. Minimum length = 1
assertionconsumerserviceurl	<String>	Read-write	URL to which the assertion is to be sent. Minimum length = 1
sendpassword	<String>	Read-write	Option to send password in assertion. Default value: OFF Possible values = ON, OFF
samlissuername	<String>	Read-write	The name to be used in requests sent from Citrix ADC to IdP to uniquely identify Citrix ADC. Minimum length = 1
rejectunsignedrequests	<String>	Read-write	Option to Reject unsigned SAML Requests. ON option denies any authentication requests that arrive without signature. Default value: ON Possible values = ON, OFF

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
signaturealg	<String>	Read-write	Algorithm to be used to sign/verify SAML transactions. Default value: RSA-SHA256 Possible values = RSA-SHA1, RSA-SHA256
digestmethod	<String>	Read-write	Algorithm to be used to compute/verify digest for SAML transactions. Default value: SHA256 Possible values = SHA1, SHA256
audience	<String>	Read-write	Audience for which assertion sent by IdP is applicable. This is typically entity name or url that represents ServiceProvider.
nameidformat	<String>	Read-write	Format of Name Identifier sent in Assertion. Default value: transient Possible values = Unspecified, emailAddress, X509SubjectName, WindowsDomainQualifiedName,

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
			kerberos, entity, persistent, transient
nameidexpr	<String>	Read-write	Expression that will be evaluated to obtain NameIdentifier to be sent in assertion. Maximum length = 128
attribute1	<String>	Read-write	Name of attribute1 that needs to be sent in SAML Assertion.
attribute1expr	<String>	Read-write	Expression that will be evaluated to obtain attribute1's value to be sent in Assertion. Maximum length = 128
attribute1friendlyname	<String>	Read-write	User-Friendly Name of attribute1 that needs to be sent in SAML Assertion.
attribute1format	<String>	Read-write	Format of Attribute1 to be sent in Assertion. Possible values = URI, Basic
attribute2	<String>	Read-write	Name of attribute2 that needs to be sent in SAML Assertion.
attribute2expr	<String>	Read-write	Expression that will be evaluated to obtain attribute2's

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
			value to be sent in Assertion. Maximum length = 128
attribute2friendlyname	<String>	Read-write	User-Friendly Name of attribute2 that needs to be sent in SAML Assertion.
attribute2format	<String>	Read-write	Format of Attribute2 to be sent in Assertion. Possible values = URI, Basic
attribute3	<String>	Read-write	Name of attribute3 that needs to be sent in SAML Assertion.
attribute3expr	<String>	Read-write	Expression that will be evaluated to obtain attribute3's value to be sent in Assertion. Maximum length = 128
attribute3friendlyname	<String>	Read-write	User-Friendly Name of attribute3 that needs to be sent in SAML Assertion.
attribute3format	<String>	Read-write	Format of Attribute3 to be sent in Assertion. Possible values = URI, Basic

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
attribute4	<String>	Read-write	Name of attribute4 that needs to be sent in SAML Assertion.
attribute4expr	<String>	Read-write	Expression that will be evaluated to obtain attribute4's value to be sent in Assertion. Maximum length = 128
attribute4friendlyname	<String>	Read-write	User-Friendly Name of attribute4 that needs to be sent in SAML Assertion.
attribute4format	<String>	Read-write	Format of Attribute4 to be sent in Assertion. Possible values = URI, Basic
attribute5	<String>	Read-write	Name of attribute5 that needs to be sent in SAML Assertion.
attribute5expr	<String>	Read-write	Expression that will be evaluated to obtain attribute5's value to be sent in Assertion. Maximum length = 128
attribute5friendlyname	<String>	Read-write	User-Friendly Name of attribute5 that needs to be sent in SAML Assertion.

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
attribute5format	<String>	Read-write	Format of Attribute5 to be sent in Assertion. Possible values = URI, Basic
attribute6	<String>	Read-write	Name of attribute6 that needs to be sent in SAML Assertion.
attribute6expr	<String>	Read-write	Expression that will be evaluated to obtain attribute6's value to be sent in Assertion. Maximum length = 128
attribute6friendlyname	<String>	Read-write	User-Friendly Name of attribute6 that needs to be sent in SAML Assertion.
attribute6format	<String>	Read-write	Format of Attribute6 to be sent in Assertion. Possible values = URI, Basic
attribute7	<String>	Read-write	Name of attribute7 that needs to be sent in SAML Assertion.
attribute7expr	<String>	Read-write	Expression that will be evaluated to obtain attribute7's value to be sent in Assertion. Maximum length = 128

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
attribute7friendlyname	<String>	Read-write	User-Friendly Name of attribute7 that needs to be sent in SAML Assertion.
attribute7format	<String>	Read-write	Format of Attribute7 to be sent in Assertion. Possible values = URI, Basic
attribute8	<String>	Read-write	Name of attribute8 that needs to be sent in SAML Assertion.
attribute8expr	<String>	Read-write	Expression that will be evaluated to obtain attribute8's value to be sent in Assertion. Maximum length = 128
attribute8friendlyname	<String>	Read-write	User-Friendly Name of attribute8 that needs to be sent in SAML Assertion.
attribute8format	<String>	Read-write	Format of Attribute8 to be sent in Assertion. Possible values = URI, Basic
attribute9	<String>	Read-write	Name of attribute9 that needs to be sent in SAML Assertion.
attribute9expr	<String>	Read-write	Expression that will be evaluated to

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
			obtain attribute9's value to be sent in Assertion. Maximum length = 128
attribute9friendlyname	<String>	Read-write	User-Friendly Name of attribute9 that needs to be sent in SAML Assertion.
attribute9format	<String>	Read-write	Format of Attribute9 to be sent in Assertion. Possible values = URI, Basic
attribute10	<String>	Read-write	Name of attribute10 that needs to be sent in SAML Assertion.
attribute10expr	<String>	Read-write	Expression that will be evaluated to obtain attribute10's value to be sent in Assertion. Maximum length = 128
attribute10friendlyname	<String>	Read-write	User-Friendly Name of attribute10 that needs to be sent in SAML Assertion.
attribute10format	<String>	Read-write	Format of Attribute10 to be sent in Assertion. Possible values = URI, Basic

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
attribute11	<String>	Read-write	Name of attribute11 that needs to be sent in SAML Assertion.
attribute11expr	<String>	Read-write	Expression that will be evaluated to obtain attribute11's value to be sent in Assertion. Maximum length = 128
attribute11friendlyname	<String>	Read-write	User-Friendly Name of attribute11 that needs to be sent in SAML Assertion.
attribute11format	<String>	Read-write	Format of Attribute11 to be sent in Assertion. Possible values = URI, Basic
attribute12	<String>	Read-write	Name of attribute12 that needs to be sent in SAML Assertion.
attribute12expr	<String>	Read-write	Expression that will be evaluated to obtain attribute12's value to be sent in Assertion. Maximum length = 128
attribute12friendlyname	<String>	Read-write	User-Friendly Name of attribute12 that needs to be sent in SAML Assertion.

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
attribute12format	<String>	Read-write	Format of Attribute12 to be sent in Assertion. Possible values = URI, Basic
attribute13	<String>	Read-write	Name of attribute13 that needs to be sent in SAML Assertion.
attribute13expr	<String>	Read-write	Expression that will be evaluated to obtain attribute13's value to be sent in Assertion. Maximum length = 128
attribute13friendlyname	<String>	Read-write	User-Friendly Name of attribute13 that needs to be sent in SAML Assertion.
attribute13format	<String>	Read-write	Format of Attribute13 to be sent in Assertion. Possible values = URI, Basic
attribute14	<String>	Read-write	Name of attribute14 that needs to be sent in SAML Assertion.
attribute14expr	<String>	Read-write	Expression that will be evaluated to obtain attribute14's value to be sent in Assertion. Maximum length = 128

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
attribute14friendlyname	<String>	Read-write	User-Friendly Name of attribute14 that needs to be sent in SAML Assertion.
attribute14format	<String>	Read-write	Format of Attribute14 to be sent in Assertion. Possible values = URI, Basic
attribute15	<String>	Read-write	Name of attribute15 that needs to be sent in SAML Assertion.
attribute15expr	<String>	Read-write	Expression that will be evaluated to obtain attribute15's value to be sent in Assertion. Maximum length = 128
attribute15friendlyname	<String>	Read-write	User-Friendly Name of attribute15 that needs to be sent in SAML Assertion.
attribute15format	<String>	Read-write	Format of Attribute15 to be sent in Assertion. Possible values = URI, Basic
attribute16	<String>	Read-write	Name of attribute16 that needs to be sent in SAML Assertion.
attribute16expr	<String>	Read-write	Expression that will be evaluated to

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
			obtain attribute16's value to be sent in Assertion. Maximum length = 128
attribute16friendlyname	<String>	Read-write	User-Friendly Name of attribute16 that needs to be sent in SAML Assertion.
attribute16format	<String>	Read-write	Format of Attribute16 to be sent in Assertion. Possible values = URI, Basic
encryptassertion	<String>	Read-write	Option to encrypt assertion when Citrix ADC IDP sends one. Default value: OFF Possible values = ON, OFF
encryptionalgorithm	<String>	Read-write	Algorithm to be used to encrypt SAML assertion. Default value: AES256 Possible values = DES3, AES128, AES192, AES256
samlbinding	<String>	Read-write	This element specifies the transport mechanism of saml messages.

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
			Default value: POST Possible values = REDIRECT, POST, ARTIFACT
skewtime	<Double>	Read-write	This option specifies the number of minutes on either side of current time that the assertion would be valid. For example, if skewTime is 10, then assertion would be valid from (current time - 10) min to (current time + 10) min, ie 20min in all. Default value: 5
serviceproviderid	<String>	Read-write	Unique identifier of the Service Provider that sends SAML Request. Citrix ADC will ensure that the Issuer of the SAML Request matches this URI. Minimum length = 1
signassertion	<String>	Read-write	Option to sign portions of assertion when Citrix ADC IDP sends one. Based on the user selection, either Assertion or Response or Both or none can be signed.

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
			Default value: ASSERTION Possible values = NONE, ASSERTION, RESPONSE, BOTH
keytransportalg	<String>	Read-write	Key transport algorithm to be used in encryption of SAML assertion. Default value: RSA_OAEP Possible values = RSA-V1_5, RSA_OAEP
splogouturl	<String>	Read-write	Endpoint on the ServiceProvider (SP) to which logout messages are to be sent. Minimum length = 1
logoutbinding	<String>	Read-write	This element specifies the transport mechanism of saml logout messages. Default value: POST Possible values = REDIRECT, POST
defaultauthentication group	<String>	Read-write	This is the default group that is chosen when the

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
			authentication succeeds in addition to extracted groups. This group can be used in policies to select specific authentication policy during logon.
metadataurl	<String>	Read-write	This URL is used for obtaining samlidp metadata.
metadatarefreshinterval	<Double>	Read-write	Interval in minute for fetching metadata from specified metadata URL. Default value: 3600
signatureservice	<String>	Read-write	Name of the service in cloud used to sign the data. Minimum length = 1
samlsigningcertversion	<String>	Read-write	version of the certificate in signature service used to sign the SAMLResposne that is sent to Relying Party or Service Provider after successful authentication. Minimum length = 1
samlspcertversion	<String>	Read-write	version of the certificate in signature service used to verify the signature of the

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
			incoming AuthnRequest from a Relying Party or Service Provider. Minimum length = 1
metadataimportstatus	<String>	Read-only	Describes metadata import status. Default value: INIT Possible values = INIT, SUCCESS, FAIL
__count	<Double>	Read-only	count parameter

Operations

(click to see Properties)

ADD| DELETE| UPDATE| UNSET| GET (ALL)| GET| COUNT

Some options that you can use for each operations:

- **Getting warnings in response:**NITRO allows you to get warnings in an operation by specifying the "warning" query parameter as "yes". For example, to get warnings while connecting to the Citrix ADC appliance, the URL is as follows:

  http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/login?warning=yes

  If any, the warnings are displayed in the response payload with the HTTP code "209 X-NITRO-WARNING".

- **Authenticated access for individual NITRO operations:**NITRO allows you to logon to the Citrix ADC appliance to perform individual operations. You can use this option instead of creating a NITRO session (using the login object) and then using that session to perform all operations,

  To do this, you must specify the username and password in the request header of the NITRO request as follows:

  X-NITRO-USER:<username>

X-NITRO-PASS:<password>

Note: In such cases, make sure that the request header DOES not include the following:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue>

`*Note:`*

Mandatory parameters are marked in redand placeholder content is marked in <green>.

add

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/authenticationsamlidpprofile

HTTP Method:POST

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Content-Type:application/json

Request Payload:

```
{"authenticationsamlidpprofile":{

<b>"name":<String_value>,

</b>"samlspcertname":<String_value>,

"samlidpcertname":<String_value>,

"assertionconsumerserviceurl":<String_value>,

"sendpassword":<String_value>,

"samlissuername":<String_value>,

"rejectunsignedrequests":<String_value>,

"signaturealg":<String_value>,

"digestmethod":<String_value>,

"audience":<String_value>,
```

```
"nameidformat":<String_value>,

"nameidexpr":<String_value>,

"attribute1":<String_value>,

"attribute1expr":<String_value>,

"attribute1friendlyname":<String_value>,

"attribute1format":<String_value>,

"attribute2":<String_value>,

"attribute2expr":<String_value>,

"attribute2friendlyname":<String_value>,

"attribute2format":<String_value>,

"attribute3":<String_value>,

"attribute3expr":<String_value>,

"attribute3friendlyname":<String_value>,

"attribute3format":<String_value>,

"attribute4":<String_value>,

"attribute4expr":<String_value>,

"attribute4friendlyname":<String_value>,

"attribute4format":<String_value>,

"attribute5":<String_value>,

"attribute5expr":<String_value>,

"attribute5friendlyname":<String_value>,

"attribute5format":<String_value>,
```

```
"attribute6":<String_value>,

"attribute6expr":<String_value>,

"attribute6friendlyname":<String_value>,

"attribute6format":<String_value>,

"attribute7":<String_value>,

"attribute7expr":<String_value>,

"attribute7friendlyname":<String_value>,

"attribute7format":<String_value>,

"attribute8":<String_value>,

"attribute8expr":<String_value>,

"attribute8friendlyname":<String_value>,

"attribute8format":<String_value>,

"attribute9":<String_value>,

"attribute9expr":<String_value>,

"attribute9friendlyname":<String_value>,

"attribute9format":<String_value>,

"attribute10":<String_value>,

"attribute10expr":<String_value>,

"attribute10friendlyname":<String_value>,

"attribute10format":<String_value>,

"attribute11":<String_value>,

"attribute11expr":<String_value>,
```

```
"attribute11friendlyname":<String_value>,

"attribute11format":<String_value>,

"attribute12":<String_value>,

"attribute12expr":<String_value>,

"attribute12friendlyname":<String_value>,

"attribute12format":<String_value>,

"attribute13":<String_value>,

"attribute13expr":<String_value>,

"attribute13friendlyname":<String_value>,

"attribute13format":<String_value>,

"attribute14":<String_value>,

"attribute14expr":<String_value>,

"attribute14friendlyname":<String_value>,

"attribute14format":<String_value>,

"attribute15":<String_value>,

"attribute15expr":<String_value>,

"attribute15friendlyname":<String_value>,

"attribute15format":<String_value>,

"attribute16":<String_value>,

"attribute16expr":<String_value>,

"attribute16friendlyname":<String_value>,

"attribute16format":<String_value>,
```

```
"encryptassertion":<String_value>,

"encryptionalgorithm":<String_value>,

"samlbinding":<String_value>,

"skewtime":<Double_value>,

"serviceproviderid":<String_value>,

"signassertion":<String_value>,

"keytransportalg":<String_value>,

"splogouturl":<String_value>,

"logoutbinding":<String_value>,

"defaultauthenticationgroup":<String_value>,

"metadataurl":<String_value>,

"metadatarefreshinterval":<Double_value>,

"signatureservice":<String_value>,

"samlsigningcertversion":<String_value>,

"samlspcertversion":<String_value>

}}
```

Response:

HTTP Status Code on Success: 201 Created HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error

delete

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/authenticationsamlidpprofile/name_value<String>

HTTP Method:DELETE

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue>

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error

update

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/authenticationsamlidpprofile

HTTP Method:PUT

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Content-Type:application/json

Request Payload:

```
{"authenticationsamlidpprofile":{

<b>"name":<String_value>,

</b>"samlspcertname":<String_value>,

"samlidpcertname":<String_value>,

"assertionconsumerserviceurl":<String_value>,

"sendpassword":<String_value>,

"samlissuername":<String_value>,

"rejectunsignedrequests":<String_value>,

"signaturealg":<String_value>,

"digestmethod":<String_value>,
```

```
"audience":<String_value>,

"nameidformat":<String_value>,

"nameidexpr":<String_value>,

"attribute1":<String_value>,

"attribute1expr":<String_value>,

"attribute1friendlyname":<String_value>,

"attribute1format":<String_value>,

"attribute2":<String_value>,

"attribute2expr":<String_value>,

"attribute2friendlyname":<String_value>,

"attribute2format":<String_value>,

"attribute3":<String_value>,

"attribute3expr":<String_value>,

"attribute3friendlyname":<String_value>,

"attribute3format":<String_value>,

"attribute4":<String_value>,

"attribute4expr":<String_value>,

"attribute4friendlyname":<String_value>,

"attribute4format":<String_value>,

"attribute5":<String_value>,

"attribute5expr":<String_value>,

"attribute5friendlyname":<String_value>,
```

```
"attribute5format":<String_value>,

"attribute6":<String_value>,

"attribute6expr":<String_value>,

"attribute6friendlyname":<String_value>,

"attribute6format":<String_value>,

"attribute7":<String_value>,

"attribute7expr":<String_value>,

"attribute7friendlyname":<String_value>,

"attribute7format":<String_value>,

"attribute8":<String_value>,

"attribute8expr":<String_value>,

"attribute8friendlyname":<String_value>,

"attribute8format":<String_value>,

"attribute9":<String_value>,

"attribute9expr":<String_value>,

"attribute9friendlyname":<String_value>,

"attribute9format":<String_value>,

"attribute10":<String_value>,

"attribute10expr":<String_value>,

"attribute10friendlyname":<String_value>,

"attribute10format":<String_value>,

"attribute11":<String_value>,
```

```
"attribute11expr":<String_value>,

"attribute11friendlyname":<String_value>,

"attribute11format":<String_value>,

"attribute12":<String_value>,

"attribute12expr":<String_value>,

"attribute12friendlyname":<String_value>,

"attribute12format":<String_value>,

"attribute13":<String_value>,

"attribute13expr":<String_value>,

"attribute13friendlyname":<String_value>,

"attribute13format":<String_value>,

"attribute14":<String_value>,

"attribute14expr":<String_value>,

"attribute14friendlyname":<String_value>,

"attribute14format":<String_value>,

"attribute15":<String_value>,

"attribute15expr":<String_value>,

"attribute15friendlyname":<String_value>,

"attribute15format":<String_value>,

"attribute16":<String_value>,

"attribute16expr":<String_value>,

"attribute16friendlyname":<String_value>,
```

```
"attribute16format":<String_value>,

"encryptassertion":<String_value>,

"encryptionalgorithm":<String_value>,

"samlbinding":<String_value>,

"skewtime":<Double_value>,

"serviceproviderid":<String_value>,

"signassertion":<String_value>,

"keytransportalg":<String_value>,

"splogouturl":<String_value>,

"logoutbinding":<String_value>,

"defaultauthenticationgroup":<String_value>,

"metadataurl":<String_value>,

"metadatarefreshinterval":<Double_value>,

"signatureservice":<String_value>,

"samlsigningcertversion":<String_value>,

"samlspcertversion":<String_value>

}}
```

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error

unset

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/authenticationsamlidpprofile?**action=unset**

HTTP Method:POST

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Content-Type:application/json

Request Payload:

```
{"authenticationsamlidpprofile":{

<b>"name":<String_value>,

</b>"samlspcertname":true,

"samlidpcertname":true,

"assertionconsumerserviceurl":true,

"sendpassword":true,

"samlissuername":true,

"rejectunsignedrequests":true,

"signaturealg":true,

"digestmethod":true,

"audience":true,

"nameidformat":true,

"nameidexpr":true,

"attribute1":true,

"attribute1friendlyname":true,
```

```
"attribute1format":true,
"attribute2":true,
"attribute2friendlyname":true,
"attribute2format":true,
"attribute3":true,
"attribute3friendlyname":true,
"attribute3format":true,
"attribute4":true,
"attribute4friendlyname":true,
"attribute4format":true,
"attribute5":true,
"attribute5friendlyname":true,
"attribute5format":true,
"attribute6":true,
"attribute6friendlyname":true,
"attribute6format":true,
"attribute7":true,
"attribute7friendlyname":true,
"attribute7format":true,
"attribute8":true,
"attribute8friendlyname":true,
"attribute8format":true,
```

```
"attribute9":true,
"attribute9friendlyname":true,
"attribute9format":true,
"attribute10":true,
"attribute10friendlyname":true,
"attribute10format":true,
"attribute11":true,
"attribute11friendlyname":true,
"attribute11format":true,
"attribute12":true,
"attribute12friendlyname":true,
"attribute12format":true,
"attribute13":true,
"attribute13friendlyname":true,
"attribute13format":true,
"attribute14":true,
"attribute14friendlyname":true,
"attribute14format":true,
"attribute15":true,
"attribute15friendlyname":true,
"attribute15format":true,
"attribute16":true,
```

```
"attribute16friendlyname":true,

"attribute16format":true,

"encryptassertion":true,

"encryptionalgorithm":true,

"samlbinding":true,

"skewtime":true,

"serviceproviderid":true,

"signassertion":true,

"keytransportalg":true,

"splogouturl":true,

"logoutbinding":true,

"defaultauthenticationgroup":true,

"metadataurl":true,

"metadatarefreshinterval":true,

"signatureservice":true,

"samlsigningcertversion":true,

"samlspcertversion":true

}}
```


Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error

get (all)

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/authenticationsamlidpprofile

Query-parameters:

attrs

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/authenticationsamlidpprofile?**attrs=property-name1,property-name2**

Use this query parameter to specify the resource details that you want to retrieve.

filter

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/authenticationsamlidpprofile?**filter=property-name1:property-val1,property-name2:property-val2**

Use this query-parameter to get the filtered set of authenticationsamlidpprofile resources configured on Citrix ADC. Filtering can be done on any of the properties of the resource.

view

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/authenticationsamlidpprofile?**view=summary**

Note: By default, the retrieved results are displayed in detail view (?view=detail).

pagination

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/authenticationsamlidpprofile?**pagesize=#no;pageno=#no**

Use this query-parameter to get the authenticationsamlidpprofile resources in chunks.

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "authenticationsamlidpprofile": [ {

"name":<String_value>,

"samlspcertname":<String_value>,

"samlidpcertname":<String_value>,

"assertionconsumerserviceurl":<String_value>,

"sendpassword":<String_value>,

"samlissuername":<String_value>,

"rejectunsignedrequests":<String_value>,

"signaturealg":<String_value>,

"digestmethod":<String_value>,

"audience":<String_value>,

"nameidformat":<String_value>,

"nameidexpr":<String_value>,

"attribute1":<String_value>,

"attribute2":<String_value>,

"attribute3":<String_value>,

"attribute4":<String_value>,

"attribute5":<String_value>,

"attribute6":<String_value>,
```

```
"attribute7":<String_value>,

"attribute8":<String_value>,

"attribute9":<String_value>,

"attribute10":<String_value>,

"attribute11":<String_value>,

"attribute12":<String_value>,

"attribute13":<String_value>,

"attribute14":<String_value>,

"attribute15":<String_value>,

"attribute16":<String_value>,

"attribute1friendlyname":<String_value>,

"attribute2friendlyname":<String_value>,

"attribute3friendlyname":<String_value>,

"attribute4friendlyname":<String_value>,

"attribute5friendlyname":<String_value>,

"attribute6friendlyname":<String_value>,

"attribute7friendlyname":<String_value>,

"attribute8friendlyname":<String_value>,

"attribute9friendlyname":<String_value>,

"attribute10friendlyname":<String_value>,

"attribute11friendlyname":<String_value>,

"attribute12friendlyname":<String_value>,
```

```
"attribute13friendlyname":<String_value>,

"attribute14friendlyname":<String_value>,

"attribute15friendlyname":<String_value>,

"attribute16friendlyname":<String_value>,

"attribute1format":<String_value>,

"attribute2format":<String_value>,

"attribute3format":<String_value>,

"attribute4format":<String_value>,

"attribute5format":<String_value>,

"attribute6format":<String_value>,

"attribute7format":<String_value>,

"attribute8format":<String_value>,

"attribute9format":<String_value>,

"attribute10format":<String_value>,

"attribute11format":<String_value>,

"attribute12format":<String_value>,

"attribute13format":<String_value>,

"attribute14format":<String_value>,

"attribute15format":<String_value>,

"attribute16format":<String_value>,

"attribute1expr":<String_value>,

"attribute2expr":<String_value>,
```

```
"attribute3expr":<String_value>,

"attribute4expr":<String_value>,

"attribute5expr":<String_value>,

"attribute6expr":<String_value>,

"attribute7expr":<String_value>,

"attribute8expr":<String_value>,

"attribute9expr":<String_value>,

"attribute10expr":<String_value>,

"attribute11expr":<String_value>,

"attribute12expr":<String_value>,

"attribute13expr":<String_value>,

"attribute14expr":<String_value>,

"attribute15expr":<String_value>,

"attribute16expr":<String_value>,

"encryptassertion":<String_value>,

"encryptionalgorithm":<String_value>,

"samlbinding":<String_value>,

"skewtime":<Double_value>,

"serviceproviderid":<String_value>,

"signassertion":<String_value>,

"keytransportalg":<String_value>,

"splogouturl":<String_value>,
```

```
"logoutbinding":<String_value>,

"defaultauthenticationgroup":<String_value>,

"metadataurl":<String_value>,

"metadatarefreshinterval":<Double_value>,

"metadataimportstatus":<String_value>,

"signatureservice":<String_value>,

"samlsigningcertversion":<String_value>,

"samlspcertversion":<String_value>

}]}
```

get

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/authenticationsamlidpprofile/name_value<String>

Query-parameters:

attrs

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/authenticationsamlidpprofile/name_value<String>?**attrs=property-name1,property-name2**

Use this query parameter to specify the resource details that you want to retrieve.

view

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/authenticationsamlidpprofile/name_value<String>?**view=summary**

Note: By default, the retrieved results are displayed in detail view (?view=detail).

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "authenticationsamlidpprofile": [ {

"name":<String_value>,

"samlspcertname":<String_value>,

"samlidpcertname":<String_value>,

"assertionconsumerserviceurl":<String_value>,

"sendpassword":<String_value>,

"samlissuername":<String_value>,

"rejectunsignedrequests":<String_value>,

"signaturealg":<String_value>,

"digestmethod":<String_value>,

"audience":<String_value>,

"nameidformat":<String_value>,

"nameidexpr":<String_value>,

"attribute1":<String_value>,

"attribute2":<String_value>,
```

```
"attribute3":<String_value>,

"attribute4":<String_value>,

"attribute5":<String_value>,

"attribute6":<String_value>,

"attribute7":<String_value>,

"attribute8":<String_value>,

"attribute9":<String_value>,

"attribute10":<String_value>,

"attribute11":<String_value>,

"attribute12":<String_value>,

"attribute13":<String_value>,

"attribute14":<String_value>,

"attribute15":<String_value>,

"attribute16":<String_value>,

"attribute1friendlyname":<String_value>,

"attribute2friendlyname":<String_value>,

"attribute3friendlyname":<String_value>,

"attribute4friendlyname":<String_value>,

"attribute5friendlyname":<String_value>,

"attribute6friendlyname":<String_value>,

"attribute7friendlyname":<String_value>,

"attribute8friendlyname":<String_value>,
```

```
"attribute9friendlyname":<String_value>,

"attribute10friendlyname":<String_value>,

"attribute11friendlyname":<String_value>,

"attribute12friendlyname":<String_value>,

"attribute13friendlyname":<String_value>,

"attribute14friendlyname":<String_value>,

"attribute15friendlyname":<String_value>,

"attribute16friendlyname":<String_value>,

"attribute1format":<String_value>,

"attribute2format":<String_value>,

"attribute3format":<String_value>,

"attribute4format":<String_value>,

"attribute5format":<String_value>,

"attribute6format":<String_value>,

"attribute7format":<String_value>,

"attribute8format":<String_value>,

"attribute9format":<String_value>,

"attribute10format":<String_value>,

"attribute11format":<String_value>,

"attribute12format":<String_value>,

"attribute13format":<String_value>,

"attribute14format":<String_value>,
```

```
"attribute15format":<String_value>,

"attribute16format":<String_value>,

"attribute1expr":<String_value>,

"attribute2expr":<String_value>,

"attribute3expr":<String_value>,

"attribute4expr":<String_value>,

"attribute5expr":<String_value>,

"attribute6expr":<String_value>,

"attribute7expr":<String_value>,

"attribute8expr":<String_value>,

"attribute9expr":<String_value>,

"attribute10expr":<String_value>,

"attribute11expr":<String_value>,

"attribute12expr":<String_value>,

"attribute13expr":<String_value>,

"attribute14expr":<String_value>,

"attribute15expr":<String_value>,

"attribute16expr":<String_value>,

"encryptassertion":<String_value>,

"encryptionalgorithm":<String_value>,

"samlbinding":<String_value>,

"skewtime":<Double_value>,
```

```
"serviceproviderid":<String_value>,

"signassertion":<String_value>,

"keytransportalg":<String_value>,

"splogouturl":<String_value>,

"logoutbinding":<String_value>,

"defaultauthenticationgroup":<String_value>,

"metadataurl":<String_value>,

"metadatarefreshinterval":<Double_value>,

"metadataimportstatus":<String_value>,

"signatureservice":<String_value>,

"samlsigningcertversion":<String_value>,

"samlspcertversion":<String_value>

}]}
```

count

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/authenticationsamlidpprofile?**count=yes**

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "authenticationsamlidpprofile": [ { "__count": "#no"} ] }
```

authenticationsamlpolicy

Configuration for AAA Saml policy resource.

Properties

(click to see Operations)

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
name	<String>	Read-write	Name for the SAML policy. Must begin with a letter, number, or the underscore character (_), and must contain only letters, numbers, and the hyphen (-), period (.) pound (#), space ( ), at (@), equals (=), colon (:), and underscore characters. Cannot be changed after SAML policy is created. The following requirement applies

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
			only to the Citrix ADC CLI: If the name includes one or more spaces, enclose the name in double or single quotation marks (for example, "my authentication policy" or 'my authentication policy'). Minimum length = 1
rule	<String>	Read-write	Name of the Citrix ADC named rule, or an expression, that the policy uses to determine whether to attempt to authenticate the user with the SAML server. Minimum length = 1
reqaction	<String>	Read-write	Name of the SAML authentication action to be performed if the policy matches. Minimum length = 1
__count	<Double>	Read-only	count parameter

Operations

(click to see Properties)

ADD| DELETE| UPDATE| UNSET| GET (ALL)| GET| COUNT

Some options that you can use for each operations:

- **Getting warnings in response:**NITRO allows you to get warnings in an operation by specifying the "warning" query parameter as "yes". For example, to get warnings while connecting to the Citrix ADC appliance, the URL is as follows:

  http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/login?warning=yes

  If any, the warnings are displayed in the response payload with the HTTP code "209 X-NITRO-WARNING".

- **Authenticated access for individual NITRO operations:**NITRO allows you to logon to the Citrix ADC appliance to perform individual operations. You can use this option instead of creating a NITRO session (using the login object) and then using that session to perform all operations,

  To do this, you must specify the username and password in the request header of the NITRO request as follows:

  X-NITRO-USER:<username>

  X-NITRO-PASS:<password>

  Note: In such cases, make sure that the request header DOES not include the following:

  Cookie:NITRO_AUTH_TOKEN=<tokenvalue>

`*Note:`*

Mandatory parameters are marked in redand placeholder content is marked in <green>.

add

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/authenticationsamlpolicy

HTTP Method:POST

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Content-Type:application/json

Request Payload:

```
{"authenticationsamlpolicy":{

<b>"name":<String_value>,

</b><b>"rule":<String_value>,

</b><b>"reqaction":<String_value>

</b>}}
```

Response:

HTTP Status Code on Success: 201 Created HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error

delete

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/authenticationsamlpolicy/name_value<String>

HTTP Method:DELETE

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue>

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error

update

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/authenticationsamlpolicy

HTTP Method:PUT

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Content-Type:application/json

Request Payload:

```
{"authenticationsamlpolicy":{

<b>"name":<String_value>,

</b>"rule":<String_value>,

"reqaction":<String_value>

}}
```

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error

unset

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/authenticationsamlpolicy?**action=unset**

HTTP Method:POST

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Content-Type:application/json

Request Payload:

```
{"authenticationsamlpolicy":{

<b>"name":<String_value>,

</b>"rule":true,

"reqaction":true

}}
```

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error

get (all)

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/authenticationsamlpolicy

Query-parameters:

attrs

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/authenticationsamlpolicy?**attrs=property-name1,property-name2**

Use this query parameter to specify the resource details that you want to retrieve.

filter

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/authenticationsamlpolicy?**filter=property-name1:property-val1,property-name2:property-val2**

Use this query-parameter to get the filtered set of authenticationsamlpolicy resources configured on Citrix ADC. Filtering can be done on any of the properties of the resource.

view

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/authenticationsamlpolicy?**view=summary**

Note: By default, the retrieved results are displayed in detail view (?view=detail).

pagination

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/authenticationsamlpolicy?**pagesize=#no;pageno=#no**

Use this query-parameter to get the authenticationsamlpolicy resources in chunks.

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "authenticationsamlpolicy": [ {

"name":<String_value>,

"rule":<String_value>,

"reqaction":<String_value>

}]}
```

get

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/authenticationsamlpolicy/name_value<String>

Query-parameters:

attrs

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/authenticationsamlpolicy/name_value<String>?**attrs=property-name1,property-name2**

Use this query parameter to specify the resource details that you want to retrieve.

view

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/authenticationsamlpolicy/name_value<String>?**view=summary**

Note: By default, the retrieved results are displayed in detail view (?view=detail).

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "authenticationsamlpolicy": [ {

"name":<String_value>,

"rule":<String_value>,

"reqaction":<String_value>

}]}
```

count

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/authenticationsamlpolicy?**count=yes**

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "authenticationsamlpolicy": [ { "__count": "#no"} ] }
```

authenticationsamlpolicy_authenticationvserver_binding

Binding object showing the authenticationvserver that can be bound to authenticationsamlpolicy.

Properties

(click to see Operations)

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
boundto	<String>	Read-write	The entity name to which policy is bound.
name	<String>	Read-write	Name of the SAML policy. Minimum length = 1
priority	<Double>	Read-only	.
activepolicy	<Double>	Read-only	.
__count	<Double>	Read-only	count parameter

Operations

(click to see Properties)

GET| GET (ALL)| COUNT

Some options that you can use for each operations:

- **Getting warnings in response:**NITRO allows you to get warnings in an operation by specifying the "warning" query parameter as "yes". For example, to get warnings while connecting to the Citrix ADC appliance, the URL is as follows:

  http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/login?warning=yes

  If any, the warnings are displayed in the response payload with the HTTP code "209 X-NITRO-WARNING".

- **Authenticated access for individual NITRO operations:**NITRO allows you to logon to the Citrix ADC appliance to perform individual operations. You can use this option instead of creating a NITRO session (using the login object) and then using that session to perform all operations,

  To do this, you must specify the username and password in the request header of the NITRO request as follows:

  X-NITRO-USER:<username>

  X-NITRO-PASS:<password>

  Note: In such cases, make sure that the request header DOES not include the following:

  Cookie:NITRO_AUTH_TOKEN=<tokenvalue>

`*Note:`*

Mandatory parameters are marked in redand placeholder content is marked in <green>.

get

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/authenticationsamlpolicy_authenticationvserver_binding/name_value<String>

Query-parameters:

filter

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/authenticationsamlpolicy_authenticationvserver_binding/name_value<String>?**filter=property-name1:property-value1,property-name2:property-value2**

Use this query-parameter to get the filtered set of authenticationsamlpolicy_authenticationvserver_binding resources configured on Citrix ADC. Filtering can be done on any of the properties of the resource.

pagination

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/authenticationsamlpolicy_authenticationvserver_binding/name_value<String>?**pagesize=#no;pageno=#no**

Use this query-parameter to get the authenticationsamlpolicy_authenticationvserver_binding resources in chunks.

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "authenticationsamlpolicy_authenticationvserver_binding": [ {

"boundto":<String_value>,

"name":<String_value>,

"priority":<Double_value>,

"activepolicy":<Double_value>

}]}
```

get (all)

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/authenticationsamlpolicy_authenticationvserver_binding

Query-parameters:

bulkbindings

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/authenticationsamlpolicy_authenticationvserver_binding?**bulkbindings=yes**

NITRO allows you to fetch bindings in bulk.

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "authenticationsamlpolicy_authenticationvserver_binding": [ {

"boundto":<String_value>,

"name":<String_value>,

"priority":<Double_value>,

"activepolicy":<Double_value>

}]}
```

count

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/authenticationsamlpolicy_authenticationvserver_binding/name_value<String>?**count=yes**

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{"authenticationsamlpolicy_authenticationvserver_binding": [ { "__count":
"#no"} ] }
```

authenticationsamlpolicy_binding

Binding object which returns the resources bound to authenticationsamlpolicy.

Properties

(click to see Operations)

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
name	<String>	Read-write	Name of the SAML policy. Minimum length = 1
authenticationsamlpolicy_vpnglobal_binding	<authenticationsamlpolicy_vpnglobal_binding[]>	Read-only	vpnglobal that can be bound to authenticationsamlpolicy.

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
authenticationsamlpolicy_vpnvserver_binding	<authenticationsamlpolicy_vpnvserver_binding[]>	Read-only	vpnvserver that can be bound to authenticationsamlpolicy.
authenticationsamlpolicy_authenticationvserver_binding	<authenticationsamlpolicy_authenticationvserver_binding[]>	Read-only	authenticationvserver that can be bound to authenticationsamlpolicy.

Operations

(click to see Properties)

GET| GET (ALL)

Some options that you can use for each operations:

- **Getting warnings in response:**NITRO allows you to get warnings in an operation by specifying the "warning" query parameter as "yes". For example, to get warnings while connecting to the Citrix ADC appliance, the URL is as follows:

  http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/login?warning=yes

  If any, the warnings are displayed in the response payload with the HTTP code "209 X-NITRO-WARNING".

- **Authenticated access for individual NITRO operations:**NITRO allows you to logon to the Citrix ADC appliance to perform individual operations. You can use this option instead of creating a NITRO session (using the login object) and then using that session to perform all operations,

  To do this, you must specify the username and password in the request header of the NITRO request as follows:

  X-NITRO-USER:<username>

  X-NITRO-PASS:<password>

  Note: In such cases, make sure that the request header DOES not include the following:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue>

`*Note:`*

Mandatory parameters are marked in redand placeholder content is marked in <green>.

get

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/authenticationsamlpolicy_binding/name_value<String>

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "authenticationsamlpolicy_binding": [ {

"name":<String_value>,

"authenticationsamlpolicy_vpnvserver_binding":<authenticationsamlpolicy_vpnvserver_binding[]_value>,

"authenticationsamlpolicy_vpnglobal_binding":<authenticationsamlpolicy_vpnglobal_binding[]_value>,

"authenticationsamlpolicy_authenticationvserver_binding":<authenticationsamlpolicy_authenticationvserver_binding[]_value>

}]}
```

get (all)

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/authenticationsamlpolicy_binding

Query-parameters:

bulkbindings

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/authenticationsamlpolicy_binding?**bulkbindings=yes**

NITRO allows you to fetch bindings in bulk.

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "authenticationsamlpolicy_binding": [ {

"name":<String_value>,

"authenticationsamlpolicy_vpnvserver_binding":<authenticationsamlpolicy_vpnvserver_binding[]_value>,

"authenticationsamlpolicy_vpnglobal_binding":<authenticationsamlpolicy_vpnglobal_binding[]_value>,

"authenticationsamlpolicy_authenticationvserver_binding":<authenticationsamlpolicy_authenticationvserver_binding[]_value>

}]}
```

authenticationsamlpolicy_vpnglobal_binding

Binding object showing the vpnglobal that can be bound to authenticationsamlpolicy.

Properties

(click to see Operations)

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
boundto	<String>	Read-write	The entity name to which policy is bound.
name	<String>	Read-write	Name of the SAML policy. Minimum length = 1
priority	<Double>	Read-only	.
activepolicy	<Double>	Read-only	.
__count	<Double>	Read-only	count parameter

Operations

(click to see Properties)

GET| GET (ALL)| COUNT

Some options that you can use for each operations:

- **Getting warnings in response:**NITRO allows you to get warnings in an operation by specifying the "warning" query parameter as "yes". For example, to get warnings while connecting to the Citrix ADC appliance, the URL is as follows:

  http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/login?warning=yes

  If any, the warnings are displayed in the response payload with the HTTP code "209 X-NITRO-WARNING".

- **Authenticated access for individual NITRO operations:**NITRO allows you to logon to the Citrix ADC appliance to perform individual operations. You can use this option instead of creating a NITRO session (using the login object) and then using that session to perform all operations,

  To do this, you must specify the username and password in the request header of the NITRO request as follows:

  X-NITRO-USER:<username>

  X-NITRO-PASS:<password>

  Note: In such cases, make sure that the request header DOES not include the following:

  Cookie:NITRO_AUTH_TOKEN=<tokenvalue>

`*Note:`*

Mandatory parameters are marked in redand placeholder content is marked in <green>.

get

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/authenticationsamlpolicy_vpnglobal_binding/name_value<String>

Query-parameters:

filter

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/authenticationsamlpolicy_vpnglobal_binding/name_value<String>?**filter=property-name1:property-value1,property-name2:property-value2**

Use this query-parameter to get the filtered set of authenticationsamlpolicy_vpnglobal_binding resources configured on Citrix ADC. Filtering can be done on any of the properties of the resource.

pagination

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/authenticationsamlpolicy_vpnglobal_binding/name_value<String>?**pagesize=#no;pageno=#no**

Use this query-parameter to get the authenticationsamlpolicy_vpnglobal_binding resources in chunks.

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "authenticationsamlpolicy_vpnglobal_binding": [ {

"boundto":<String_value>,

"name":<String_value>,

"priority":<Double_value>,

"activepolicy":<Double_value>

}]}
```

get (all)

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/authenticationsamlpolicy_vpnglobal_binding

Query-parameters:

bulkbindings

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/authenticationsamlpolicy_vpnglobal_binding?**bulkbindings=yes**

NITRO allows you to fetch bindings in bulk.

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "authenticationsamlpolicy_vpnglobal_binding": [ {

"boundto":<String_value>,

"name":<String_value>,

"priority":<Double_value>,

"activepolicy":<Double_value>

}]}
```

count

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/authenticationsamlpolicy_vpnglobal_binding/name_value<String>?**count=yes**

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{"authenticationsamlpolicy_vpnglobal_binding": [ { "__count": "#no"} ] }
```

authenticationsamlpolicy_vpnvserver_binding

Binding object showing the vpnvserver that can be bound to authenticationsamlpolicy.

Properties

(click to see Operations)

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
boundto	<String>	Read-write	The entity name to which policy is bound.
name	<String>	Read-write	Name of the SAML policy. Minimum length = 1
priority	<Double>	Read-only	.
activepolicy	<Double>	Read-only	.
__count	<Double>	Read-only	count parameter

Operations

(click to see Properties)

GET| GET (ALL)| COUNT

Some options that you can use for each operations:

- **Getting warnings in response:**NITRO allows you to get warnings in an operation by specifying the "warning" query parameter as "yes". For example, to get warnings while connecting to the Citrix ADC appliance, the URL is as follows:

  http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/login?warning=yes

  If any, the warnings are displayed in the response payload with the HTTP code "209 X-NITRO-WARNING".

- **Authenticated access for individual NITRO operations:**NITRO allows you to logon to the Citrix ADC appliance to perform individual operations. You can use this option instead of creating a NITRO session (using the login object) and then using that session to perform all operations,

  To do this, you must specify the username and password in the request header of the NITRO request as follows:

  X-NITRO-USER:<username>

  X-NITRO-PASS:<password>

  Note: In such cases, make sure that the request header DOES not include the following:

  Cookie:NITRO_AUTH_TOKEN=<tokenvalue>

`*Note:`*

Mandatory parameters are marked in redand placeholder content is marked in <green>.

get

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/authenticationsamlpolicy_vpnvserver_binding/name_value<String>

Query-parameters:

filter

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/authenticationsamlpolicy_vpnvserver_binding/name_value<String>?**filter=property-name1:property-value1,property-name2:property-value2**

Use this query-parameter to get the filtered set of authenticationsamlpolicy_vpnvserver_binding resources configured on Citrix ADC. Filtering can be done on any of the properties of the resource.

pagination

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/authenticationsamlpolicy_vpnvserver_binding/name_value<String>?**pagesize=#no;pageno=#no**

Use this query-parameter to get the authenticationsamlpolicy_vpnvserver_binding resources in chunks.

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "authenticationsamlpolicy_vpnvserver_binding": [ {

"boundto":<String_value>,

"name":<String_value>,

"priority":<Double_value>,

"activepolicy":<Double_value>

}]}
```

get (all)

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/authenticationsamlpolicy_vpnvserver_binding

Query-parameters:

bulkbindings

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/authenticationsamlpolicy_vpnvserver_binding?**bulkbindings=yes**

NITRO allows you to fetch bindings in bulk.

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "authenticationsamlpolicy_vpnvserver_binding": [ {

"boundto":<String_value>,

"name":<String_value>,

"priority":<Double_value>,

"activepolicy":<Double_value>

}]}
```

count

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/authenticationsamlpolicy_vpnvserver_binding/name_value<String>?**count=yes**

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{"authenticationsamlpolicy_vpnvserver_binding": [ { "__count": "#no"} ] }
```

authenticationstorefrontauthaction

Configuration for Storefront authentication action resource.

Properties

(click to see Operations)

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
name	<String>	Read-write	Name for the Storefront Authentication action. Must begin with a letter, number, or the underscore character (_), and must contain only letters, numbers, and the hyphen (-), period (.) pound (#), space ( ), at (@),

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
			equals (=), colon (:), and underscore characters. Cannot be changed after the profile is created. The following requirement applies only to the Citrix ADC CLI: If the name includes one or more spaces, enclose the name in double or single quotation marks (for example, "my authentication action" or 'my authentication action'). Minimum length = 1
serverurl	<String>	Read-write	URL of the Storefront server. This is the FQDN of the Storefront server. example: https://storefront.com/. Authentication endpoints are learned dynamically by Gateway.
domain	<String>	Read-write	Domain of the server that is used for authentication. If

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
			users enter name without domain, this parameter is added to username in the authentication request to server.
defaultauthentication group	<String>	Read-write	This is the default group that is chosen when the authentication succeeds in addition to extracted groups.
success	<Double>	Read-only	.
failure	<Double>	Read-only	.
__count	<Double>	Read-only	count parameter

Operations

(click to see Properties)

ADD| DELETE| UPDATE| UNSET| GET (ALL)| GET| COUNT

Some options that you can use for each operations:

- **Getting warnings in response:**NITRO allows you to get warnings in an operation by specifying the "warning" query parameter as "yes". For example, to get warnings while connecting to the Citrix ADC appliance, the URL is as follows:

  http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/login?warning=yes

  If any, the warnings are displayed in the response payload with the HTTP code "209 X-NITRO-WARNING".

- **Authenticated access for individual NITRO operations:**NITRO allows you to logon to the Citrix ADC appliance to perform individual operations. You can use this option instead of creating a NITRO session (using the login object) and then using that session to perform all operations,

To do this, you must specify the username and password in the request header of the NITRO request as follows:

X-NITRO-USER:<username>

X-NITRO-PASS:<password>

Note: In such cases, make sure that the request header DOES not include the following:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue>

`*Note:`*

Mandatory parameters are marked in redand placeholder content is marked in <green>.

add

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/authenticationstorefrontauthaction

HTTP Method:POST

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Content-Type:application/json

Request Payload:

```
{"authenticationstorefrontauthaction":{

<b>"name":<String_value>,

</b><b>"serverurl":<String_value>,

</b>"domain":<String_value>,

"defaultauthenticationgroup":<String_value>

}}
```

Response:

HTTP Status Code on Success: 201 Created HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error

delete

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/authenticationstorefrontauthaction/name_value<String>

HTTP Method:DELETE

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue>

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error

update

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/authenticationstorefrontauthaction

HTTP Method:PUT

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Content-Type:application/json

Request Payload:

```
{"authenticationstorefrontauthaction":{

<b>"name":<String_value>,

</b>"serverurl":<String_value>,

"domain":<String_value>,

"defaultauthenticationgroup":<String_value>
```

```
}}
```

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error

unset

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/authenticationstorefrontauthaction?**action=unset**

HTTP Method:POST

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Content-Type:application/json

Request Payload:

```
{"authenticationstorefrontauthaction":{

<b>"name":<String_value>,

</b>"domain":true,

"defaultauthenticationgroup":true

}}
```

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error

get (all)

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/authenticationstorefrontauthaction

Query-parameters:

attrs

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/authenticationstorefrontauthaction?**attrs=property-name1,property-name2**

Use this query parameter to specify the resource details that you want to retrieve.

filter

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/authenticationstorefrontauthaction?**filter=property-name1:property-val1,property-name2:property-val2**

Use this query-parameter to get the filtered set of authenticationstorefrontauthaction resources configured on Citrix ADC. Filtering can be done on any of the properties of the resource.

view

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/authenticationstorefrontauthaction?**view=summary**

Note: By default, the retrieved results are displayed in detail view (?view=detail).

pagination

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/authenticationstorefrontauthaction?**pagesize=#no;pageno=#no**

Use this query-parameter to get the authenticationstorefrontauthaction resources in chunks.

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "authenticationstorefrontauthaction": [ {

"name":<String_value>,

"serverurl":<String_value>,

"domain":<String_value>,

"defaultauthenticationgroup":<String_value>,

"success":<Double_value>,

"failure":<Double_value>

}]}
```

get

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/authenticationstorefrontauthaction/name_value<String>

Query-parameters:

attrs

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/authenticationstorefrontauthaction/name_value<String>?**attrs=property-name1,property-name2**

Use this query parameter to specify the resource details that you want to retrieve.

view

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/authenticationstorefrontauthaction/name_value<String>?**view=summary**

Note: By default, the retrieved results are displayed in detail view (?view=detail).

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "authenticationstorefrontauthaction": [ {

"name":<String_value>,

"serverurl":<String_value>,

"domain":<String_value>,

"defaultauthenticationgroup":<String_value>,

"success":<Double_value>,

"failure":<Double_value>

}]}
```

count

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/authenticationstorefrontauthaction?**count=yes**

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "authenticationstorefrontauthaction": [ { "__count": "#no"} ] }
```

authenticationtacacsaction

Configuration for TACACS action resource.

Properties

(click to see Operations)

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
name	<String>	Read-write	Name for the TACACS+ profile (action). Must begin with a letter, number, or the underscore character (_), and must contain only letters, numbers, and the hyphen (-), period (.) pound (#), space ( ), at (@), equals (=), colon (:), and underscore

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
			characters. Cannot be changed after TACACS profile is created. The following requirement applies only to the Citrix ADC CLI: If the name includes one or more spaces, enclose the name in double or single quotation marks (for example, "my authentication action" or 'y authentication action'). Minimum length = 1
serverip	<String>	Read-write	IP address assigned to the TACACS+ server. Minimum length = 1
serverport	<Integer>	Read-write	Port number on which the TACACS+ server listens for connections. Default value: 49 Minimum value = 1
authtimeout	<Double>	Read-write	Number of seconds the Citrix ADC waits for a response from

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
			the TACACS+ server. Default value: 3 Minimum value = 1
tacacssecret	<String>	Read-write	Key shared between the TACACS+ server and the Citrix ADC. Required for allowing the Citrix ADC to communicate with the TACACS+ server. Minimum length = 1
authorization	<String>	Read-write	Use streaming authorization on the TACACS+ server. Possible values = ON, OFF
accounting	<String>	Read-write	Whether the TACACS+ server is currently accepting accounting messages. Possible values = ON, OFF
auditfailedcmds	<String>	Read-write	The state of the TACACS+ server that will receive accounting messages. Possible values = ON, OFF

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
groupattrname	<String>	Read-write	TACACS+ group attribute name. Used for group extraction on the TACACS+ server.
defaultauthentication group	<String>	Read-write	This is the default group that is chosen when the authentication succeeds in addition to extracted groups.
attribute1	<String>	Read-write	Name of the custom attribute to be extracted from server and stored at index '1' (where '1' changes for each attribute).
attribute2	<String>	Read-write	Name of the custom attribute to be extracted from server and stored at index '2' (where '2' changes for each attribute).
attribute3	<String>	Read-write	Name of the custom attribute to be extracted from server and stored at index '3' (where '3' changes for each attribute).
attribute4	<String>	Read-write	Name of the custom attribute to be extracted from server and stored at index

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
			'4' (where '4' changes for each attribute).
attribute5	<String>	Read-write	Name of the custom attribute to be extracted from server and stored at index '5' (where '5' changes for each attribute).
attribute6	<String>	Read-write	Name of the custom attribute to be extracted from server and stored at index '6' (where '6' changes for each attribute).
attribute7	<String>	Read-write	Name of the custom attribute to be extracted from server and stored at index '7' (where '7' changes for each attribute).
attribute8	<String>	Read-write	Name of the custom attribute to be extracted from server and stored at index '8' (where '8' changes for each attribute).
attribute9	<String>	Read-write	Name of the custom attribute to be extracted from server and stored at index '9' (where '9'

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
			changes for each attribute).
attribute10	<String>	Read-write	Name of the custom attribute to be extracted from server and stored at index '10' (where '10' changes for each attribute).
attribute11	<String>	Read-write	Name of the custom attribute to be extracted from server and stored at index '11' (where '11' changes for each attribute).
attribute12	<String>	Read-write	Name of the custom attribute to be extracted from server and stored at index '12' (where '12' changes for each attribute).
attribute13	<String>	Read-write	Name of the custom attribute to be extracted from server and stored at index '13' (where '13' changes for each attribute).
attribute14	<String>	Read-write	Name of the custom attribute to be extracted from server and stored at index '14' (where '14'

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
			changes for each attribute).
attribute15	<String>	Read-write	Name of the custom attribute to be extracted from server and stored at index '15' (where '15' changes for each attribute).
attribute16	<String>	Read-write	Name of the custom attribute to be extracted from server and stored at index '16' (where '16' changes for each attribute).
attributes	<String>	Read-write	List of attribute names separated by ',' which needs to be fetched from tacacs server. Note that preceeding and trailing spaces will be removed. Attribute name can be 127 bytes and total length of this string should not cross 2047 bytes. These attributes have multi-value support separated by ',' and stored as

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
			key-value pair in AAA session.
success	<Double>	Read-only	.
failure	<Double>	Read-only	.
__count	<Double>	Read-only	count parameter

Operations

(click to see Properties)

ADD| DELETE| UPDATE| UNSET| GET (ALL)| GET| COUNT

Some options that you can use for each operations:

- **Getting warnings in response:**NITRO allows you to get warnings in an operation by specifying the "warning" query parameter as "yes". For example, to get warnings while connecting to the Citrix ADC appliance, the URL is as follows:

  http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/login?warning=yes

  If any, the warnings are displayed in the response payload with the HTTP code "209 X-NITRO-WARNING".

- **Authenticated access for individual NITRO operations:**NITRO allows you to logon to the Citrix ADC appliance to perform individual operations. You can use this option instead of creating a NITRO session (using the login object) and then using that session to perform all operations,

  To do this, you must specify the username and password in the request header of the NITRO request as follows:

  X-NITRO-USER:<username>

  X-NITRO-PASS:<password>

  Note: In such cases, make sure that the request header DOES not include the following:

  Cookie:NITRO_AUTH_TOKEN=<tokenvalue>

`*Note:`*

Mandatory parameters are marked in redand placeholder content is marked in <green>.

add

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/authenticationtacacsaction

HTTP Method:POST

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Content-Type:application/json

Request Payload:

```
{"authenticationtacacsaction":{

<b>"name":<String_value>,

</b>"serverip":<String_value>,

"serverport":<Integer_value>,

"authtimeout":<Double_value>,

"tacacssecret":<String_value>,

"authorization":<String_value>,

"accounting":<String_value>,

"auditfailedcmds":<String_value>,

"groupattrname":<String_value>,

"defaultauthenticationgroup":<String_value>,

"attribute1":<String_value>,

"attribute2":<String_value>,

"attribute3":<String_value>,

"attribute4":<String_value>,
```

```
"attribute5":<String_value>,

"attribute6":<String_value>,

"attribute7":<String_value>,

"attribute8":<String_value>,

"attribute9":<String_value>,

"attribute10":<String_value>,

"attribute11":<String_value>,

"attribute12":<String_value>,

"attribute13":<String_value>,

"attribute14":<String_value>,

"attribute15":<String_value>,

"attribute16":<String_value>,

"attributes":<String_value>

}}
```

Response:

HTTP Status Code on Success: 201 Created HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error

delete

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/authenticationtacacsaction/name_value<String>

HTTP Method:DELETE

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue>

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error

update

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/authenticationtacacsaction

HTTP Method:PUT

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Content-Type:application/json

Request Payload:

```
{"authenticationtacacsaction":{

<b>"name":<String_value>,

</b>"serverip":<String_value>,

"serverport":<Integer_value>,

"authtimeout":<Double_value>,

"tacacssecret":<String_value>,

"authorization":<String_value>,

"accounting":<String_value>,

"auditfailedcmds":<String_value>,

"groupattrname":<String_value>,

"defaultauthenticationgroup":<String_value>,

"attribute1":<String_value>,
```

```
"attribute2":<String_value>,

"attribute3":<String_value>,

"attribute4":<String_value>,

"attribute5":<String_value>,

"attribute6":<String_value>,

"attribute7":<String_value>,

"attribute8":<String_value>,

"attribute9":<String_value>,

"attribute10":<String_value>,

"attribute11":<String_value>,

"attribute12":<String_value>,

"attribute13":<String_value>,

"attribute14":<String_value>,

"attribute15":<String_value>,

"attribute16":<String_value>,

"attributes":<String_value>

}}
```

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error

unset

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/authenticationtacacsaction?**action=unset**

HTTP Method:POST

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Content-Type:application/json

Request Payload:

```
{"authenticationtacacsaction":{

<b>"name":<String_value>,

</b>"serverip":true,

"serverport":true,

"authtimeout":true,

"tacacssecret":true,

"authorization":true,

"accounting":true,

"auditfailedcmds":true,

"groupattrname":true,

"defaultauthenticationgroup":true,

"attribute1":true,

"attribute2":true,

"attribute3":true,

"attribute4":true,
```

```
"attribute5":true,

"attribute6":true,

"attribute7":true,

"attribute8":true,

"attribute9":true,

"attribute10":true,

"attribute11":true,

"attribute12":true,

"attribute13":true,

"attribute14":true,

"attribute15":true,

"attribute16":true,

"attributes":true

}}
```

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error

get (all)

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/authenticationtacacsaction

Query-parameters:

attrs

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/authenticationtacacsaction?**attrs=property-name1,property-name2**

Use this query parameter to specify the resource details that you want to retrieve.

filter

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/authenticationtacacsaction?**filter=property-name1:property-val1,property-name2:property-val2**

Use this query-parameter to get the filtered set of authenticationtacacsaction resources configured on Citrix ADC. Filtering can be done on any of the properties of the resource.

view

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/authenticationtacacsaction?**view=summary**

Note: By default, the retrieved results are displayed in detail view (?view=detail).

pagination

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/authenticationtacacsaction?**pagesize=#no;pageno=#no**

Use this query-parameter to get the authenticationtacacsaction resources in chunks.

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "authenticationtacacsaction": [ {
```

```
"name":<String_value>,

"serverip":<String_value>,

"serverport":<Integer_value>,

"authtimeout":<Double_value>,

"tacacssecret":<String_value>,

"authorization":<String_value>,

"accounting":<String_value>,

"auditfailedcmds":<String_value>,

"groupattrname":<String_value>,

"success":<Double_value>,

"failure":<Double_value>,

"defaultauthenticationgroup":<String_value>,

"attribute1":<String_value>,

"attribute2":<String_value>,

"attribute3":<String_value>,

"attribute4":<String_value>,

"attribute5":<String_value>,

"attribute6":<String_value>,

"attribute7":<String_value>,

"attribute8":<String_value>,

"attribute9":<String_value>,

"attribute10":<String_value>,
```

```
"attribute11":<String_value>,

"attribute12":<String_value>,

"attribute13":<String_value>,

"attribute14":<String_value>,

"attribute15":<String_value>,

"attribute16":<String_value>,

"attributes":<String_value>

}]}
```

get

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/authenticationtacacsaction/name_value<String>

Query-parameters:

attrs

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/authenticationtacacsaction/name_value<String>?**attrs=property-name1,property-name2**

Use this query parameter to specify the resource details that you want to retrieve.

view

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/authenticationtacacsaction/name_value<String>?**view=summary**

Note: By default, the retrieved results are displayed in detail view (?view=detail).

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "authenticationtacacsaction": [ {

"name":<String_value>,

"serverip":<String_value>,

"serverport":<Integer_value>,

"authtimeout":<Double_value>,

"tacacssecret":<String_value>,

"authorization":<String_value>,

"accounting":<String_value>,

"auditfailedcmds":<String_value>,

"groupattrname":<String_value>,

"success":<Double_value>,

"failure":<Double_value>,

"defaultauthenticationgroup":<String_value>,

"attribute1":<String_value>,

"attribute2":<String_value>,

"attribute3":<String_value>,
```

```
"attribute4":<String_value>,

"attribute5":<String_value>,

"attribute6":<String_value>,

"attribute7":<String_value>,

"attribute8":<String_value>,

"attribute9":<String_value>,

"attribute10":<String_value>,

"attribute11":<String_value>,

"attribute12":<String_value>,

"attribute13":<String_value>,

"attribute14":<String_value>,

"attribute15":<String_value>,

"attribute16":<String_value>,

"attributes":<String_value>

}]}
```

count

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/authenticationtacacsaction?**count=yes**

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "authenticationtacacsaction": [ { "__count": "#no"} ] }
```

authenticationtacacspolicy

Configuration for TACACS policy resource.

Properties

(click to see Operations)

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
name	<String>	Read-write	Name for the TACACS+ policy. Must begin with a letter, number, or the underscore character (_), and must contain only letters, numbers, and the hyphen (-), period (.) pound (#), space ( ), at (@), equals (=), colon (:), and underscore characters. Cannot be changed after

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
			TACACS+ policy is created. The following requirement applies only to the Citrix ADC CLI: If the name includes one or more spaces, enclose the name in double or single quotation marks (for example, "my authentication policy" or 'my authentication policy'). Minimum length = 1
rule	<String>	Read-write	Name of the Citrix ADC named rule, or an expression, that the policy uses to determine whether to attempt to authenticate the user with the TACACS+ server. Minimum length = 1
reqaction	<String>	Read-write	Name of the TACACS+ action to perform if the policy matches. Minimum length = 1
__count	<Double>	Read-only	count parameter

Operations

(click to see Properties)

ADD| DELETE| UPDATE| UNSET| GET (ALL)| GET| COUNT

Some options that you can use for each operations:

- **Getting warnings in response:**NITRO allows you to get warnings in an operation by specifying the "warning" query parameter as "yes". For example, to get warnings while connecting to the Citrix ADC appliance, the URL is as follows:

  http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/login?warning=yes

  If any, the warnings are displayed in the response payload with the HTTP code "209 X-NITRO-WARNING".

- **Authenticated access for individual NITRO operations:**NITRO allows you to logon to the Citrix ADC appliance to perform individual operations. You can use this option instead of creating a NITRO session (using the login object) and then using that session to perform all operations,

  To do this, you must specify the username and password in the request header of the NITRO request as follows:

  X-NITRO-USER:<username>

  X-NITRO-PASS:<password>

  Note: In such cases, make sure that the request header DOES not include the following:

  Cookie:NITRO_AUTH_TOKEN=<tokenvalue>

`*Note:`*

Mandatory parameters are marked in redand placeholder content is marked in <green>.

add

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/authenticationtacacspolicy

HTTP Method:POST

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Content-Type:application/json

Request Payload:

```
{"authenticationtacacspolicy":{

<b>"name":<String_value>,

</b><b>"rule":<String_value>,

</b>"reqaction":<String_value>

}}
```

Response:

HTTP Status Code on Success: 201 Created HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error

delete

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/authenticationtacacspolicy/name_value<String>

HTTP Method:DELETE

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue>

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error

update

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/authenticationtacacspolicy

HTTP Method:PUT

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Content-Type:application/json

Request Payload:

```
{"authenticationtacacspolicy":{

<b>"name":<String_value>,

</b>"rule":<String_value>,

"reqaction":<String_value>

}}
```

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error

unset

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/authenticationtacacspolicy?**action=unset**

HTTP Method:POST

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Content-Type:application/json

Request Payload:

```
{"authenticationtacacspolicy":{

<b>"name":<String_value>,

</b>"rule":true,
```

```
"reqaction":true

}}
```

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error

get (all)

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/authenticationtacacspolicy

Query-parameters:

attrs

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/authenticationtacacspolicy?**attrs=property-name1,property-name2**

Use this query parameter to specify the resource details that you want to retrieve.

filter

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/authenticationtacacspolicy?**filter=property-name1:property-val1,property-name2:property-val2**

Use this query-parameter to get the filtered set of authenticationtacacspolicy resources configured on Citrix ADC. Filtering can be done on any of the properties of the resource.

view

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/authenticationtacacspolicy?**view=summary**

Note: By default, the retrieved results are displayed in detail view (?view=detail).

pagination

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/authenticationtacacspolicy?**pagesize=#no;pageno=#no**

Use this query-parameter to get the authenticationtacacspolicy resources in chunks.

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "authenticationtacacspolicy": [ {

"name":<String_value>,

"rule":<String_value>,

"reqaction":<String_value>

}]}
```

get

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/authenticationtacacspolicy/name_value<String>

Query-parameters:

attrs

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/authenticationtacacspolicy/name_value<String>?**attrs=property-name1,property-name2**

Use this query parameter to specify the resource details that you want to retrieve.

view

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/authenticationtacacspolicy/name_value<String>?**view=summary**

Note: By default, the retrieved results are displayed in detail view (?view=detail).

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "authenticationtacacspolicy": [ {

"name":<String_value>,

"rule":<String_value>,

"reqaction":<String_value>

}]}
```

count

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/authenticationtacacspolicy?**count=yes**

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "authenticationtacacspolicy": [ { "__count": "#no"} ] }
```

authenticationtacacspolicy_authenticationvserver_binding

Binding object showing the authenticationvserver that can be bound to authenticationtacacspolicy.

Properties

(click to see Operations)

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
name	<String>	Read-write	Name of the TACACS+ policy. Minimum length = 1
boundto	<String>	Read-write	The entity name to which policy is bound.
priority	<Double>	Read-only	.
activepolicy	<Double>	Read-only	.
__count	<Double>	Read-only	count parameter

Operations

(click to see Properties)

GET| GET (ALL)| COUNT

Some options that you can use for each operations:

- **Getting warnings in response:**NITRO allows you to get warnings in an operation by specifying the "warning" query parameter as "yes". For example, to get warnings while connecting to the Citrix ADC appliance, the URL is as follows:

  http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/login?warning=yes

  If any, the warnings are displayed in the response payload with the HTTP code "209 X-NITRO-WARNING".

- **Authenticated access for individual NITRO operations:**NITRO allows you to logon to the Citrix ADC appliance to perform individual operations. You can use this option instead of creating a NITRO session (using the login object) and then using that session to perform all operations,

  To do this, you must specify the username and password in the request header of the NITRO request as follows:

  X-NITRO-USER:<username>

  X-NITRO-PASS:<password>

  Note: In such cases, make sure that the request header DOES not include the following:

  Cookie:NITRO_AUTH_TOKEN=<tokenvalue>

`*Note:`*

Mandatory parameters are marked in redand placeholder content is marked in <green>.

get

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/authenticationtacacspolicy_authenticationvserver_binding/name_value<String>

Query-parameters:

filter

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/authenticationtacacspolicy_authenticationvserver_binding/name_value<String>?**filter=property-name1:property-value1,property-name2:property-value2**

Use this query-parameter to get the filtered set of authenticationtacacspolicy_authenticationvserver_binding resources configured on Citrix ADC. Filtering can be done on any of the properties of the resource.

pagination

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/authenticationtacacspolicy_authenticationvserver_binding/name_value<String>?**pagesize=#no;pageno=#no**

Use this query-parameter to get the authenticationtacacspolicy_authenticationvserver_binding resources in chunks.

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "authenticationtacacspolicy_authenticationvserver_binding": [ {

"name":<String_value>,

"boundto":<String_value>,

"priority":<Double_value>,

"activepolicy":<Double_value>

}]}
```

get (all)

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/authenticationtacacspolicy_authenticationvserver_binding

Query-parameters:

bulkbindings

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/authenticationtacacspolicy_authenticationvserver_binding?**bulkbindings=yes**

NITRO allows you to fetch bindings in bulk.

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "authenticationtacacspolicy_authenticationvserver_binding": [ {

"name":<String_value>,

"boundto":<String_value>,

"priority":<Double_value>,

"activepolicy":<Double_value>

}]}
```

count

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/authenticationtacacspolicy_authenticationvserver_binding/name_value<String>?**count=yes**

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{"authenticationtacacspolicy_authenticationvserver_binding":
[ { "__count": "#no"} ] }
```

authenticationtacacspolicy_binding

Binding object which returns the resources bound to authenticationtacacspolicy.

Properties

(click to see Operations)

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
name	<String>	Read-write	Name of the TACACS+ policy. Minimum length = 1
authenticationtacacspolicy_vpnglobal_binding	<authenticationtacacspolicy_vpnglobal_binding[]>	Read-only	vpnglobal that can be bound to authenticationtacacspolicy.

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
authenticationtacacspolicy_vpnvserver_binding	<authenticationtacacspolicy_vpnvserver_binding[]>	Read-only	vpnvserver that can be bound to authenticationtacacspolicy.
authenticationtacacspolicy_systemglobal_binding	<authenticationtacacspolicy_systemglobal_binding[]>	Read-only	systemglobal that can be bound to authenticationtacacspolicy.
authenticationtacacspolicy_authenticationvserver_binding	<authenticationtacacspolicy_authenticationvserver_binding[]>	Read-only	authenticationvserver that can be bound to authenticationtacacspolicy.

Operations

(click to see Properties)

GET| GET (ALL)

Some options that you can use for each operations:

- **Getting warnings in response:**NITRO allows you to get warnings in an operation by specifying the "warning" query parameter as "yes". For example, to get warnings while connecting to the Citrix ADC appliance, the URL is as follows:

  http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/login?warning=yes

  If any, the warnings are displayed in the response payload with the HTTP code "209 X-NITRO-WARNING".

- **Authenticated access for individual NITRO operations:**NITRO allows you to logon to the Citrix ADC appliance to perform individual operations. You can use this option instead of creating a NITRO session (using the login object) and then using that session to perform all operations,

  To do this, you must specify the username and password in the request header of the NITRO request as follows:

  X-NITRO-USER:<username>

X-NITRO-PASS:<password>

Note: In such cases, make sure that the request header DOES not include the following:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue>

`*Note:`*

Mandatory parameters are marked in redand placeholder content is marked in <green>.

get

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/authenticationtacacspolicy_binding/name_value<String>

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "authenticationtacacspolicy_binding": [ {

"name":<String_value>,

"authenticationtacacspolicy_vpnvserver_binding":<authenticationtacacspolic
y_vpnvserver_binding[]_value>,

"authenticationtacacspolicy_systemglobal_binding":<authenticationtacacspol
icy_systemglobal_binding[]_value>,

"authenticationtacacspolicy_authenticationvserver_binding":<authentication
tacacspolicy_authenticationvserver_binding[]_value>,
```

```
"authenticationtacacspolicy_vpnglobal_binding":<authenticationtacacspolicy
_vpnglobal_binding[]_value>

}]}
```

get (all)

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/authenticationtacacspolicy_binding

Query-parameters:

bulkbindings

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/authenticationtacacspolicy_binding?**bulkbindings=yes**

NITRO allows you to fetch bindings in bulk.

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "authenticationtacacspolicy_binding": [ {

"name":<String_value>,

"authenticationtacacspolicy_vpnvserver_binding":<authenticationtacacspolic
y_vpnvserver_binding[]_value>,
```

```
"authenticationtacacspolicy_systemglobal_binding":<authenticationtacacspol
icy_systemglobal_binding[]_value>,

"authenticationtacacspolicy_authenticationvserver_binding":<authentication
tacacspolicy_authenticationvserver_binding[]_value>,

"authenticationtacacspolicy_vpnglobal_binding":<authenticationtacacspolicy
_vpnglobal_binding[]_value>

}]}
```

authenticationtacacspolicy_systemglobal_binding

Binding object showing the systemglobal that can be bound to authenticationtacacspolicy.

Properties

(click to see Operations)

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
name	<String>	Read-write	Name of the TACACS+ policy. Minimum length = 1
boundto	<String>	Read-write	The entity name to which policy is bound.
priority	<Double>	Read-only	.
activepolicy	<Double>	Read-only	.
__count	<Double>	Read-only	count parameter

Operations

(click to see Properties)

GET| GET (ALL)| COUNT

Some options that you can use for each operations:

- **Getting warnings in response:**NITRO allows you to get warnings in an operation by specifying the "warning" query parameter as "yes". For example, to get warnings while connecting to the Citrix ADC appliance, the URL is as follows:

  http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/login?warning=yes

  If any, the warnings are displayed in the response payload with the HTTP code "209 X-NITRO-WARNING".

- **Authenticated access for individual NITRO operations:**NITRO allows you to logon to the Citrix ADC appliance to perform individual operations. You can use this option instead of creating a NITRO session (using the login object) and then using that session to perform all operations,

  To do this, you must specify the username and password in the request header of the NITRO request as follows:

  X-NITRO-USER:<username>

  X-NITRO-PASS:<password>

  Note: In such cases, make sure that the request header DOES not include the following:

  Cookie:NITRO_AUTH_TOKEN=<tokenvalue>

`*Note:`*

Mandatory parameters are marked in redand placeholder content is marked in <green>.

get

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/authenticationtacacspolicy_systemglobal_binding/name_value<String>

Query-parameters:

filter

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/authenticationtacacspolicy_systemglobal_binding/name_value<String>?**filter=property-name1:property-value1,property-name2:property-value2**

Use this query-parameter to get the filtered set of authenticationtacacspolicy_systemglobal_binding resources configured on Citrix ADC. Filtering can be done on any of the properties of the resource.

pagination

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/authenticationtacacspolicy_systemglobal_binding/name_value<String>?**pagesize=#no;pageno=#no**

Use this query-parameter to get the authenticationtacacspolicy_systemglobal_binding resources in chunks.

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "authenticationtacacspolicy_systemglobal_binding": [ {

"name":<String_value>,

"boundto":<String_value>,

"priority":<Double_value>,

"activepolicy":<Double_value>

}]}
```

get (all)

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/authenticationtacacspolicy_systemglobal_binding

Query-parameters:

bulkbindings

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/authenticationtacacspolicy_systemglobal_binding?**bulkbindings=yes**

NITRO allows you to fetch bindings in bulk.

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "authenticationtacacspolicy_systemglobal_binding": [ {

"name":<String_value>,

"boundto":<String_value>,

"priority":<Double_value>,

"activepolicy":<Double_value>

}]}
```

count

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/authenticationtacacspolicy_systemglobal_binding/name_value<String>?**count=yes**

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{"authenticationtacacspolicy_systemglobal_binding": [ { "__count":
"#no"} ] }
```

authenticationtacacspolicy_vpnglobal_binding

Binding object showing the vpnglobal that can be bound to authenticationtacacspolicy.

Properties

(click to see Operations)

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
name	<String>	Read-write	Name of the TACACS+ policy. Minimum length = 1
boundto	<String>	Read-write	The entity name to which policy is bound.
priority	<Double>	Read-only	.
activepolicy	<Double>	Read-only	.

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
__count	<Double>	Read-only	count parameter

Operations

(click to see Properties)

GET| GET (ALL)| COUNT

Some options that you can use for each operations:

- **Getting warnings in response:**NITRO allows you to get warnings in an operation by specifying the "warning" query parameter as "yes". For example, to get warnings while connecting to the Citrix ADC appliance, the URL is as follows:

  http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/login?warning=yes

  If any, the warnings are displayed in the response payload with the HTTP code "209 X-NITRO-WARNING".
- **Authenticated access for individual NITRO operations:**NITRO allows you to logon to the Citrix ADC appliance to perform individual operations. You can use this option instead of creating a NITRO session (using the login object) and then using that session to perform all operations,

  To do this, you must specify the username and password in the request header of the NITRO request as follows:

  X-NITRO-USER:<username>

  X-NITRO-PASS:<password>

  Note: In such cases, make sure that the request header DOES not include the following:

  Cookie:NITRO_AUTH_TOKEN=<tokenvalue>

`*Note:`*

Mandatory parameters are marked in redand placeholder content is marked in <green>.

get

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/authenticationtacacspolicy_vpnglobal_binding/name_value<String>

Query-parameters:

filter

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/authenticationtacacspolicy_vpnglobal_binding/name_value<String>?**filter=property-name1:property-value1,property-name2:property-value2**

Use this query-parameter to get the filtered set of authenticationtacacspolicy_vpnglobal_binding resources configured on Citrix ADC. Filtering can be done on any of the properties of the resource.

pagination

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/authenticationtacacspolicy_vpnglobal_binding/name_value<String>?**pagesize=#no;pageno=#no**

Use this query-parameter to get the authenticationtacacspolicy_vpnglobal_binding resources in chunks.

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "authenticationtacacspolicy_vpnglobal_binding": [ {

"name":<String_value>,

"boundto":<String_value>,

"priority":<Double_value>,
```

```
"activepolicy":<Double_value>

}]}
```

get (all)

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/authenticationtacacspolicy_vpnglobal_binding

Query-parameters:

bulkbindings

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/authenticationtacacspolicy_vpnglobal_binding?**bulkbindings=yes**

NITRO allows you to fetch bindings in bulk.

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "authenticationtacacspolicy_vpnglobal_binding": [ {

"name":<String_value>,

"boundto":<String_value>,

"priority":<Double_value>,
```

```
"activepolicy":<Double_value>

}]}
```

count

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/authenticationtacacspolicy_vpnglobal_binding/name_value<String>?**count=yes**

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{"authenticationtacacspolicy_vpnglobal_binding": [ { "__count": "#no"} ] }
```

authenticationtacacspolicy_vpnvserver_binding

Binding object showing the vpnvserver that can be bound to authenticationtacacspolicy.

Properties

(click to see Operations)

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
name	<String>	Read-write	Name of the TACACS+ policy.

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
			Minimum length = 1
boundto	<String>	Read-write	The entity name to which policy is bound.
priority	<Double>	Read-only	.
activepolicy	<Double>	Read-only	.
__count	<Double>	Read-only	count parameter

Operations

(click to see Properties)

GET| GET (ALL)| COUNT

Some options that you can use for each operations:

- **Getting warnings in response:**NITRO allows you to get warnings in an operation by specifying the "warning" query parameter as "yes". For example, to get warnings while connecting to the Citrix ADC appliance, the URL is as follows:

  http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/login?warning=yes

  If any, the warnings are displayed in the response payload with the HTTP code "209 X-NITRO-WARNING".

- **Authenticated access for individual NITRO operations:**NITRO allows you to logon to the Citrix ADC appliance to perform individual operations. You can use this option instead of creating a NITRO session (using the login object) and then using that session to perform all operations,

  To do this, you must specify the username and password in the request header of the NITRO request as follows:

  X-NITRO-USER:<username>

  X-NITRO-PASS:<password>

  Note: In such cases, make sure that the request header DOES not include the following:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue>

`*Note:`*

Mandatory parameters are marked in redand placeholder content is marked in <green>.

get

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/authenticationtacacspolicy_vpnvserver_binding/name_value<String>

Query-parameters:

filter

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/authenticationtacacspolicy_vpnvserver_binding/name_value<String>?**filter=property-name1:property-value1,property-name2:property-value2**

Use this query-parameter to get the filtered set of authenticationtacacspolicy_vpnvserver_binding resources configured on Citrix ADC. Filtering can be done on any of the properties of the resource.

pagination

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/authenticationtacacspolicy_vpnvserver_binding/name_value<String>?**pagesize=#no;pageno=#no**

Use this query-parameter to get the authenticationtacacspolicy_vpnvserver_binding resources in chunks.

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "authenticationtacacspolicy_vpnvserver_binding": [ {

"name":<String_value>,

"boundto":<String_value>,

"priority":<Double_value>,

"activepolicy":<Double_value>

}]}
```

get (all)

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/authenticationtacacspolicy_vpnvserver_binding

Query-parameters:

bulkbindings

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/authenticationtacacspolicy_vpnvserver_binding?**bulkbindings=yes**

NITRO allows you to fetch bindings in bulk.

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "authenticationtacacspolicy_vpnvserver_binding": [ {

"name":<String_value>,

"boundto":<String_value>,

"priority":<Double_value>,

"activepolicy":<Double_value>

}]}
```

count

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/authenticationtacacspolicy_vpnvserver_binding/name_value<String>?**count=yes**

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{"authenticationtacacspolicy_vpnvserver_binding": [ { "__count":
"#no"} ] }
```

authenticationvserver

Configuration for authentication virtual server resource.

Properties

(click to see Operations)

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
name	<String>	Read-write	Name for the new authentication virtual server. Must begin with a letter, number, or the underscore character (_), and must contain only letters, numbers, and the hyphen (-), period (.) pound (#), space ( ), at (@), equals (=), colon (:), and underscore characters. Can be changed after the authentication virtual server is added by using the rename authentication vserver command. The following requirement applies only to the Citrix ADC CLI: If the name includes one or more spaces, enclose the

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
			name in double or single quotation marks (for example, "my authentication policy" or 'my authentication policy'). Minimum length = 1
servicetype	<String>	Read-write	Protocol type of the authentication virtual server. Always SSL. Default value: SSL Possible values = SSL
ipv46	<String>	Read-write	IP address of the authentication virtual server, if a single IP address is assigned to the virtual server. Minimum length = 1
range	<Double>	Read-write	If you are creating a series of virtual servers with a range of IP addresses assigned to them, the length of the range. The new range of authentication virtual servers will have IP addresses consecutively numbered, starting with the primary

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
			address specified with the IP Address parameter. Default value: 1 Minimum value = 1
port	<Integer>	Read-write	TCP port on which the virtual server accepts connections. Range 1 - 65535 * in CLI is represented as 65535 in NITRO API
state	<String>	Read-write	Initial state of the new virtual server. Default value: ENABLED Possible values = ENABLED, DISABLED
authentication	<String>	Read-write	Require users to be authenticated before sending traffic through this virtual server. Default value: ON Possible values = ON, OFF
authenticationdomain	<String>	Read-write	The domain of the authentication cookie

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
			set by Authentication vserver. Minimum length = 3 Maximum length = 252
comment	<String>	Read-write	Any comments associated with this virtual server.
td	<Double>	Read-write	Integer value that uniquely identifies the traffic domain in which you want to configure the entity. If you do not specify an ID, the entity becomes part of the default traffic domain, which has an ID of 0. Minimum value = 0 Maximum value = 4094
appflowlog	<String>	Read-write	Log AppFlow flow information. Default value: ENABLED Possible values = ENABLED, DISABLED
maxloginattempts	<Double>	Read-write	Maximum Number of login Attempts. Minimum value = 1

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
			Maximum value = 255
failedlogintimeout	<Double>	Read-write	Number of minutes an account will be locked if user exceeds maximum permissible attempts. Minimum value = 1
certkeynames	<String>	Read-write	Name of the certificate key that was bound to the corresponding SSL virtual server as the Certificate Authority for the device certificate. Minimum length = 1 Maximum length = 127
newname	<String>	Read-write	New name of the authentication virtual server. Must begin with a letter, number, or the underscore character (_), and must contain only letters, numbers, and the hyphen (-), period (.) pound (#), space ( ), at (@), equals (=), colon (:), and underscore characters.

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
			The following requirement applies only to the Citrix ADC CLI: If the name includes one or more spaces, enclose the name in double or single quotation marks (for example, 'my authentication policy' or "my authentication policy"). Minimum length = 1
ip	<String>	Read-only	The Virtual IP address of the authentication vserver.
value	<String>	Read-only	Indicates whether or not the certificate is bound or if SSL offload is disabled. Possible values = Certkey not bound, SSL feature disabled
type	<String>	Read-only	The type of Virtual Server, e.g. CONTENT based or ADDRESS based.

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
			Possible values = CONTENT, ADDRESS
curstate	<String>	Read-only	The current state of the Virtual server, e.g. UP, DOWN, BUSY, etc. Possible values = UP, DOWN, UNKNOWN, BUSY, OUT OF SERVICE, GOING OUT OF SERVICE, DOWN WHEN GOING OUT OF SERVICE, NS_EMPTY_STR, Unknown, DISABLED
status	<Integer>	Read-only	Whether or not this vserver responds to ARPs and whether or not round-robin selection is temporarily in effect.
cachetype	<String>	Read-only	Virtual server's cache type. The options are: TRANSPARENT, REVERSE and FORWARD. Possible values = TRANSPARENT, REVERSE, FORWARD

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
redirect	<String>	Read-only	The cache redirect policy. The valid redirect policies are: l. CACHE - Directs all requests to the cache. 2. POLICY - Applies cache redirection policy to determine whether the request should be directed to the cache or origin. This is the default setting. 3. ORIGIN - Directs all requests to the origin server. Possible values = CACHE, POLICY, ORIGIN
precedence	<String>	Read-only	This argument is used only when configuring content switching on the specified virtual server. This is applicable only if both the URL and RULE-based policies have been

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
			configured on the same virtual server. It specifies the type of policy (URL or RULE) that takes precedence on the content switching virtual server. The default setting is RULE. l URL - In this case, the incoming request is matched against the URL-based policies before the rule-based policies. l RULE - In this case, the incoming request is matched against the rule-based policies before the URL-based policies. For all URL-based policies, the precedence hierarchy is: 1. Domain and exact URL

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
			2. Domain, prefix and suffix 3. Domain and suffix 4. Domain and prefix 5. Domain only 6. Exact URL 7. Prefix and suffix 8. Suffix only 9. Prefix only 10. Default. Possible values = RULE, URL
redirecturl	<String>	Read-only	The URL where traffic is redirected if the virtual server in system becomes unavailable. WARNING! Make sure that the domain you specify in the URL does not match the domain specified in the -d domainName argument of the ###add cs policy### command. If the same domain is specified in both arguments, the

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
			request will be continuously redirected to the same unavailable virtual server in the system. If so, the user may not get the requested content.
curaaausers	<Double>	Read-only	The number of current users logged in to this vserver.
policy	<String>	Read-only	The name of the policy, if any, bound to the authentication vserver.
servicename	<String>	Read-only	The name of the service, if any, to which the vserver policy is bound.
weight	<Double>	Read-only	Weight for this service, if any. This weight is used when the system performs load balancing, giving greater priority to a specific service. It is useful when the services bound to a virtual server are of different capacity.
cachevserver	<String>	Read-only	The name of the default target cache virtual server, if any, to which requests are redirected.

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
backupvserver	<String>	Read-only	The name of the backup vpn virtual server for this vpn virtual server.
clttimeout	<Double>	Read-only	The idle time, if any, in seconds after which the client connection is terminated.
somethod	<String>	Read-only	VPN client applications are allocated from a block of Intranet IP addresses. That block may be exhausted after a certain number of connections. This switch specifies the method used to determine whether or not a new connection will spillover, or exhaust, the allocated block of Intranet IP addresses for that application. Possible values are CONNECTION or DYNAMICCONNECTION.

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
			CONNECTION means that a static integer value is the hard limit for the spillover threshold. The spillover threshold is described below. DYNAMICCONNECTION means that the spillover threshold is set according to the maximum number of connections defined for the vpn vserver. Possible values = CONNECTION, DYNAMICCONNECTION, BANDWIDTH, HEALTH, NONE
sothreshold	<Double>	Read-only	VPN client applications are allocated from a block of Intranet IP addresses. That block may be exhausted after a certain number of connections.

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
			The value of this option is number of client connections after which the Mapped IP address is used as the client source IP address instead of an address from the allocated block of Intranet IP addresses.
sopersistence	<String>	Read-only	Whether or not cookie-based site persistance is enabled for this VPN vserver. Possible values are 'ConnectionProxy', HTTPRedirect, or NONE. Possible values = ENABLED, DISABLED
sopersistencetimeout	<Double>	Read-only	The timeout, if any, for cookie-based site persistance of this VPN vserver.
priority	<Double>	Read-only	The priority, if any, of the vpn vserver policy.
downstateflush	<String>	Read-only	Perform delayed clean up of

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
			connections on this vserver. Possible values = ENABLED, DISABLED
disableprimaryondown	<String>	Read-only	Tells whether traffic will continue reaching backup vservers even after primary comes UP from DOWN state. Possible values = ENABLED, DISABLED
listenpolicy	<String>	Read-only	Listenpolicy configured for authentication vserver.
listenpriority	<Double>	Read-only	Priority of listen policy for authentication vserver.
tcpprofilename	<String>	Read-only	The name of the TCP profile.
httpprofilename	<String>	Read-only	Name of the HTTP profile.
vstype	<Double>	Read-only	Virtual Server Type, e.g. Load Balancing, Content Switch, Cache Redirection.
ngname	<String>	Read-only	Nodegroup devno to which this

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
			authentication vsever belongs to.
secondary	<Boolean>	Read-only	Bind the authentication policy to the secondary chain. Provides for multifactor authentication in which a user must authenticate via both a primary authentication method and, afterward, via a secondary authentication method. Because user groups are aggregated across authentication systems, usernames must be the same on all authentication servers. Passwords can be different.
groupextraction	<Boolean>	Read-only	Bind the Authentication policy to a tertiary chain which will be used only for group extraction. The user will not authenticate

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
			against this server, and this will only be called if primary and/or secondary authentication has succeeded.
__count	<Double>	Read-only	count parameter

Operations

(click to see Properties)

ADD| DELETE| UPDATE| UNSET| ENABLE| DISABLE| RENAME| GET (ALL)| GET| COUNT

Some options that you can use for each operations:

- **Getting warnings in response:**NITRO allows you to get warnings in an operation by specifying the "warning" query parameter as "yes". For example, to get warnings while connecting to the Citrix ADC appliance, the URL is as follows:

  http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/login?warning=yes

  If any, the warnings are displayed in the response payload with the HTTP code "209 X-NITRO-WARNING".

- **Authenticated access for individual NITRO operations:**NITRO allows you to logon to the Citrix ADC appliance to perform individual operations. You can use this option instead of creating a NITRO session (using the login object) and then using that session to perform all operations,

  To do this, you must specify the username and password in the request header of the NITRO request as follows:

  X-NITRO-USER:<username>

  X-NITRO-PASS:<password>

  Note: In such cases, make sure that the request header DOES not include the following:

  Cookie:NITRO_AUTH_TOKEN=<tokenvalue>

`*Note:`*

Mandatory parameters are marked in redand placeholder content is marked in <green>.

add

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/authenticationvserver

HTTP Method:POST

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Content-Type:application/json

Request Payload:

```
{"authenticationvserver":{

<b>"name":<String_value>,

</b><b>"servicetype":<String_value>,

</b>"ipv46":<String_value>,

"range":<Double_value>,

"port":<Integer_value>,

"state":<String_value>,

"authentication":<String_value>,

"authenticationdomain":<String_value>,

"comment":<String_value>,

"td":<Double_value>,

"appflowlog":<String_value>,

"maxloginattempts":<Double_value>,

"failedlogintimeout":<Double_value>,

"certkeynames":<String_value>
```

```
}}
```

Response:

HTTP Status Code on Success: 201 Created HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error

delete

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/authenticationvserver/name_value<String>

HTTP Method:DELETE

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue>

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error

update

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/authenticationvserver

HTTP Method:PUT

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Content-Type:application/json

Request Payload:

```
{"authenticationvserver":{

<b>"name":<String_value>,
```

```
</b>"ipv46":<String_value>,

"authentication":<String_value>,

"authenticationdomain":<String_value>,

"comment":<String_value>,

"appflowlog":<String_value>,

"maxloginattempts":<Double_value>,

"failedlogintimeout":<Double_value>,

"certkeynames":<String_value>

}}
```

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error

unset

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/authenticationvserver?**action=unset**

HTTP Method:POST

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Content-Type:application/json

Request Payload:

```
{"authenticationvserver":{

<b>"name":<String_value>,
```

```
</b>"authenticationdomain":true,

"maxloginattempts":true,

"authentication":true,

"comment":true,

"appflowlog":true,

"failedlogintimeout":true,

"certkeynames":true

}}
```

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error

enable

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/authenticationvserver?**action=enable**

HTTP Method:POST

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Content-Type:application/json

Request Payload:

```
{"authenticationvserver":{

<b>"name":<String_value>

</b>}}
```

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error

disable

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/authenticationvserver?**action=disable**

HTTP Method:POST

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Content-Type:application/json

Request Payload:

```
{"authenticationvserver":{

<b>"name":<String_value>

</b>}}
```

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error

rename

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/authenticationvserver?**action=rename**

HTTP Method:POST

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Content-Type:application/json

Request Payload:

```
{"authenticationvserver":{

<b>"name":<String_value>,

</b><b>"newname":<String_value>

</b>}}
```

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error

get (all)

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/authenticationvserver

Query-parameters:

attrs

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/authenticationvserver?**attrs=property-name1,property-name2**

Use this query parameter to specify the resource details that you want to retrieve.

filter

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/authenticationvserver?**filter=property-name1:property-val1,property-name2:property-val2**

Use this query-parameter to get the filtered set of authenticationvserver resources configured on Citrix ADC. Filtering can be done on any of the properties of the resource.

view

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/authenticationvserver?**view=summary**

Note: By default, the retrieved results are displayed in detail view (?view=detail).

pagination

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/authenticationvserver?**pagesize=#no;pageno=#no**

Use this query-parameter to get the authenticationvserver resources in chunks.

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "authenticationvserver": [ {

"name":<String_value>,

"ip":<String_value>,

"td":<Double_value>,

"ipv46":<String_value>,

"value":<String_value>,

"port":<Integer_value>,

"range":<Double_value>,

"servicetype":<String_value>,
```

```
"type":<String_value>,

"curstate":<String_value>,

"status":<Integer_value>,

"cachetype":<String_value>,

"redirect":<String_value>,

"precedence":<String_value>,

"redirecturl":<String_value>,

"authentication":<String_value>,

"curaaausers":<Double_value>,

"authenticationdomain":<String_value>,

"policyname":<String_value>,

"policy":<String_value>,

"servicename":<String_value>,

"weight":<Double_value>,

"cachevserver":<String_value>,

"backupvserver":<String_value>,

"clttimeout":<Double_value>,

"somethod":<String_value>,

"sothreshold":<Double_value>,

"sopersistence":<String_value>,

"sopersistencetimeout":<Double_value>,

"priority":<Double_value>,
```

```
"downstateflush":<String_value>,

"disableprimaryondown":<String_value>,

"listenpolicy":<String_value>,

"listenpriority":<Double_value>,

"tcpprofilename":<String_value>,

"httpprofilename":<String_value>,

"comment":<String_value>,

"appflowlog":<String_value>,

"vstype":<Double_value>,

"ngname":<String_value>,

"maxloginattempts":<Double_value>,

"failedlogintimeout":<Double_value>,

"secondary":<Boolean_value>,

"groupextraction":<Boolean_value>,

"certkeynames":<String_value>

}]}
```

get

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/authenticationvserver/name_value<String>

Query-parameters:

attrs

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/authenticationvserver/name_value<String>?**attrs=property-name1,property-name2**

Use this query parameter to specify the resource details that you want to retrieve.

view

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/authenticationvserver/name_value<String>?**view=summary**

Note: By default, the retrieved results are displayed in detail view (?view=detail).

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "authenticationvserver": [ {

"name":<String_value>,

"ip":<String_value>,

"td":<Double_value>,

"ipv46":<String_value>,

"value":<String_value>,

"port":<Integer_value>,
```

```
"range":<Double_value>,

"servicetype":<String_value>,

"type":<String_value>,

"curstate":<String_value>,

"status":<Integer_value>,

"cachetype":<String_value>,

"redirect":<String_value>,

"precedence":<String_value>,

"redirecturl":<String_value>,

"authentication":<String_value>,

"curaaausers":<Double_value>,

"authenticationdomain":<String_value>,

"policyname":<String_value>,

"policy":<String_value>,

"servicename":<String_value>,

"weight":<Double_value>,

"cachevserver":<String_value>,

"backupvserver":<String_value>,

"clttimeout":<Double_value>,

"somethod":<String_value>,

"sothreshold":<Double_value>,

"sopersistence":<String_value>,
```

```
"sopersistencetimeout":<Double_value>,

"priority":<Double_value>,

"downstateflush":<String_value>,

"disableprimaryondown":<String_value>,

"listenpolicy":<String_value>,

"listenpriority":<Double_value>,

"tcpprofilename":<String_value>,

"httpprofilename":<String_value>,

"comment":<String_value>,

"appflowlog":<String_value>,

"vstype":<Double_value>,

"ngname":<String_value>,

"maxloginattempts":<Double_value>,

"failedlogintimeout":<Double_value>,

"secondary":<Boolean_value>,

"groupextraction":<Boolean_value>,

"certkeynames":<String_value>

}]}
```

count

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/authenticationvserver?**count=yes**

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "authenticationvserver": [ { "__count": "#no"} ] }
```

authenticationvserver_auditnslogpolicy_binding

Binding object showing the auditnslogpolicy that can be bound to authenticationvserver.

Properties

(click to see Operations)

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
priority	<Double>	Read-write	The priority, if any, of the vpn vserver policy.
name	<String>	Read-write	Name of the authentication virtual server to which to bind the policy. Minimum length = 1
nextfactor	<String>	Read-write	Applicable only while binding advance authentication policy as classic

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
			authentication policy does not support nFactor.
gotopriorityexpression	<String>	Read-write	Applicable only to advance authentication policy. Expression or other value specifying the next policy to be evaluated if the current policy evaluates to TRUE. Specify one of the following values: * NEXT - Evaluate the policy with the next higher priority number. * END - End policy evaluation. * USE_INVOCATION_RESULT - Applicable if this policy invokes another policy label. If the final goto in the invoked policy label has a value of END, the evaluation stops. If the final goto is anything other than END, the current policy label performs a NEXT. * An expression that evaluates to a number. If you specify an expression, the number to which it evaluates determines the next

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
			policy to evaluate, as follows: * If the expression evaluates to a higher numbered priority, the policy with that priority is evaluated next. * If the expression evaluates to the priority of the current policy, the policy with the next higher numbered priority is evaluated next. * If the expression evaluates to a priority number that is numerically higher than the highest numbered priority, policy evaluation ends. An UNDEF event is triggered if: * The expression is invalid. * The expression evaluates to a priority number that is numerically lower than the current policy's priority. * The expression evaluates to a priority number that is between the current policy's priority number (say, 30) and the highest priority number (say, 100), but does not match any configured priority

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
			number (for example, the expression evaluates to the number 85). This example assumes that the priority number increments by 10 for every successive policy, and therefore a priority number of 85 does not exist in the policy label.
secondary	<Boolean>	Read-write	Applicable only while bindind classic authentication policy as advance authentication policy use nFactor.
policy	<String>	Read-write	The name of the policy, if any, bound to the authentication vserver.
groupextraction	<Boolean>	Read-write	Applicable only while bindind classic authentication policy as advance authentication policy use nFactor.
__count	<Double>	Read-write	count parameter

Operations

(click to see Properties)

ADD:| DELETE:| GET| GET (ALL)| COUNT

Some options that you can use for each operations:

- **Getting warnings in response:**NITRO allows you to get warnings in an operation by specifying the "warning" query parameter as "yes". For example, to get warnings while connecting to the Citrix ADC appliance, the URL is as follows:

  http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/login?warning=yes

  If any, the warnings are displayed in the response payload with the HTTP code "209 X-NITRO-WARNING".

- **Authenticated access for individual NITRO operations:**NITRO allows you to logon to the Citrix ADC appliance to perform individual operations. You can use this option instead of creating a NITRO session (using the login object) and then using that session to perform all operations,

  To do this, you must specify the username and password in the request header of the NITRO request as follows:

  X-NITRO-USER:<username>

  X-NITRO-PASS:<password>

  Note: In such cases, make sure that the request header DOES not include the following:

  Cookie:NITRO_AUTH_TOKEN=<tokenvalue>

`*Note:`*

Mandatory parameters are marked in redand placeholder content is marked in <green>.

add

URL:http://<Citrix-ADC-IP-address(NSIP)/nitro/v1/config/authenticationvserver_auditnslogpolicy_binding

HTTP Method:PUT

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Content-Type:application/json

Request Payload:

```
{
```

```
"authenticationvserver_auditnslogpolicy_binding":{

<b>"name":<String_value>,

</b>"policy":<String_value>,

"priority":<Double_value>,

"secondary":<Boolean_value>,

"groupextraction":<Boolean_value>,

"nextfactor":<String_value>,

"gotopriorityexpression":<String_value>

}}
```

Response:

HTTP Status Code on Success: 201 Created HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error

delete

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/authenticationvserver_auditnslogpolicy_binding/name_value<String>

Query-parameters:

args

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/authenticationvserver_auditnslogpolicy_binding/name_value<String>?**args=policy:<String_value>,secondary:<Boolean_value>,groupextraction:<Boolean_value>**

HTTP Method:DELETE

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue>

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error

get

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/authenticationvserver_auditnslogpolicy_binding/name_value<String>

Query-parameters:

filter

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/authenticationvserver_auditnslogpolicy_binding/name_value<String>?**filter=property-name1:property-value1,property-name2:property-value2**

Use this query-parameter to get the filtered set of authenticationvserver_auditnslogpolicy_binding resources configured on Citrix ADC. Filtering can be done on any of the properties of the resource.

pagination

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/authenticationvserver_auditnslogpolicy_binding/name_value<String>?**pagesize=#no;pageno=#no**

Use this query-parameter to get the authenticationvserver_auditnslogpolicy_binding resources in chunks.

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "authenticationvserver_auditnslogpolicy_binding": [ {

"priority":<Double_value>,

"name":<String_value>,

"nextfactor":<String_value>,

"gotopriorityexpression":<String_value>,

"secondary":<Boolean_value>,

"policy":<String_value>,

"groupextraction":<Boolean_value>

}]}
```

get (all)

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/authenticationvserver_auditnslogpolicy_binding

Query-parameters:

bulkbindings

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/authenticationvserver_auditnslogpolicy_binding?**bulkbindings=yes**

NITRO allows you to fetch bindings in bulk.

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "authenticationvserver_auditnslogpolicy_binding": [ {

"priority":<Double_value>,

"name":<String_value>,

"nextfactor":<String_value>,

"gotopriorityexpression":<String_value>,

"secondary":<Boolean_value>,

"policy":<String_value>,

"groupextraction":<Boolean_value>

}]}
```

count

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/authenticationvserver_auditnslogpolicy_binding/name_value<String>?**count=yes**

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{"authenticationvserver_auditnslogpolicy_binding": [ { "__count":
"#no"} ] }
```

authenticationvserver_auditsyslogpolicy_binding

Binding object showing the auditsyslogpolicy that can be bound to authenticationvserver.

Properties

(click to see Operations)

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
priority	<Double>	Read-write	The priority, if any, of the vpn vserver policy.
name	<String>	Read-write	Name of the authentication virtual server to which to bind the policy. Minimum length = 1
nextfactor	<String>	Read-write	Applicable only while binding advance authentication policy as classic authentication policy does not support nFactor.

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
gotopriorityexpression	<String>	Read-write	Applicable only to advance authentication policy. Expression or other value specifying the next policy to be evaluated if the current policy evaluates to TRUE. Specify one of the following values: * NEXT - Evaluate the policy with the next higher priority number. * END - End policy evaluation. * USE_INVOCATION_RESULT - Applicable if this policy invokes another policy label. If the final goto in the invoked policy label has a value of END, the evaluation stops. If the final goto is anything other than END, the current policy label performs a NEXT. * An expression that evaluates to a number. If you specify an expression, the number to which it evaluates determines the next policy to evaluate, as follows: * If the expression evaluates to a higher

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
			numbered priority, the policy with that priority is evaluated next. * If the expression evaluates to the priority of the current policy, the policy with the next higher numbered priority is evaluated next. * If the expression evaluates to a priority number that is numerically higher than the highest numbered priority, policy evaluation ends. An UNDEF event is triggered if: * The expression is invalid. * The expression evaluates to a priority number that is numerically lower than the current policy's priority. * The expression evaluates to a priority number that is between the current policy's priority number (say, 30) and the highest priority number (say, 100), but does not match any configured priority number (for example, the expression evaluates to the number 85).

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
			This example assumes that the priority number increments by 10 for every successive policy, and therefore a priority number of 85 does not exist in the policy label.
secondary	<Boolean>	Read-write	Applicable only while bindind classic authentication policy as advance authentication policy use nFactor.
policy	<String>	Read-write	The name of the policy, if any, bound to the authentication vserver.
groupextraction	<Boolean>	Read-write	Applicable only while bindind classic authentication policy as advance authentication policy use nFactor.
__count	<Double>	Read-write	count parameter

Operations

(click to see Properties)

ADD:| DELETE:| GET| GET (ALL)| COUNT

Some options that you can use for each operations:

- **Getting warnings in response:**NITRO allows you to get warnings in an operation by specifying the "warning" query parameter as "yes". For example, to get warnings while connecting to the Citrix ADC appliance, the URL is as follows:

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/login?warning=yes

If any, the warnings are displayed in the response payload with the HTTP code "209 X-NITRO-WARNING".

- **Authenticated access for individual NITRO operations:**NITRO allows you to logon to the Citrix ADC appliance to perform individual operations. You can use this option instead of creating a NITRO session (using the login object) and then using that session to perform all operations,

  To do this, you must specify the username and password in the request header of the NITRO request as follows:

  X-NITRO-USER:<username>

  X-NITRO-PASS:<password>

  Note: In such cases, make sure that the request header DOES not include the following:

  Cookie:NITRO_AUTH_TOKEN=<tokenvalue>

`*Note:`*

Mandatory parameters are marked in redand placeholder content is marked in <green>.

add

URL:http://<Citrix-ADC-IP-address(NSIP)/nitro/v1/config/authenticationvserver_auditsyslogpolicy_binding

HTTP Method:PUT

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Content-Type:application/json

Request Payload:

```
{

"authenticationvserver_auditsyslogpolicy_binding":{

<b>"name":<String_value>,
```

```
</b>"policy":<String_value>,

"priority":<Double_value>,

"secondary":<Boolean_value>,

"groupextraction":<Boolean_value>,

"nextfactor":<String_value>,

"gotopriorityexpression":<String_value>

}}
```

Response:

HTTP Status Code on Success: 201 Created HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error

delete

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/authenticationvserver_auditsyslogpolicy_binding/name_value<String>

Query-parameters:

args

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/authenticationvserver_auditsyslogpolicy_binding/name_value<String>?**args=policy:<String_value>,secondary:<Boolean_value>,groupextraction:<Boolean_value>**

HTTP Method:DELETE

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue>

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error

get

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/authenticationvserver_auditsyslogpolicy_binding/name_value<String>

Query-parameters:

filter

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/authenticationvserver_auditsyslogpolicy_binding/name_value<String>?**filter=property-name1:property-value1,property-name2:property-value2**

Use this query-parameter to get the filtered set of authenticationvserver_auditsyslogpolicy_binding resources configured on Citrix ADC. Filtering can be done on any of the properties of the resource.

pagination

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/authenticationvserver_auditsyslogpolicy_binding/name_value<String>?**pagesize=#no;pageno=#no**

Use this query-parameter to get the authenticationvserver_auditsyslogpolicy_binding resources in chunks.

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "authenticationvserver_auditsyslogpolicy_binding": [ {

"priority":<Double_value>,

"name":<String_value>,

"nextfactor":<String_value>,

"gotopriorityexpression":<String_value>,

"secondary":<Boolean_value>,

"policy":<String_value>,

"groupextraction":<Boolean_value>

}]}
```

get (all)

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/authenticationvserver_auditsyslogpolicy_binding

Query-parameters:

bulkbindings

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/authenticationvserver_auditsyslogpolicy_binding?**bulkbindings=yes**

NITRO allows you to fetch bindings in bulk.

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "authenticationvserver_auditsyslogpolicy_binding": [ {

"priority":<Double_value>,

"name":<String_value>,

"nextfactor":<String_value>,

"gotopriorityexpression":<String_value>,

"secondary":<Boolean_value>,

"policy":<String_value>,

"groupextraction":<Boolean_value>

}]}
```

count

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/authenticationvserver_auditsyslogpolicy_binding/name_value<String>?**count=yes**

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{"authenticationvserver_auditsyslogpolicy_binding": [ { "__count":
"#no"} ] }
```

authenticationvserver_authenticationcertpolicy_binding

Binding object showing the authenticationcertpolicy that can be bound to authenticationvserver.

Properties

(click to see Operations)

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
priority	<Double>	Read-write	The priority, if any, of the vpn vserver policy.
name	<String>	Read-write	Name of the authentication virtual server to which to bind the policy. Minimum length = 1
nextfactor	<String>	Read-write	Applicable only while binding advance authentication policy as classic authentication policy does not support nFactor.

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
gotopriorityexpression	<String>	Read-write	Applicable only to advance authentication policy. Expression or other value specifying the next policy to be evaluated if the current policy evaluates to TRUE. Specify one of the following values: * NEXT - Evaluate the policy with the next higher priority number. * END - End policy evaluation. * USE_INVOCATION_RESULT - Applicable if this policy invokes another policy label. If the final goto in the invoked policy label has a value of END, the evaluation stops. If the final goto is anything other than END, the current policy label performs a NEXT. * An expression that evaluates to a number. If you specify an expression, the number to which it evaluates determines the next policy to evaluate, as follows: * If the expression evaluates to a higher

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
			numbered priority, the policy with that priority is evaluated next. * If the expression evaluates to the priority of the current policy, the policy with the next higher numbered priority is evaluated next. * If the expression evaluates to a priority number that is numerically higher than the highest numbered priority, policy evaluation ends. An UNDEF event is triggered if: * The expression is invalid. * The expression evaluates to a priority number that is numerically lower than the current policy's priority. * The expression evaluates to a priority number that is between the current policy's priority number (say, 30) and the highest priority number (say, 100), but does not match any configured priority number (for example, the expression evaluates to the number 85).

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
			This example assumes that the priority number increments by 10 for every successive policy, and therefore a priority number of 85 does not exist in the policy label.
secondary	<Boolean>	Read-write	Bind the authentication policy to the secondary chain. Provides for multifactor authentication in which a user must authenticate via both a primary authentication method and, afterward, via a secondary authentication method. Because user groups are aggregated across authentication systems, usernames must be the same on all authentication servers. Passwords can be different.
policy	<String>	Read-write	The name of the policy, if any, bound to the authentication vserver.
groupextraction	<Boolean>	Read-write	Applicable only while bindind classic

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
			authentication policy as advance authentication policy use nFactor.
acttype	<Double>	Read-only	.
__count	<Double>	Read-only	count parameter

Operations

(click to see Properties)

ADD:| DELETE:| GET| GET (ALL)| COUNT

Some options that you can use for each operations:

- **Getting warnings in response:**NITRO allows you to get warnings in an operation by specifying the "warning" query parameter as "yes". For example, to get warnings while connecting to the Citrix ADC appliance, the URL is as follows:

  http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/login?warning=yes

  If any, the warnings are displayed in the response payload with the HTTP code "209 X-NITRO-WARNING".

- **Authenticated access for individual NITRO operations:**NITRO allows you to logon to the Citrix ADC appliance to perform individual operations. You can use this option instead of creating a NITRO session (using the login object) and then using that session to perform all operations,

  To do this, you must specify the username and password in the request header of the NITRO request as follows:

  X-NITRO-USER:<username>

  X-NITRO-PASS:<password>

  Note: In such cases, make sure that the request header DOES not include the following:

  Cookie:NITRO_AUTH_TOKEN=<tokenvalue>

`*Note:` *

Mandatory parameters are marked in redand placeholder content is marked in <green>.

add

URL:http://<Citrix-ADC-IP-address(NSIP)/nitro/v1/config/authenticationvserver_authenticationcertpolicy_binding

HTTP Method:PUT

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Content-Type:application/json

Request Payload:

```
{

"authenticationvserver_authenticationcertpolicy_binding":{

<b>"name":<String_value>,

</b>"policy":<String_value>,

"priority":<Double_value>,

"secondary":<Boolean_value>,

"groupextraction":<Boolean_value>,

"nextfactor":<String_value>,

"gotopriorityexpression":<String_value>

}}
```

Response:

HTTP Status Code on Success: 201 Created HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error

delete

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/authenticationvserver_authenticationcertpolicy_binding/name_value<String>

Query-parameters:

args

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/authenticationvserver_authenticationcertpolicy_binding/name_value<String>?**args=policy:<String_value>,secondary:<Boolean_value>,groupextraction:<Boolean_value>**

HTTP Method:DELETE

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue>

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error

get

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/authenticationvserver_authenticationcertpolicy_binding/name_value<String>

Query-parameters:

filter

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/authenticationvserver_authenticationcertpolicy_binding/name_value<String>?**filter=property-name1:property-value1,property-name2:property-value2**

Use this query-parameter to get the filtered set of authenticationvserver_authenticationcertpolicy_binding resources configured on Citrix ADC. Filtering can be done on any of the properties of the resource.

pagination

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/authenticationvserver_authenticationcertpolicy_binding/name_value<String>?**pagesize=#no;pageno=#no**

Use this query-parameter to get the authenticationvserver_authenticationcertpolicy_binding resources in chunks.

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "authenticationvserver_authenticationcertpolicy_binding": [ {

"priority":<Double_value>,

"name":<String_value>,

"nextfactor":<String_value>,

"gotopriorityexpression":<String_value>,

"secondary":<Boolean_value>,

"policy":<String_value>,

"groupextraction":<Boolean_value>,

"acttype":<Double_value>

}]}
```

get (all)

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/authenticationvserver_authenticationcertpolicy_binding

Query-parameters:

bulkbindings

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/authenticationvserver_authenticationcertpolicy_binding?**bulkbindings=yes**

NITRO allows you to fetch bindings in bulk.

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "authenticationvserver_authenticationcertpolicy_binding": [ {

"priority":<Double_value>,

"name":<String_value>,

"nextfactor":<String_value>,

"gotopriorityexpression":<String_value>,

"secondary":<Boolean_value>,
```

```
"policy":<String_value>,

"groupextraction":<Boolean_value>,

"acttype":<Double_value>

}]}
```

count

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/authenticationvserver_authenticationcertpolicy_binding/name_value<String>?**count=yes**

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{"authenticationvserver_authenticationcertpolicy_binding": [ { "__count":
"#no"} ] }
```

authenticationvserver_authenticationldappolicy_binding

Binding object showing the authenticationldappolicy that can be bound to authenticationvserver.

Properties

(click to see Operations)

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
priority	<Double>	Read-write	The priority, if any, of the vpn vserver policy.
name	<String>	Read-write	Name of the authentication virtual server to which to bind the policy. Minimum length = 1
nextfactor	<String>	Read-write	Applicable only while binding advance authentication policy as classic authentication policy does not support nFactor.
gotopriorityexpression	<String>	Read-write	Applicable only to advance authentication policy. Expression or other value specifying the next policy to be evaluated if the current policy evaluates to TRUE. Specify one of the following values: * NEXT - Evaluate the policy with the next higher priority number. * END - End policy evaluation. * USE_INVOCATION_RESULT - Applicable if this policy invokes another policy label. If the final goto in the invoked policy label has a value of END,

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
			the evaluation stops. If the final goto is anything other than END, the current policy label performs a NEXT. * An expression that evaluates to a number. If you specify an expression, the number to which it evaluates determines the next policy to evaluate, as follows: * If the expression evaluates to a higher numbered priority, the policy with that priority is evaluated next. * If the expression evaluates to the priority of the current policy, the policy with the next higher numbered priority is evaluated next. * If the expression evaluates to a priority number that is numerically higher than the highest numbered priority, policy evaluation ends. An UNDEF event is triggered if: * The expression is invalid. * The expression evaluates to a

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
			priority number that is numerically lower than the current policy's priority. * The expression evaluates to a priority number that is between the current policy's priority number (say, 30) and the highest priority number (say, 100), but does not match any configured priority number (for example, the expression evaluates to the number 85). This example assumes that the priority number increments by 10 for every successive policy, and therefore a priority number of 85 does not exist in the policy label.
secondary	<Boolean>	Read-write	Bind the authentication policy to the secondary chain. Provides for multifactor authentication in which a user must authenticate via both a primary authentication method and, afterward, via a secondary

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
			authentication method. Because user groups are aggregated across authentication systems, usernames must be the same on all authentication servers. Passwords can be different.
policy	<String>	Read-write	The name of the policy, if any, bound to the authentication vserver.
groupextraction	<Boolean>	Read-write	Bind the Authentication policy to a tertiary chain which will be used only for group extraction. The user will not authenticate against this server, and this will only be called if primary and/or secondary authentication has succeeded.
acttype	<Double>	Read-only	.
__count	<Double>	Read-only	count parameter

Operations

(click to see Properties)

ADD:| DELETE:| GET| GET (ALL)| COUNT

Some options that you can use for each operations:

- **Getting warnings in response:**NITRO allows you to get warnings in an operation by specifying the "warning" query parameter as "yes". For example, to get warnings while connecting to the Citrix ADC appliance, the URL is as follows:

  http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/login?warning=yes

  If any, the warnings are displayed in the response payload with the HTTP code "209 X-NITRO-WARNING".

- **Authenticated access for individual NITRO operations:**NITRO allows you to logon to the Citrix ADC appliance to perform individual operations. You can use this option instead of creating a NITRO session (using the login object) and then using that session to perform all operations,

  To do this, you must specify the username and password in the request header of the NITRO request as follows:

  X-NITRO-USER:<username>

  X-NITRO-PASS:<password>

  Note: In such cases, make sure that the request header DOES not include the following:

  Cookie:NITRO_AUTH_TOKEN=<tokenvalue>

`*Note:`*

Mandatory parameters are marked in redand placeholder content is marked in <green>.

add

URL:http://<Citrix-ADC-IP-address(NSIP)/nitro/v1/config/authenticationvserver_authenticationldappolicy_binding

HTTP Method:PUT

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Content-Type:application/json

Request Payload:

```
{
```

```
"authenticationvserver_authenticationldappolicy_binding":{

<b>"name":<String_value>,

</b>"policy":<String_value>,

"priority":<Double_value>,

"secondary":<Boolean_value>,

"groupextraction":<Boolean_value>,

"nextfactor":<String_value>,

"gotopriorityexpression":<String_value>

}}
```

Response:

HTTP Status Code on Success: 201 Created HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error

delete

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/authenticationvserver_authenticationldappolicy_binding/name_value<String>

Query-parameters:

args

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/authenticationvserver_authenticationldappolicy_binding/name_value<String>?**args=policy:<String_value>,secondary:<Boolean_value>,groupextraction:<Boolean_value>**

HTTP Method:DELETE

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue>

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error

get

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/authenticationvserver_authenticationldappolicy_binding/name_value<String>

Query-parameters:

filter

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/authenticationvserver_authenticationldappolicy_binding/name_value<String>?**filter=property-name1:property-value1,property-name2:property-value2**

Use this query-parameter to get the filtered set of authenticationvserver_authenticationldappolicy_binding resources configured on Citrix ADC. Filtering can be done on any of the properties of the resource.

pagination

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/authenticationvserver_authenticationldappolicy_binding/name_value<String>?**pagesize=#no;pageno=#no**

Use this query-parameter to get the authenticationvserver_authenticationldappolicy_binding resources in chunks.

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "authenticationvserver_authenticationldappolicy_binding": [ {

"priority":<Double_value>,

"name":<String_value>,

"nextfactor":<String_value>,

"gotopriorityexpression":<String_value>,

"secondary":<Boolean_value>,

"policy":<String_value>,

"groupextraction":<Boolean_value>,

"acttype":<Double_value>

}]}
```

get (all)

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/authenticationvserver_authenticationldappolicy_binding

Query-parameters:

bulkbindings

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/authenticationvserver_authenticationldappolicy_binding?**bulkbindings=yes**

NITRO allows you to fetch bindings in bulk.

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "authenticationvserver_authenticationldappolicy_binding": [ {

"priority":<Double_value>,

"name":<String_value>,

"nextfactor":<String_value>,

"gotopriorityexpression":<String_value>,

"secondary":<Boolean_value>,

"policy":<String_value>,

"groupextraction":<Boolean_value>,

"acttype":<Double_value>

}]}
```

count

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/authenticationvserver_authenticationldappolicy_binding/name_value<String>?**count=yes**

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{"authenticationvserver_authenticationldappolicy_binding": [ { "__count":
"#no"} ] }
```

authenticationvserver_authenticationlocalpolicy_binding

Binding object showing the authenticationlocalpolicy that can be bound to authenticationvserver.

Properties

(click to see Operations)

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
priority	<Double>	Read-write	The priority, if any, of the vpn vserver policy.
name	<String>	Read-write	Name of the authentication virtual server to which to bind the policy. Minimum length = 1
nextfactor	<String>	Read-write	Applicable only while binding advance authentication policy as classic authentication policy does not support nFactor.

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
gotopriorityexpression	<String>	Read-write	Applicable only to advance authentication policy. Expression or other value specifying the next policy to be evaluated if the current policy evaluates to TRUE. Specify one of the following values: * NEXT - Evaluate the policy with the next higher priority number. * END - End policy evaluation. * USE_INVOCATION_RESULT - Applicable if this policy invokes another policy label. If the final goto in the invoked policy label has a value of END, the evaluation stops. If the final goto is anything other than END, the current policy label performs a NEXT. * An expression that evaluates to a number. If you specify an expression, the number to which it evaluates determines the next policy to evaluate, as follows: * If the expression evaluates to a higher

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
			numbered priority, the policy with that priority is evaluated next. * If the expression evaluates to the priority of the current policy, the policy with the next higher numbered priority is evaluated next. * If the expression evaluates to a priority number that is numerically higher than the highest numbered priority, policy evaluation ends. An UNDEF event is triggered if: * The expression is invalid. * The expression evaluates to a priority number that is numerically lower than the current policy's priority. * The expression evaluates to a priority number that is between the current policy's priority number (say, 30) and the highest priority number (say, 100), but does not match any configured priority number (for example, the expression evaluates to the number 85).

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
			This example assumes that the priority number increments by 10 for every successive policy, and therefore a priority number of 85 does not exist in the policy label.
secondary	<Boolean>	Read-write	Bind the authentication policy to the secondary chain. Provides for multifactor authentication in which a user must authenticate via both a primary authentication method and, afterward, via a secondary authentication method. Because user groups are aggregated across authentication systems, usernames must be the same on all authentication servers. Passwords can be different.
policy	<String>	Read-write	The name of the policy, if any, bound to the authentication vserver.
groupextraction	<Boolean>	Read-write	Applicable only while bindind classic

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
			authentication policy as advance authentication policy use nFactor.
acttype	<Double>	Read-only	.
__count	<Double>	Read-only	count parameter

Operations

(click to see Properties)

ADD:| DELETE:| GET| GET (ALL)| COUNT

Some options that you can use for each operations:

- **Getting warnings in response:**NITRO allows you to get warnings in an operation by specifying the "warning" query parameter as "yes". For example, to get warnings while connecting to the Citrix ADC appliance, the URL is as follows:

  http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/login?warning=yes

  If any, the warnings are displayed in the response payload with the HTTP code "209 X-NITRO-WARNING".

- **Authenticated access for individual NITRO operations:**NITRO allows you to logon to the Citrix ADC appliance to perform individual operations. You can use this option instead of creating a NITRO session (using the login object) and then using that session to perform all operations,

  To do this, you must specify the username and password in the request header of the NITRO request as follows:

  X-NITRO-USER:<username>

  X-NITRO-PASS:<password>

  Note: In such cases, make sure that the request header DOES not include the following:

  Cookie:NITRO_AUTH_TOKEN=<tokenvalue>

`*Note:` *

Mandatory parameters are marked in redand placeholder content is marked in <green>.

add

URL:http://<Citrix-ADC-IP-address(NSIP)/nitro/v1/config/authenticationvserver_authenticationlocalpolicy_binding

HTTP Method:PUT

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Content-Type:application/json

Request Payload:

```
{

"authenticationvserver_authenticationlocalpolicy_binding":{

<b>"name":<String_value>,

</b>"policy":<String_value>,

"priority":<Double_value>,

"secondary":<Boolean_value>,

"groupextraction":<Boolean_value>,

"nextfactor":<String_value>,

"gotopriorityexpression":<String_value>

}}
```

Response:

HTTP Status Code on Success: 201 Created HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error

delete

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/authenticationvserver_authenticationlocalpolicy_binding/name_value<String>

Query-parameters:

args

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/authenticationvserver_authenticationlocalpolicy_binding/name_value<String>?**args=policy:<String_value>,secondary:<Boolean_value>,groupextraction:<Boolean_value>**

HTTP Method:DELETE

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue>

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error

get

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/authenticationvserver_authenticationlocalpolicy_binding/name_value<String>

Query-parameters:

filter

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/authenticationvserver_authenticationlocalpolicy_binding/name_value<String>?**filter=property-name1:property-value1,property-name2:property-value2**

Use this query-parameter to get the filtered set of authenticationvserver_authenticationlocalpolicy_binding resources configured on Citrix ADC. Filtering can be done on any of the properties of the resource.

pagination

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/authenticationvserver_authenticationlocalpolicy_binding/name_value<String>?**pagesize=#no;pageno=#no**

Use this query-parameter to get the authenticationvserver_authenticationlocalpolicy_binding resources in chunks.

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "authenticationvserver_authenticationlocalpolicy_binding": [ {

"priority":<Double_value>,

"name":<String_value>,

"nextfactor":<String_value>,

"gotopriorityexpression":<String_value>,

"secondary":<Boolean_value>,

"policy":<String_value>,

"groupextraction":<Boolean_value>,

"acttype":<Double_value>

}]}
```

get (all)

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/authenticationvserver_authenticationlocalpolicy_binding

Query-parameters:

bulkbindings

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/authenticationvserver_authenticationlocalpolicy_binding?**bulkbindings=yes**

NITRO allows you to fetch bindings in bulk.

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "authenticationvserver_authenticationlocalpolicy_binding": [ {

"priority":<Double_value>,

"name":<String_value>,

"nextfactor":<String_value>,

"gotopriorityexpression":<String_value>,

"secondary":<Boolean_value>,
```

```
"policy":<String_value>,

"groupextraction":<Boolean_value>,

"acttype":<Double_value>

}]}
```

count

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/authenticationvserver_authenticationlocalpolicy_binding/name_value<String>?**count=yes**

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{"authenticationvserver_authenticationlocalpolicy_binding": [ { "__count":
"#no"} ] }
```

authenticationvserver_authenticationloginschemapolicy_binding

Binding object showing the authenticationloginschemapolicy that can be bound to authenticationvserver.

Properties

(click to see Operations)

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
priority	<Double>	Read-write	The priority, if any, of the vpn vserver policy.
name	<String>	Read-write	Name of the authentication virtual server to which to bind the policy. Minimum length = 1
nextfactor	<String>	Read-write	Applicable only while binding advance authentication policy as classic authentication policy does not support nFactor.
gotopriorityexpression	<String>	Read-write	Applicable only to advance authentication policy. Expression or other value specifying the next policy to be evaluated if the current policy evaluates to TRUE. Specify one of the following values: * NEXT - Evaluate the policy with the next higher priority number. * END - End policy evaluation. * USE_INVOCATION_RESULT - Applicable if this policy invokes

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
			another policy label. If the final goto in the invoked policy label has a value of END, the evaluation stops. If the final goto is anything other than END, the current policy label performs a NEXT. * An expression that evaluates to a number. If you specify an expression, the number to which it evaluates determines the next policy to evaluate, as follows: * If the expression evaluates to a higher numbered priority, the policy with that priority is evaluated next. * If the expression evaluates to the priority of the current policy, the policy with the next higher numbered priority is evaluated next. * If the expression evaluates to a priority number that is numerically higher than the highest numbered priority, policy evaluation ends. An UNDEF event is

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
			triggered if: * The expression is invalid. * The expression evaluates to a priority number that is numerically lower than the current policy's priority. * The expression evaluates to a priority number that is between the current policy's priority number (say, 30) and the highest priority number (say, 100), but does not match any configured priority number (for example, the expression evaluates to the number 85). This example assumes that the priority number increments by 10 for every successive policy, and therefore a priority number of 85 does not exist in the policy label.
secondary	<Boolean>	Read-write	Applicable only while bindind classic authentication policy as advance authentication policy use nFactor.
policy	<String>	Read-write	The name of the policy, if any, bound

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
			to the authentication vserver.
groupextraction	<Boolean>	Read-write	Applicable only while bindind classic authentication policy as advance authentication policy use nFactor.
acttype	<Double>	Read-only	.
__count	<Double>	Read-only	count parameter

Operations

(click to see Properties)

ADD:| DELETE:| GET| GET (ALL)| COUNT

Some options that you can use for each operations:

- **Getting warnings in response:**NITRO allows you to get warnings in an operation by specifying the "warning" query parameter as "yes". For example, to get warnings while connecting to the Citrix ADC appliance, the URL is as follows:

  http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/login?warning=yes

  If any, the warnings are displayed in the response payload with the HTTP code "209 X-NITRO-WARNING".
- **Authenticated access for individual NITRO operations:**NITRO allows you to logon to the Citrix ADC appliance to perform individual operations. You can use this option instead of creating a NITRO session (using the login object) and then using that session to perform all operations,

  To do this, you must specify the username and password in the request header of the NITRO request as follows:

  X-NITRO-USER:<username>

  X-NITRO-PASS:<password>

Note: In such cases, make sure that the request header DOES not include the following:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue>

`*Note:` *

Mandatory parameters are marked in redand placeholder content is marked in <green>.

add

URL:http://<Citrix-ADC-IP-address(NSIP)/nitro/v1/config/authenticationvserver_authenticationloginschemapolicy_binding

HTTP Method:PUT

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Content-Type:application/json

Request Payload:

```
{
"authenticationvserver_authenticationloginschemapolicy_binding":{
<b>"name":<String_value>,
</b>"policy":<String_value>,
"priority":<Double_value>,
"secondary":<Boolean_value>,
"groupextraction":<Boolean_value>,
"nextfactor":<String_value>,
"gotopriorityexpression":<String_value>
}}
```

Response:

HTTP Status Code on Success: 201 Created HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error

delete

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/authenticationvserver_authenticationloginschemapolicy_binding/name_value<String>

Query-parameters:

args

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/authenticationvserver_authenticationloginschemapolicy_binding/name_value<String>?**args=policy:<String_value>,secondary:<Boolean_value>,groupextraction:<Boolean_value>**

HTTP Method:DELETE

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue>

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error

get

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/authenticationvserver_authenticationloginschemapolicy_binding/name_value<String>

Query-parameters:

filter

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/authenticationvserver_authenticationloginschemapolicy_binding/name_value<String>?**filter=property-name1:property-value1,property-name2:property-value2**

Use this query-parameter to get the filtered set of authenticationvserver_authenticationloginschemapolicy_binding resources configured on Citrix ADC. Filtering can be done on any of the properties of the resource.

pagination

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/authenticationvserver_authenticationloginschemapolicy_binding/name_value<String>?**pagesize=#no;pageno=#no**

Use this query-parameter to get the authenticationvserver_authenticationloginschemapolicy_binding resources in chunks.

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "authenticationvserver_authenticationloginschemapolicy_binding": [ {

"priority":<Double_value>,

"name":<String_value>,

"nextfactor":<String_value>,

"gotopriorityexpression":<String_value>,

"secondary":<Boolean_value>,

"policy":<String_value>,

"groupextraction":<Boolean_value>,

"acttype":<Double_value>
```

```
}]}
```

get (all)

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/authenticationvserver_authenticationloginschemapolicy_binding

Query-parameters:

bulkbindings

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/authenticationvserver_authenticationloginschemapolicy_binding?**bulkbindings=yes**

NITRO allows you to fetch bindings in bulk.

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "authenticationvserver_authenticationloginschemapolicy_binding": [ {

"priority":<Double_value>,

"name":<String_value>,

"nextfactor":<String_value>,

"gotopriorityexpression":<String_value>,
```

```
"secondary":<Boolean_value>,

"policy":<String_value>,

"groupextraction":<Boolean_value>,

"acttype":<Double_value>

}]}
```

count

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/authenticationvserver_authenticationloginschemapolicy_binding/name_value<String>?**count=yes**

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{"authenticationvserver_authenticationloginschemapolicy_binding":
[ { "__count": "#no"} ] }
```

authenticationvserver_authenticationnegotiatepolicy_binding

Binding object showing the authenticationnegotiatepolicy that can be bound to authenticationvserver.

Properties

(click to see Operations)

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
priority	<Double>	Read-write	The priority, if any, of the vpn vserver policy.
name	<String>	Read-write	Name of the authentication virtual server to which to bind the policy. Minimum length = 1
nextfactor	<String>	Read-write	Applicable only while binding advance authentication policy as classic authentication policy does not support nFactor.
gotopriorityexpression	<String>	Read-write	Applicable only to advance authentication policy. Expression or other value specifying the next policy to be evaluated if the current policy evaluates to TRUE. Specify one of the following values: * NEXT - Evaluate the policy with the next higher priority number. * END - End policy evaluation. * USE_INVOCATION_RESULT - Applicable if this policy invokes

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
			another policy label. If the final goto in the invoked policy label has a value of END, the evaluation stops. If the final goto is anything other than END, the current policy label performs a NEXT. * An expression that evaluates to a number. If you specify an expression, the number to which it evaluates determines the next policy to evaluate, as follows: * If the expression evaluates to a higher numbered priority, the policy with that priority is evaluated next. * If the expression evaluates to the priority of the current policy, the policy with the next higher numbered priority is evaluated next. * If the expression evaluates to a priority number that is numerically higher than the highest numbered priority, policy evaluation ends. An UNDEF event is

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
			triggered if: * The expression is invalid. * The expression evaluates to a priority number that is numerically lower than the current policy's priority. * The expression evaluates to a priority number that is between the current policy's priority number (say, 30) and the highest priority number (say, 100), but does not match any configured priority number (for example, the expression evaluates to the number 85). This example assumes that the priority number increments by 10 for every successive policy, and therefore a priority number of 85 does not exist in the policy label.
secondary	<Boolean>	Read-write	Bind the authentication policy to the secondary chain. Provides for multifactor authentication in which a user must authenticate via both a primary

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
			authentication method and, afterward, via a secondary authentication method. Because user groups are aggregated across authentication systems, usernames must be the same on all authentication servers. Passwords can be different.
policy	<String>	Read-write	The name of the policy, if any, bound to the authentication vserver.
groupextraction	<Boolean>	Read-write	Applicable only while bindind classic authentication policy as advance authentication policy use nFactor.
acttype	<Double>	Read-only	.
__count	<Double>	Read-only	count parameter

Operations

(click to see Properties)

ADD:| DELETE:| GET| GET (ALL)| COUNT

Some options that you can use for each operations:

- **Getting warnings in response:**NITRO allows you to get warnings in an operation by specifying the "warning" query parameter as "yes". For example, to get warnings while connecting to the Citrix ADC appliance, the URL is as follows:

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/login?warning=yes

If any, the warnings are displayed in the response payload with the HTTP code "209 X-NITRO-WARNING".

- **Authenticated access for individual NITRO operations:**NITRO allows you to logon to the Citrix ADC appliance to perform individual operations. You can use this option instead of creating a NITRO session (using the login object) and then using that session to perform all operations,

  To do this, you must specify the username and password in the request header of the NITRO request as follows:

  X-NITRO-USER:<username>

  X-NITRO-PASS:<password>

  Note: In such cases, make sure that the request header DOES not include the following:

  Cookie:NITRO_AUTH_TOKEN=<tokenvalue>

`*Note:`*

Mandatory parameters are marked in redand placeholder content is marked in <green>.

add

URL:http://<Citrix-ADC-IP-address(NSIP)/nitro/v1/config/authenticationvserver_authenticationnegotiatepolicy_binding

HTTP Method:PUT

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Content-Type:application/json

Request Payload:

```
{

"authenticationvserver_authenticationnegotiatepolicy_binding":{

<b>"name":<String_value>,
```

```
</b>"policy":<String_value>,

"priority":<Double_value>,

"secondary":<Boolean_value>,

"groupextraction":<Boolean_value>,

"nextfactor":<String_value>,

"gotopriorityexpression":<String_value>

}}
```

Response:

HTTP Status Code on Success: 201 Created HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error

delete

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/authenticationvserver_authenticationnegotiatepolicy_binding/name_value<String>

Query-parameters:

args

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/authenticationvserver_authenticationnegotiatepolicy_binding/name_value<String>?**args=policy:<String_value>,secondary:<Boolean_value>,groupextraction:<Boolean_value>**

HTTP Method:DELETE

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue>

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error

get

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/authenticationvserver_authenticationnegotiatepolicy_binding/name_value<String>

Query-parameters:

filter

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/authenticationvserver_authenticationnegotiatepolicy_binding/name_value<String>?**filter=property-name1:property-value1,property-name2:property-value2**

Use this query-parameter to get the filtered set of authenticationvserver_authenticationnegotiatepolicy_binding resources configured on Citrix ADC. Filtering can be done on any of the properties of the resource.

pagination

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/authenticationvserver_authenticationnegotiatepolicy_binding/name_value<String>?**pagesize=#no;pageno=#no**

Use this query-parameter to get the authenticationvserver_authenticationnegotiatepolicy_binding resources in chunks.

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "authenticationvserver_authenticationnegotiatepolicy_binding": [ {

"priority":<Double_value>,

"name":<String_value>,

"nextfactor":<String_value>,

"gotopriorityexpression":<String_value>,

"secondary":<Boolean_value>,

"policy":<String_value>,

"groupextraction":<Boolean_value>,

"acttype":<Double_value>

}]}
```

get (all)

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/authenticationvserver_authenticationnegotiatepolicy_binding

Query-parameters:

bulkbindings

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/authenticationvserver_authenticationnegotiatepolicy_binding?**bulkbindings=yes**

NITRO allows you to fetch bindings in bulk.

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "authenticationvserver_authenticationnegotiatepolicy_binding": [ {

"priority":<Double_value>,

"name":<String_value>,

"nextfactor":<String_value>,

"gotopriorityexpression":<String_value>,

"secondary":<Boolean_value>,

"policy":<String_value>,

"groupextraction":<Boolean_value>,

"acttype":<Double_value>

}]}
```

count

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/authenticationvserver_authenticationnegotiatepolicy_binding/name_value<String>?**count=yes**

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{"authenticationvserver_authenticationnegotiatepolicy_binding":
[ { "__count": "#no"} ] }
```

authenticationvserver_authenticationpolicy_binding

Binding object showing the authenticationpolicy that can be bound to authenticationvserver.

Properties

(click to see Operations)

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
priority	<Double>	Read-write	The priority, if any, of the vpn vserver policy.
name	<String>	Read-write	Name of the authentication virtual server to which to bind the policy. Minimum length = 1
nextfactor	<String>	Read-write	On success invoke label.
gotopriorityexpression	<String>	Read-write	Expression specifying the priority of the next policy which will get evaluated if the

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
			current policy rule evaluates to TRUE.
secondary	<Boolean>	Read-write	Applicable only while bindind classic authentication policy as advance authentication policy use nFactor.
policy	<String>	Read-write	The name of the policy, if any, bound to the authentication vserver.
groupextraction	<Boolean>	Read-write	Applicable only while bindind classic authentication policy as advance authentication policy use nFactor.
acttype	<Double>	Read-only	.
__count	<Double>	Read-only	count parameter

Operations

(click to see Properties)

ADD:| DELETE:| GET| GET (ALL)| COUNT

Some options that you can use for each operations:

- **Getting warnings in response:**NITRO allows you to get warnings in an operation by specifying the "warning" query parameter as "yes". For example, to get warnings while connecting to the Citrix ADC appliance, the URL is as follows:

  http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/login?warning=yes

  If any, the warnings are displayed in the response payload with the HTTP code "209 X-NITRO-WARNING".

- **Authenticated access for individual NITRO operations:**NITRO allows you to logon to the Citrix ADC appliance to perform individual operations. You can use this option instead of creating a NITRO session (using the login object) and then using that session to perform all operations,

  To do this, you must specify the username and password in the request header of the NITRO request as follows:

  X-NITRO-USER:<username>

  X-NITRO-PASS:<password>

  Note: In such cases, make sure that the request header DOES not include the following:

  Cookie:NITRO_AUTH_TOKEN=<tokenvalue>

`*Note:`*

Mandatory parameters are marked in redand placeholder content is marked in <green>.

add

URL:http://<Citrix-ADC-IP-address(NSIP)/nitro/v1/config/authenticationvserver_authenticationpolicy_binding

HTTP Method:PUT

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Content-Type:application/json

Request Payload:

```
{

"authenticationvserver_authenticationpolicy_binding":{

<b>"name":<String_value>,

</b>"policy":<String_value>,

"priority":<Double_value>,
```

```
"secondary":<Boolean_value>,

"groupextraction":<Boolean_value>,

"nextfactor":<String_value>,

"gotopriorityexpression":<String_value>

}}
```

Response:

HTTP Status Code on Success: 201 Created HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error

delete

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/authenticationvserver_authenticationpolicy_binding/name_value<String>

Query-parameters:

args

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/authenticationvserver_authenticationpolicy_binding/name_value<String>?**args=policy:<String_value>,secondary:<Boolean_value>,groupextraction:<Boolean_value>**

HTTP Method:DELETE

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue>

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error

get

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/authenticationvserver_authenticationpolicy_binding/name_value<String>

Query-parameters:

filter

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/authenticationvserver_authenticationpolicy_binding/name_value<String>?**filter=property-name1:property-value1,property-name2:property-value2**

Use this query-parameter to get the filtered set of authenticationvserver_authenticationpolicy_binding resources configured on Citrix ADC. Filtering can be done on any of the properties of the resource.

pagination

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/authenticationvserver_authenticationpolicy_binding/name_value<String>?**pagesize=#no;pageno=#no**

Use this query-parameter to get the authenticationvserver_authenticationpolicy_binding resources in chunks.

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "authenticationvserver_authenticationpolicy_binding": [ {

"priority":<Double_value>,

"name":<String_value>,

"nextfactor":<String_value>,

"gotopriorityexpression":<String_value>,

"secondary":<Boolean_value>,

"policy":<String_value>,

"groupextraction":<Boolean_value>,

"acttype":<Double_value>

}]}
```

get (all)

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/authenticationvserver_authenticationpolicy_binding

Query-parameters:

bulkbindings

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/authenticationvserver_authenticationpolicy_binding?**bulkbindings=yes**

NITRO allows you to fetch bindings in bulk.

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "authenticationvserver_authenticationpolicy_binding": [ {

"priority":<Double_value>,

"name":<String_value>,

"nextfactor":<String_value>,

"gotopriorityexpression":<String_value>,

"secondary":<Boolean_value>,

"policy":<String_value>,

"groupextraction":<Boolean_value>,

"acttype":<Double_value>

}]}
```

count

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/authenticationvserver_authenticationpolicy_binding/name_value<String>?**count=yes**

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{"authenticationvserver_authenticationpolicy_binding": [ { "__count":
"#no"} ] }
```

authenticationvserver_authenticationradiuspolicy_binding

Binding object showing the authenticationradiuspolicy that can be bound to authenticationvserver.

Properties

(click to see Operations)

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
priority	<Double>	Read-write	The priority, if any, of the vpn vserver policy.
name	<String>	Read-write	Name of the authentication virtual server to which to bind the policy. Minimum length = 1
nextfactor	<String>	Read-write	Applicable only while binding advance authentication policy as classic authentication policy does not support nFactor.

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
gotopriorityexpression	<String>	Read-write	Applicable only to advance authentication policy. Expression or other value specifying the next policy to be evaluated if the current policy evaluates to TRUE. Specify one of the following values: * NEXT - Evaluate the policy with the next higher priority number. * END - End policy evaluation. * USE_INVOCATION_RESULT - Applicable if this policy invokes another policy label. If the final goto in the invoked policy label has a value of END, the evaluation stops. If the final goto is anything other than END, the current policy label performs a NEXT. * An expression that evaluates to a number. If you specify an expression, the number to which it evaluates determines the next policy to evaluate, as follows: * If the expression evaluates to a higher

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
			numbered priority, the policy with that priority is evaluated next. * If the expression evaluates to the priority of the current policy, the policy with the next higher numbered priority is evaluated next. * If the expression evaluates to a priority number that is numerically higher than the highest numbered priority, policy evaluation ends. An UNDEF event is triggered if: * The expression is invalid. * The expression evaluates to a priority number that is numerically lower than the current policy's priority. * The expression evaluates to a priority number that is between the current policy's priority number (say, 30) and the highest priority number (say, 100), but does not match any configured priority number (for example, the expression evaluates to the number 85).

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
			This example assumes that the priority number increments by 10 for every successive policy, and therefore a priority number of 85 does not exist in the policy label.
secondary	<Boolean>	Read-write	Bind the authentication policy to the secondary chain. Provides for multifactor authentication in which a user must authenticate via both a primary authentication method and, afterward, via a secondary authentication method. Because user groups are aggregated across authentication systems, usernames must be the same on all authentication servers. Passwords can be different.
policy	<String>	Read-write	The name of the policy, if any, bound to the authentication vserver.
groupextraction	<Boolean>	Read-write	Applicable only while bindind classic

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
			authentication policy as advance authentication policy use nFactor.
acttype	<Double>	Read-only	.
__count	<Double>	Read-only	count parameter

Operations

(click to see Properties)

ADD:| DELETE:| GET| GET (ALL)| COUNT

Some options that you can use for each operations:

- **Getting warnings in response:**NITRO allows you to get warnings in an operation by specifying the "warning" query parameter as "yes". For example, to get warnings while connecting to the Citrix ADC appliance, the URL is as follows:

  http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/login?warning=yes

  If any, the warnings are displayed in the response payload with the HTTP code "209 X-NITRO-WARNING".

- **Authenticated access for individual NITRO operations:**NITRO allows you to logon to the Citrix ADC appliance to perform individual operations. You can use this option instead of creating a NITRO session (using the login object) and then using that session to perform all operations,

  To do this, you must specify the username and password in the request header of the NITRO request as follows:

  X-NITRO-USER:<username>

  X-NITRO-PASS:<password>

  Note: In such cases, make sure that the request header DOES not include the following:

  Cookie:NITRO_AUTH_TOKEN=<tokenvalue>

`*Note:` *

Mandatory parameters are marked in redand placeholder content is marked in <green>.

add

URL:http://<Citrix-ADC-IP-address(NSIP)/nitro/v1/config/authenticationvserver_authenticationradiuspolicy_binding

HTTP Method:PUT

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Content-Type:application/json

Request Payload:

```
{

"authenticationvserver_authenticationradiuspolicy_binding":{

<b>"name":<String_value>,

</b>"policy":<String_value>,

"priority":<Double_value>,

"secondary":<Boolean_value>,

"groupextraction":<Boolean_value>,

"nextfactor":<String_value>,

"gotopriorityexpression":<String_value>

}}
```

Response:

HTTP Status Code on Success: 201 Created HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error

delete

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/authenticationvserver_authenticationradiuspolicy_binding/name_value<String>

Query-parameters:

args

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/authenticationvserver_authenticationradiuspolicy_binding/name_value<String>?**args=policy:<String_value>,secondary:<Boolean_value>,groupextraction:<Boolean_value>**

HTTP Method:DELETE

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue>

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error

get

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/authenticationvserver_authenticationradiuspolicy_binding/name_value<String>

Query-parameters:

filter

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/authenticationvserver_authenticationradiuspolicy_binding/name_value<String>?**filter=property-name1:property-value1,property-name2:property-value2**

Use this query-parameter to get the filtered set of authenticationvserver_authenticationradiuspolicy_binding resources configured on Citrix ADC. Filtering can be done on any of the properties of the resource.

pagination

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/authenticationvserver_authenticationradiuspolicy_binding/name_value<String>?**pagesize=#no;pageno=#no**

Use this query-parameter to get the authenticationvserver_authenticationradiuspolicy_binding resources in chunks.

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "authenticationvserver_authenticationradiuspolicy_binding": [ {

"priority":<Double_value>,

"name":<String_value>,

"nextfactor":<String_value>,

"gotopriorityexpression":<String_value>,

"secondary":<Boolean_value>,

"policy":<String_value>,

"groupextraction":<Boolean_value>,

"acttype":<Double_value>

}]}
```

get (all)

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/authenticationvserver_authenticationradiuspolicy_binding

Query-parameters:

bulkbindings

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/authenticationvserver_authenticationradiuspolicy_binding?**bulkbindings=yes**

NITRO allows you to fetch bindings in bulk.

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "authenticationvserver_authenticationradiuspolicy_binding": [ {

"priority":<Double_value>,

"name":<String_value>,

"nextfactor":<String_value>,

"gotopriorityexpression":<String_value>,

"secondary":<Boolean_value>,
```

```
"policy":<String_value>,

"groupextraction":<Boolean_value>,

"acttype":<Double_value>

}]}
```

count

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/authenticationvserver_authenticationradiuspolicy_binding/name_value<String>?**count=yes**

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{"authenticationvserver_authenticationradiuspolicy_binding":
[ { "__count": "#no"} ] }
```

authenticationvserver_authenticationsamlidppolicy_binding

Binding object showing the authenticationsamlidppolicy that can be bound to authenticationvserver.

Properties

(click to see Operations)

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
priority	<Double>	Read-write	The priority, if any, of the vpn vserver policy.
name	<String>	Read-write	Name of the authentication virtual server to which to bind the policy. Minimum length = 1
nextfactor	<String>	Read-write	Applicable only while binding advance authentication policy as classic authentication policy does not support nFactor.
gotopriorityexpression	<String>	Read-write	Applicable only to advance authentication policy. Expression or other value specifying the next policy to be evaluated if the current policy evaluates to TRUE. Specify one of the following values: * NEXT - Evaluate the policy with the next higher priority number. * END - End policy evaluation. * USE_INVOCATION_RESULT - Applicable if this policy invokes another policy label. If the final goto in the invoked policy label has a value of END,

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
			the evaluation stops. If the final goto is anything other than END, the current policy label performs a NEXT. * An expression that evaluates to a number. If you specify an expression, the number to which it evaluates determines the next policy to evaluate, as follows: * If the expression evaluates to a higher numbered priority, the policy with that priority is evaluated next. * If the expression evaluates to the priority of the current policy, the policy with the next higher numbered priority is evaluated next. * If the expression evaluates to a priority number that is numerically higher than the highest numbered priority, policy evaluation ends. An UNDEF event is triggered if: * The expression is invalid. * The expression evaluates to a

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
			priority number that is numerically lower than the current policy's priority. * The expression evaluates to a priority number that is between the current policy's priority number (say, 30) and the highest priority number (say, 100), but does not match any configured priority number (for example, the expression evaluates to the number 85). This example assumes that the priority number increments by 10 for every successive policy, and therefore a priority number of 85 does not exist in the policy label.
secondary	<Boolean>	Read-write	Applicable only while bindind classic authentication policy as advance authentication policy use nFactor.
policy	<String>	Read-write	The name of the policy, if any, bound to the authentication vserver.

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
groupextraction	<Boolean>	Read-write	Applicable only while bindind classic authentication policy as advance authentication policy use nFactor.
acttype	<Double>	Read-only	.
__count	<Double>	Read-only	count parameter

Operations

(click to see Properties)

ADD:| DELETE:| GET| GET (ALL)| COUNT

Some options that you can use for each operations:

- **Getting warnings in response:**NITRO allows you to get warnings in an operation by specifying the "warning" query parameter as "yes". For example, to get warnings while connecting to the Citrix ADC appliance, the URL is as follows:

  http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/login?warning=yes

  If any, the warnings are displayed in the response payload with the HTTP code "209 X-NITRO-WARNING".

- **Authenticated access for individual NITRO operations:**NITRO allows you to logon to the Citrix ADC appliance to perform individual operations. You can use this option instead of creating a NITRO session (using the login object) and then using that session to perform all operations,

  To do this, you must specify the username and password in the request header of the NITRO request as follows:

  X-NITRO-USER:<username>

  X-NITRO-PASS:<password>

  Note: In such cases, make sure that the request header DOES not include the following:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue>

`*Note:`*

Mandatory parameters are marked in redand placeholder content is marked in <green>.

add

URL:http://<Citrix-ADC-IP-address(NSIP)/nitro/v1/config/authenticationvserver_authenticationsamlidppolicy_binding

HTTP Method:PUT

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Content-Type:application/json

Request Payload:

```
{

"authenticationvserver_authenticationsamlidppolicy_binding":{

<b>"name":<String_value>,

</b>"policy":<String_value>,

"priority":<Double_value>,

"secondary":<Boolean_value>,

"groupextraction":<Boolean_value>,

"nextfactor":<String_value>,

"gotopriorityexpression":<String_value>

}}
```

Response:

HTTP Status Code on Success: 201 Created HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error

delete

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/authenticationvserver_authenticationsamlidppolicy_binding/name_value<String>

Query-parameters:

args

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/authenticationvserver_authenticationsamlidppolicy_binding/name_value<String>?**args=policy:<String_value>,secondary:<Boolean_value>,groupextraction:<Boolean_value>**

HTTP Method:DELETE

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue>

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error

get

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/authenticationvserver_authenticationsamlidppolicy_binding/name_value<String>

Query-parameters:

filter

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/authenticationvserver_authenticationsamlidppolicy_binding/name_value<String>?**filter=property-name1:property-value1,property-name2:property-value2**

Use this query-parameter to get the filtered set of authenticationvserver_authenticationsamlidppolicy_binding resources configured on Citrix ADC. Filtering can be done on any of the properties of the resource.

pagination

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/authenticationvserver_authenticationsamlidppolicy_binding/name_value<String>?**pagesize=#no;pageno=#no**

Use this query-parameter to get the authenticationvserver_authenticationsamlidppolicy_binding resources in chunks.

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "authenticationvserver_authenticationsamlidppolicy_binding": [ {

"priority":<Double_value>,

"name":<String_value>,

"nextfactor":<String_value>,

"gotopriorityexpression":<String_value>,

"secondary":<Boolean_value>,

"policy":<String_value>,

"groupextraction":<Boolean_value>,

"acttype":<Double_value>
```

```
}]}
```

get (all)

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/authenticationvserver_authenticationsamlidppolicy_binding

Query-parameters:

bulkbindings

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/authenticationvserver_authenticationsamlidppolicy_binding?**bulkbindings=yes**

NITRO allows you to fetch bindings in bulk.

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "authenticationvserver_authenticationsamlidppolicy_binding": [ {

"priority":<Double_value>,

"name":<String_value>,

"nextfactor":<String_value>,

"gotopriorityexpression":<String_value>,
```

```
"secondary":<Boolean_value>,

"policy":<String_value>,

"groupextraction":<Boolean_value>,

"acttype":<Double_value>

}]}
```

count

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/authenticationvserver_authenticationsamlidppolicy_binding/name_value<String>?**count=yes**

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{"authenticationvserver_authenticationsamlidppolicy_binding":
[ { "__count": "#no"} ] }
```

authenticationvserver_authenticationsamlpolicy_binding

Binding object showing the authenticationsamlpolicy that can be bound to authenticationvserver.

Properties

(click to see Operations)

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
priority	<Double>	Read-write	The priority, if any, of the vpn vserver policy.
name	<String>	Read-write	Name of the authentication virtual server to which to bind the policy. Minimum length = 1
nextfactor	<String>	Read-write	Applicable only while binding advance authentication policy as classic authentication policy does not support nFactor.
gotopriorityexpression	<String>	Read-write	Applicable only to advance authentication policy. Expression or other value specifying the next policy to be evaluated if the current policy evaluates to TRUE. Specify one of the following values: * NEXT - Evaluate the policy with the next higher priority number. * END - End policy evaluation. * USE_INVOCATION_RESULT - Applicable if this policy invokes

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
			another policy label. If the final goto in the invoked policy label has a value of END, the evaluation stops. If the final goto is anything other than END, the current policy label performs a NEXT. * An expression that evaluates to a number. If you specify an expression, the number to which it evaluates determines the next policy to evaluate, as follows: * If the expression evaluates to a higher numbered priority, the policy with that priority is evaluated next. * If the expression evaluates to the priority of the current policy, the policy with the next higher numbered priority is evaluated next. * If the expression evaluates to a priority number that is numerically higher than the highest numbered priority, policy evaluation ends. An UNDEF event is

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
			triggered if: * The expression is invalid. * The expression evaluates to a priority number that is numerically lower than the current policy's priority. * The expression evaluates to a priority number that is between the current policy's priority number (say, 30) and the highest priority number (say, 100), but does not match any configured priority number (for example, the expression evaluates to the number 85). This example assumes that the priority number increments by 10 for every successive policy, and therefore a priority number of 85 does not exist in the policy label.
secondary	<Boolean>	Read-write	Bind the authentication policy to the secondary chain. Provides for multifactor authentication in which a user must authenticate via both a primary

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
			authentication method and, afterward, via a secondary authentication method. Because user groups are aggregated across authentication systems, usernames must be the same on all authentication servers. Passwords can be different.
policy	<String>	Read-write	The name of the policy, if any, bound to the authentication vserver.
groupextraction	<Boolean>	Read-write	Applicable only while bindind classic authentication policy as advance authentication policy use nFactor.
acttype	<Double>	Read-only	.
__count	<Double>	Read-only	count parameter

Operations

(click to see Properties)

ADD:| DELETE:| GET| GET (ALL)| COUNT

Some options that you can use for each operations:

- **Getting warnings in response:**NITRO allows you to get warnings in an operation by specifying the "warning" query parameter as "yes". For example, to get warnings while connecting to the Citrix ADC appliance, the URL is as follows:

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/login?warning=yes

If any, the warnings are displayed in the response payload with the HTTP code "209 X-NITRO-WARNING".

- **Authenticated access for individual NITRO operations:**NITRO allows you to logon to the Citrix ADC appliance to perform individual operations. You can use this option instead of creating a NITRO session (using the login object) and then using that session to perform all operations,

  To do this, you must specify the username and password in the request header of the NITRO request as follows:

  X-NITRO-USER:<username>

  X-NITRO-PASS:<password>

  Note: In such cases, make sure that the request header DOES not include the following:

  Cookie:NITRO_AUTH_TOKEN=<tokenvalue>

`*Note:`*

Mandatory parameters are marked in redand placeholder content is marked in <green>.

add

URL:http://<Citrix-ADC-IP-address(NSIP)/nitro/v1/config/authenticationvserver_authenticationsamlpolicy_binding

HTTP Method:PUT

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Content-Type:application/json

Request Payload:

```
{

"authenticationvserver_authenticationsamlpolicy_binding":{

<b>"name":<String_value>,
```

```
</b>"policy":<String_value>,

"priority":<Double_value>,

"secondary":<Boolean_value>,

"groupextraction":<Boolean_value>,

"nextfactor":<String_value>,

"gotopriorityexpression":<String_value>

}}
```

Response:

HTTP Status Code on Success: 201 Created HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error

delete

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/authenticationvserver_authenticationsamlpolicy_binding/name_value<String>

Query-parameters:

args

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/authenticationvserver_authenticationsamlpolicy_binding/name_value<String>?**args=policy:<String_value>,secondary:<Boolean_value>,groupextraction:<Boolean_value>**

HTTP Method:DELETE

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue>

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error

get

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/authenticationvserver_authenticationsamlpolicy_binding/name_value<String>

Query-parameters:

filter

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/authenticationvserver_authenticationsamlpolicy_binding/name_value<String>?**filter=property-name1:property-value1,property-name2:property-value2**

Use this query-parameter to get the filtered set of authenticationvserver_authenticationsamlpolicy_binding resources configured on Citrix ADC. Filtering can be done on any of the properties of the resource.

pagination

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/authenticationvserver_authenticationsamlpolicy_binding/name_value<String>?**pagesize=#no;pageno=#no**

Use this query-parameter to get the authenticationvserver_authenticationsamlpolicy_binding resources in chunks.

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "authenticationvserver_authenticationsamlpolicy_binding": [ {

"priority":<Double_value>,

"name":<String_value>,

"nextfactor":<String_value>,

"gotopriorityexpression":<String_value>,

"secondary":<Boolean_value>,

"policy":<String_value>,

"groupextraction":<Boolean_value>,

"acttype":<Double_value>

}]}
```

get (all)

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/authenticationvserver_authenticationsamlpolicy_binding

Query-parameters:

bulkbindings

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/authenticationvserver_authenticationsamlpolicy_binding?**bulkbindings=yes**

NITRO allows you to fetch bindings in bulk.

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "authenticationvserver_authenticationsamlpolicy_binding": [ {

"priority":<Double_value>,

"name":<String_value>,

"nextfactor":<String_value>,

"gotopriorityexpression":<String_value>,

"secondary":<Boolean_value>,

"policy":<String_value>,

"groupextraction":<Boolean_value>,

"acttype":<Double_value>

}]}
```

count

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/authenticationvserver_authenticationsamlpolicy_binding/name_value<String>?**count=yes**

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{"authenticationvserver_authenticationsamlpolicy_binding": [ { "__count":
"#no"} ] }
```

authenticationvserver_authenticationtacacspolicy_binding

Binding object showing the authenticationtacacspolicy that can be bound to authenticationvserver.

Properties

(click to see Operations)

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
priority	<Double>	Read-write	The priority, if any, of the vpn vserver policy.
name	<String>	Read-write	Name of the authentication virtual server to which to bind the policy. Minimum length = 1
nextfactor	<String>	Read-write	Applicable only while binding advance authentication policy as classic authentication policy does not support nFactor.

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
gotopriorityexpression	<String>	Read-write	Applicable only to advance authentication policy. Expression or other value specifying the next policy to be evaluated if the current policy evaluates to TRUE. Specify one of the following values: * NEXT - Evaluate the policy with the next higher priority number. * END - End policy evaluation. * USE_INVOCATION_RESULT - Applicable if this policy invokes another policy label. If the final goto in the invoked policy label has a value of END, the evaluation stops. If the final goto is anything other than END, the current policy label performs a NEXT. * An expression that evaluates to a number. If you specify an expression, the number to which it evaluates determines the next policy to evaluate, as follows: * If the expression evaluates to a higher

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
			numbered priority, the policy with that priority is evaluated next. * If the expression evaluates to the priority of the current policy, the policy with the next higher numbered priority is evaluated next. * If the expression evaluates to a priority number that is numerically higher than the highest numbered priority, policy evaluation ends. An UNDEF event is triggered if: * The expression is invalid. * The expression evaluates to a priority number that is numerically lower than the current policy's priority. * The expression evaluates to a priority number that is between the current policy's priority number (say, 30) and the highest priority number (say, 100), but does not match any configured priority number (for example, the expression evaluates to the number 85).

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
			This example assumes that the priority number increments by 10 for every successive policy, and therefore a priority number of 85 does not exist in the policy label.
secondary	<Boolean>	Read-write	Bind the authentication policy to the secondary chain. Provides for multifactor authentication in which a user must authenticate via both a primary authentication method and, afterward, via a secondary authentication method. Because user groups are aggregated across authentication systems, usernames must be the same on all authentication servers. Passwords can be different.
policy	<String>	Read-write	The name of the policy, if any, bound to the authentication vserver.
groupextraction	<Boolean>	Read-write	Applicable only while bindind classic

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
			authentication policy as advance authentication policy use nFactor.
acttype	<Double>	Read-only	.
__count	<Double>	Read-only	count parameter

Operations

(click to see Properties)

ADD:| DELETE:| GET| GET (ALL)| COUNT

Some options that you can use for each operations:

- **Getting warnings in response:**NITRO allows you to get warnings in an operation by specifying the "warning" query parameter as "yes". For example, to get warnings while connecting to the Citrix ADC appliance, the URL is as follows:

  http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/login?warning=yes

  If any, the warnings are displayed in the response payload with the HTTP code "209 X-NITRO-WARNING".

- **Authenticated access for individual NITRO operations:**NITRO allows you to logon to the Citrix ADC appliance to perform individual operations. You can use this option instead of creating a NITRO session (using the login object) and then using that session to perform all operations,

  To do this, you must specify the username and password in the request header of the NITRO request as follows:

  X-NITRO-USER:<username>

  X-NITRO-PASS:<password>

  Note: In such cases, make sure that the request header DOES not include the following:

  Cookie:NITRO_AUTH_TOKEN=<tokenvalue>

`*Note:` *

Mandatory parameters are marked in redand placeholder content is marked in <green>.

add

URL:http://<Citrix-ADC-IP-address(NSIP)/nitro/v1/config/authenticationvserver_authenticationtacacspolicy_binding

HTTP Method:PUT

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Content-Type:application/json

Request Payload:

```
{

"authenticationvserver_authenticationtacacspolicy_binding":{

<b>"name":<String_value>,

</b>"policy":<String_value>,

"priority":<Double_value>,

"secondary":<Boolean_value>,

"groupextraction":<Boolean_value>,

"nextfactor":<String_value>,

"gotopriorityexpression":<String_value>

}}
```

Response:

HTTP Status Code on Success: 201 Created HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error

delete

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/authenticationvserver_authenticationtacacspolicy_binding/name_value<String>

Query-parameters:

args

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/authenticationvserver_authenticationtacacspolicy_binding/name_value<String>?**args=policy:<String_value>,secondary:<Boolean_value>,groupextraction:<Boolean_value>**

HTTP Method:DELETE

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue>

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error

get

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/authenticationvserver_authenticationtacacspolicy_binding/name_value<String>

Query-parameters:

filter

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/authenticationvserver_authenticationtacacspolicy_binding/name_value<String>?**filter=property-name1:property-value1,property-name2:property-value2**

Use this query-parameter to get the filtered set of authenticationvserver_authenticationtacacspolicy_binding resources configured on Citrix ADC. Filtering can be done on any of the properties of the resource.

pagination

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/authenticationvserver_authenticationtacacspolicy_binding/name_value<String>?**pagesize=#no;pageno=#no**

Use this query-parameter to get the authenticationvserver_authenticationtacacspolicy_binding resources in chunks.

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "authenticationvserver_authenticationtacacspolicy_binding": [ {

"priority":<Double_value>,

"name":<String_value>,

"nextfactor":<String_value>,

"gotopriorityexpression":<String_value>,

"secondary":<Boolean_value>,

"policy":<String_value>,

"groupextraction":<Boolean_value>,

"acttype":<Double_value>

}]}
```

get (all)

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/authenticationvserver_authenticationtacacspolicy_binding

Query-parameters:

bulkbindings

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/authenticationvserver_authenticationtacacspolicy_binding?**bulkbindings=yes**

NITRO allows you to fetch bindings in bulk.

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "authenticationvserver_authenticationtacacspolicy_binding": [ {

"priority":<Double_value>,

"name":<String_value>,

"nextfactor":<String_value>,

"gotopriorityexpression":<String_value>,

"secondary":<Boolean_value>,
```

```
"policy":<String_value>,

"groupextraction":<Boolean_value>,

"acttype":<Double_value>

}]}
```

count

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/authenticationvserver_authenticationtacacspolicy_binding/name_value<String>?**count=yes**

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{"authenticationvserver_authenticationtacacspolicy_binding":
[ { "__count": "#no"} ] }
```

authenticationvserver_authenticationwebauthpolicy_binding

Binding object showing the authenticationwebauthpolicy that can be bound to authenticationvserver.

Properties

(click to see Operations)

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
priority	<Double>	Read-write	The priority, if any, of the vpn vserver policy.
name	<String>	Read-write	Name of the authentication virtual server to which to bind the policy. Minimum length = 1
nextfactor	<String>	Read-write	Applicable only while binding advance authentication policy as classic authentication policy does not support nFactor.
gotopriorityexpression	<String>	Read-write	Applicable only to advance authentication policy. Expression or other value specifying the next policy to be evaluated if the current policy evaluates to TRUE. Specify one of the following values: * NEXT - Evaluate the policy with the next higher priority number. * END - End policy evaluation. * USE_INVOCATION_RESULT - Applicable if this policy invokes

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
			another policy label. If the final goto in the invoked policy label has a value of END, the evaluation stops. If the final goto is anything other than END, the current policy label performs a NEXT. * An expression that evaluates to a number. If you specify an expression, the number to which it evaluates determines the next policy to evaluate, as follows: * If the expression evaluates to a higher numbered priority, the policy with that priority is evaluated next. * If the expression evaluates to the priority of the current policy, the policy with the next higher numbered priority is evaluated next. * If the expression evaluates to a priority number that is numerically higher than the highest numbered priority, policy evaluation ends. An UNDEF event is

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
			triggered if: * The expression is invalid. * The expression evaluates to a priority number that is numerically lower than the current policy's priority. * The expression evaluates to a priority number that is between the current policy's priority number (say, 30) and the highest priority number (say, 100), but does not match any configured priority number (for example, the expression evaluates to the number 85). This example assumes that the priority number increments by 10 for every successive policy, and therefore a priority number of 85 does not exist in the policy label.
secondary	<Boolean>	Read-write	Bind the authentication policy to the secondary chain. Provides for multifactor authentication in which a user must authenticate via both a primary

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
			authentication method and, afterward, via a secondary authentication method. Because user groups are aggregated across authentication systems, usernames must be the same on all authentication servers. Passwords can be different.
policy	<String>	Read-write	The name of the policy, if any, bound to the authentication vserver.
groupextraction	<Boolean>	Read-write	Applicable only while bindind classic authentication policy as advance authentication policy use nFactor.
acttype	<Double>	Read-only	.
__count	<Double>	Read-only	count parameter

Operations

(click to see Properties)

ADD:| DELETE:| GET| GET (ALL)| COUNT

Some options that you can use for each operations:

- **Getting warnings in response:**NITRO allows you to get warnings in an operation by specifying the "warning" query parameter as "yes". For example, to get warnings while connecting to the Citrix ADC appliance, the URL is as follows:

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/login?warning=yes

If any, the warnings are displayed in the response payload with the HTTP code "209 X-NITRO-WARNING".

- **Authenticated access for individual NITRO operations:**NITRO allows you to logon to the Citrix ADC appliance to perform individual operations. You can use this option instead of creating a NITRO session (using the login object) and then using that session to perform all operations,

  To do this, you must specify the username and password in the request header of the NITRO request as follows:

  X-NITRO-USER:<username>

  X-NITRO-PASS:<password>

  Note: In such cases, make sure that the request header DOES not include the following:

  Cookie:NITRO_AUTH_TOKEN=<tokenvalue>

`*Note:`*

Mandatory parameters are marked in redand placeholder content is marked in <green>.

add

URL:http://<Citrix-ADC-IP-address(NSIP)/nitro/v1/config/authenticationvserver_authenticationwebauthpolicy_binding

HTTP Method:PUT

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Content-Type:application/json

Request Payload:

```
{

"authenticationvserver_authenticationwebauthpolicy_binding":{

<b>"name":<String_value>,
```

```
</b>"policy":<String_value>,

"priority":<Double_value>,

"secondary":<Boolean_value>,

"groupextraction":<Boolean_value>,

"nextfactor":<String_value>,

"gotopriorityexpression":<String_value>

}}
```

Response:

HTTP Status Code on Success: 201 Created HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error

delete

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/authenticationvserver_authenticationwebauthpolicy_binding/name_value<String>

Query-parameters:

args

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/authenticationvserver_authenticationwebauthpolicy_binding/name_value<String>?**args=policy:<String_value>,secondary:<Boolean_value>,groupextraction:<Boolean_value>**

HTTP Method:DELETE

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue>

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error

get

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/authenticationvserver_authenticationwebauthpolicy_binding/name_value<String>

Query-parameters:

filter

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/authenticationvserver_authenticationwebauthpolicy_binding/name_value<String>?**filter=property-name1:property-value1,property-name2:property-value2**

Use this query-parameter to get the filtered set of authenticationvserver_authenticationwebauthpolicy_binding resources configured on Citrix ADC. Filtering can be done on any of the properties of the resource.

pagination

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/authenticationvserver_authenticationwebauthpolicy_binding/name_value<String>?**pagesize=#no;pageno=#no**

Use this query-parameter to get the authenticationvserver_authenticationwebauthpolicy_binding resources in chunks.

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "authenticationvserver_authenticationwebauthpolicy_binding": [ {

"priority":<Double_value>,

"name":<String_value>,

"nextfactor":<String_value>,

"gotopriorityexpression":<String_value>,

"secondary":<Boolean_value>,

"policy":<String_value>,

"groupextraction":<Boolean_value>,

"acttype":<Double_value>

}]}
```

get (all)

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/authenticationvserver_authenticationwebauthpolicy_binding

Query-parameters:

bulkbindings

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/authenticationvserver_authenticationwebauthpolicy_binding?**bulkbindings=yes**

NITRO allows you to fetch bindings in bulk.

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "authenticationvserver_authenticationwebauthpolicy_binding": [ {

"priority":<Double_value>,

"name":<String_value>,

"nextfactor":<String_value>,

"gotopriorityexpression":<String_value>,

"secondary":<Boolean_value>,

"policy":<String_value>,

"groupextraction":<Boolean_value>,

"acttype":<Double_value>

}]}
```

count

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/authenticationvserver_authenticationwebauthpolicy_binding/name_value<String>?**count=yes**

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{"authenticationvserver_authenticationwebauthpolicy_binding":
[ { "__count": "#no"} ] }
```

authenticationvserver_binding

Binding object which returns the resources bound to authenticationvserver.

Properties

(click to see Operations)

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
name	<String>	Read-write	Name of the authentication virtual server. Minimum length = 1
authenticationvserver_auditnslogpolicy_binding	<authenticationvserver_auditnslogpolicy_binding[]>	Read-only	auditnslogpolicy that can be bound to authenticationvserver.
authenticationvserver_authenticationpolicy_binding	<authenticationvserver_authenticationpolicy_binding[]>	Read-only	authenticationpolicy that can be bound to authenticationvserver.
authenticationvserver_authenticationradiuspolicy_binding	<authenticationvserver_authenticationradiuspolicy_binding[]>	Read-only	authenticationradiuspolicy that can be bound to

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
			authenticationvserver.
authenticationvserver_authenticationldappolicy_binding	<authenticationvserver_authenticationldappolicy_binding[]>	Read-only	authenticationldappolicy that can be bound to authenticationvserver.
authenticationvserver_authenticationsamlidppolicy_binding	<authenticationvserver_authenticationsamlidppolicy_binding[]>	Read-only	authenticationsamlidppolicy that can be bound to authenticationvserver.
authenticationvserver_authenticationwebauthpolicy_binding	<authenticationvserver_authenticationwebauthpolicy_binding[]>	Read-only	authenticationwebauthpolicy that can be bound to authenticationvserver.
authenticationvserver_authenticationlocalpolicy_binding	<authenticationvserver_authenticationlocalpolicy_binding[]>	Read-only	authenticationlocalpolicy that can be bound to authenticationvserver.
authenticationvserver_authenticationnegotiatepolicy_binding	<authenticationvserver_authenticationnegotiatepolicy_binding[]>	Read-only	authenticationnegotiatepolicy that can be bound to authenticationvserver.
authenticationvserver_authenticationtacacspolicy_binding	<authenticationvserver_authenticationtacacspolicy_binding[]>	Read-only	authenticationtacacspolicy that can be bound to authenticationvserver.

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
authenticationvserver_vpnportaltheme_binding	<authenticationvserver_vpnportaltheme_binding[]>	Read-only	vpnportaltheme that can be bound to authenticationvserver.
authenticationvserver_authenticationsamlpolicy_binding	<authenticationvserver_authenticationsamlpolicy_binding[]>	Read-only	authenticationsamlpolicy that can be bound to authenticationvserver.
authenticationvserver_cspolicy_binding	<authenticationvserver_cspolicy_binding[]>	Read-only	cspolicy that can be bound to authenticationvserver.
authenticationvserver_auditsyslogpolicy_binding	<authenticationvserver_auditsyslogpolicy_binding[]>	Read-only	auditsyslogpolicy that can be bound to authenticationvserver.
authenticationvserver_authenticationloginschemapolicy_binding	<authenticationvserver_authenticationloginschemapolicy_binding[]>	Read-only	authenticationloginschemapolicy that can be bound to authenticationvserver.
authenticationvserver_authenticationcertpolicy_binding	<authenticationvserver_authenticationcertpolicy_binding[]>	Read-only	authenticationcertpolicy that can be bound to authenticationvserver.
authenticationvserver_tmsessionpolicy_binding	<authenticationvserver_tmsessionpolicy_binding[]>	Read-only	tmsessionpolicy that can be bound to authenticationvserver.
authenticationvserver_authenticationoauthidppolicy_binding	<authenticationvserver_authenticationoau	Read-only	authenticationoauthidppolicy that can be bound to

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
	thidppolicy_binding[]>		authenticationvserver.

Operations

(click to see Properties)

GET| GET (ALL)

Some options that you can use for each operations:

- **Getting warnings in response:**NITRO allows you to get warnings in an operation by specifying the "warning" query parameter as "yes". For example, to get warnings while connecting to the Citrix ADC appliance, the URL is as follows:

  http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/login?warning=yes

  If any, the warnings are displayed in the response payload with the HTTP code "209 X-NITRO-WARNING".

- **Authenticated access for individual NITRO operations:**NITRO allows you to logon to the Citrix ADC appliance to perform individual operations. You can use this option instead of creating a NITRO session (using the login object) and then using that session to perform all operations,

  To do this, you must specify the username and password in the request header of the NITRO request as follows:

  X-NITRO-USER:<username>

  X-NITRO-PASS:<password>

  Note: In such cases, make sure that the request header DOES not include the following:

  Cookie:NITRO_AUTH_TOKEN=<tokenvalue>

`*Note:`*

Mandatory parameters are marked in redand placeholder content is marked in <green>.

get

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/authenticationvserver_binding/name_value<String>

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "authenticationvserver_binding": [ {

"name":<String_value>,

"authenticationvserver_authenticationlocalpolicy_binding":<authenticationv
server_authenticationlocalpolicy_binding[]_value>,

"authenticationvserver_tmsessionpolicy_binding":<authenticationvserver_tms
essionpolicy_binding[]_value>,

"authenticationvserver_cspolicy_binding":<authenticationvserver_cspolicy_b
inding[]_value>,

"authenticationvserver_auditsyslogpolicy_binding":<authenticationvserver_a
uditsyslogpolicy_binding[]_value>,

"authenticationvserver_authenticationsamlidppolicy_binding":<authenticatio
nvserver_authenticationsamlidppolicy_binding[]_value>,

"authenticationvserver_auditnslogpolicy_binding":<authenticationvserver_au
ditnslogpolicy_binding[]_value>,

"authenticationvserver_authenticationldappolicy_binding":<authenticationvs
erver_authenticationldappolicy_binding[]_value>,

"authenticationvserver_authenticationcertpolicy_binding":<authenticationvs
```

```
erver_authenticationcertpolicy_binding[]_value>,

"authenticationvserver_authenticationsamlpolicy_binding":<authenticationvs
erver_authenticationsamlpolicy_binding[]_value>,

"authenticationvserver_authenticationpolicy_binding":<authenticationvserve
r_authenticationpolicy_binding[]_value>,

"authenticationvserver_vpnportaltheme_binding":<authenticationvserver_vpnp
ortaltheme_binding[]_value>,

"authenticationvserver_authenticationoauthidppolicy_binding":<authenticati
onvserver_authenticationoauthidppolicy_binding[]_value>,

"authenticationvserver_authenticationradiuspolicy_binding":<authentication
vserver_authenticationradiuspolicy_binding[]_value>,

"authenticationvserver_authenticationloginschemapolicy_binding":<authentic
ationvserver_authenticationloginschemapolicy_binding[]_value>,

"authenticationvserver_authenticationtacacspolicy_binding":<authentication
vserver_authenticationtacacspolicy_binding[]_value>,

"authenticationvserver_authenticationwebauthpolicy_binding":<authenticatio
nvserver_authenticationwebauthpolicy_binding[]_value>,

"authenticationvserver_authenticationnegotiatepolicy_binding":<authenticat
ionvserver_authenticationnegotiatepolicy_binding[]_value>

}]}
```

get (all)

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/authenticationvserver_binding

Query-parameters:

bulkbindings

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/authenticationvserver_binding?
bulkbindings=yes

NITRO allows you to fetch bindings in bulk.

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "authenticationvserver_binding": [ {

"name":<String_value>,

"authenticationvserver_authenticationlocalpolicy_binding":<authenticationv
server_authenticationlocalpolicy_binding[]_value>,

"authenticationvserver_tmsessionpolicy_binding":<authenticationvserver_tms
essionpolicy_binding[]_value>,

"authenticationvserver_cspolicy_binding":<authenticationvserver_cspolicy_b
inding[]_value>,

"authenticationvserver_auditsyslogpolicy_binding":<authenticationvserver_a
uditsyslogpolicy_binding[]_value>,

"authenticationvserver_authenticationsamlidppolicy_binding":<authenticatio
nvserver_authenticationsamlidppolicy_binding[]_value>,

"authenticationvserver_auditnslogpolicy_binding":<authenticationvserver_au
ditnslogpolicy_binding[]_value>,

"authenticationvserver_authenticationldappolicy_binding":<authenticationvs
erver_authenticationldappolicy_binding[]_value>,

"authenticationvserver_authenticationcertpolicy_binding":<authenticationvs
```

```
erver_authenticationcertpolicy_binding[]_value>,

"authenticationvserver_authenticationsamlpolicy_binding":<authenticationvs
erver_authenticationsamlpolicy_binding[]_value>,

"authenticationvserver_authenticationpolicy_binding":<authenticationvserve
r_authenticationpolicy_binding[]_value>,

"authenticationvserver_vpnportaltheme_binding":<authenticationvserver_vpnp
ortaltheme_binding[]_value>,

"authenticationvserver_authenticationoauthidppolicy_binding":<authenticati
onvserver_authenticationoauthidppolicy_binding[]_value>,

"authenticationvserver_authenticationradiuspolicy_binding":<authentication
vserver_authenticationradiuspolicy_binding[]_value>,

"authenticationvserver_authenticationloginschemapolicy_binding":<authentic
ationvserver_authenticationloginschemapolicy_binding[]_value>,

"authenticationvserver_authenticationtacacspolicy_binding":<authentication
vserver_authenticationtacacspolicy_binding[]_value>,

"authenticationvserver_authenticationwebauthpolicy_binding":<authenticatio
nvserver_authenticationwebauthpolicy_binding[]_value>,

"authenticationvserver_authenticationnegotiatepolicy_binding":<authenticat
ionvserver_authenticationnegotiatepolicy_binding[]_value>

}]}
```

authenticationvserver_cspolicy_binding

Binding object showing the cspolicy that can be bound to authenticationvserver.

Properties

(click to see Operations)

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
priority	<Double>	Read-write	The priority, if any, of the vpn vserver policy.
name	<String>	Read-write	Name of the authentication virtual server to which to bind the policy. Minimum length = 1
nextfactor	<String>	Read-write	Applicable only while binding advance authentication policy as classic authentication policy does not support nFactor.
gotopriorityexpression	<String>	Read-write	Expression specifying the priority of the next policy which will get evaluated if the current policy rule evaluates to TRUE.
secondary	<Boolean>	Read-write	Applicable only while bindind classic authentication policy as advance authentication policy use nFactor.
policy	<String>	Read-write	The name of the policy, if any, bound to the authentication vserver.
groupextraction	<Boolean>	Read-write	Applicable only while bindind classic authentication policy

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
			as advance authentication policy use nFactor.
__count	<Double>	Read-write	count parameter

Operations

(click to see Properties)

ADD:| DELETE:| GET| GET (ALL)| COUNT

Some options that you can use for each operations:

- **Getting warnings in response:**NITRO allows you to get warnings in an operation by specifying the "warning" query parameter as "yes". For example, to get warnings while connecting to the Citrix ADC appliance, the URL is as follows:

  http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/login?warning=yes

  If any, the warnings are displayed in the response payload with the HTTP code "209 X-NITRO-WARNING".

- **Authenticated access for individual NITRO operations:**NITRO allows you to logon to the Citrix ADC appliance to perform individual operations. You can use this option instead of creating a NITRO session (using the login object) and then using that session to perform all operations,

  To do this, you must specify the username and password in the request header of the NITRO request as follows:

  X-NITRO-USER:<username>

  X-NITRO-PASS:<password>

  Note: In such cases, make sure that the request header DOES not include the following:

  Cookie:NITRO_AUTH_TOKEN=<tokenvalue>

`*Note:`*

Mandatory parameters are marked in redand placeholder content is marked in <green>.

add

URL:http://<Citrix-ADC-IP-address(NSIP)/nitro/v1/config/authenticationvserver_cspolicy_binding

HTTP Method:PUT

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Content-Type:application/json

Request Payload:

```
{

"authenticationvserver_cspolicy_binding":{

<b>"name":<String_value>,

</b>"policy":<String_value>,

"priority":<Double_value>,

"secondary":<Boolean_value>,

"groupextraction":<Boolean_value>,

"nextfactor":<String_value>,

"gotopriorityexpression":<String_value>

}}
```

Response:

HTTP Status Code on Success: 201 Created HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error

delete

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/authenticationvserver_cspolicy_binding/name_value<String>

Query-parameters:

args

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/authenticationvserver_cspolicy_binding/name_value<String>?**args=policy:<String_value>,secondary:<Boolean_value>,groupextraction:<Boolean_value>**

HTTP Method:DELETE

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue>

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error

get

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/authenticationvserver_cspolicy_binding/name_value<String>

Query-parameters:

filter

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/authenticationvserver_cspolicy_binding/name_value<String>?**filter=property-name1:property-value1,property-name2:property-value2**

Use this query-parameter to get the filtered set of authenticationvserver_cspolicy_binding resources configured on Citrix ADC. Filtering can be done on any of the properties of the resource.

pagination

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/authenticationvserver_cspolicy_binding/name_value<String>?**pagesize=#no;pageno=#no**

Use this query-parameter to get the authenticationvserver_cspolicy_binding resources in chunks.

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "authenticationvserver_cspolicy_binding": [ {

"priority":<Double_value>,

"name":<String_value>,

"nextfactor":<String_value>,

"gotopriorityexpression":<String_value>,

"secondary":<Boolean_value>,

"policy":<String_value>,

"groupextraction":<Boolean_value>

}]}
```

get (all)

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/authenticationvserver_cspolicy_binding

Query-parameters:

bulkbindings

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/authenticationvserver_cspolicy_binding?**bulkbindings=yes**

NITRO allows you to fetch bindings in bulk.

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "authenticationvserver_cspolicy_binding": [ {

"priority":<Double_value>,

"name":<String_value>,

"nextfactor":<String_value>,

"gotopriorityexpression":<String_value>,

"secondary":<Boolean_value>,

"policy":<String_value>,

"groupextraction":<Boolean_value>

}]}
```

count

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/authenticationvserver_cspolicy_binding/name_value<String>?**count=yes**

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{"authenticationvserver_cspolicy_binding": [ { "__count": "#no"} ] }
```

authenticationvserver_tmsessionpolicy_binding

Binding object showing the tmsessionpolicy that can be bound to authenticationvserver.

Properties

(click to see Operations)

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
priority	<Double>	Read-write	The priority, if any, of the vpn vserver policy.
name	<String>	Read-write	Name of the authentication virtual server to which to bind the policy.

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
			Minimum length = 1
nextfactor	<String>	Read-write	Applicable only while binding advance authentication policy as classic authentication policy does not support nFactor.
gotopriorityexpression	<String>	Read-write	Expression specifying the priority of the next policy which will get evaluated if the current policy rule evaluates to TRUE.
secondary	<Boolean>	Read-write	Applicable only while bindind classic authentication policy as advance authentication policy use nFactor.
policy	<String>	Read-write	The name of the policy, if any, bound to the authentication vserver.
groupextraction	<Boolean>	Read-write	Applicable only while bindind classic authentication policy as advance authentication policy use nFactor.
__count	<Double>	Read-write	count parameter

Operations

(click to see Properties)

ADD:| DELETE:| GET| GET (ALL)| COUNT

Some options that you can use for each operations:

- **Getting warnings in response:**NITRO allows you to get warnings in an operation by specifying the "warning" query parameter as "yes". For example, to get warnings while connecting to the Citrix ADC appliance, the URL is as follows:

  http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/login?warning=yes

  If any, the warnings are displayed in the response payload with the HTTP code "209 X-NITRO-WARNING".

- **Authenticated access for individual NITRO operations:**NITRO allows you to logon to the Citrix ADC appliance to perform individual operations. You can use this option instead of creating a NITRO session (using the login object) and then using that session to perform all operations,

  To do this, you must specify the username and password in the request header of the NITRO request as follows:

  X-NITRO-USER:<username>

  X-NITRO-PASS:<password>

  Note: In such cases, make sure that the request header DOES not include the following:

  Cookie:NITRO_AUTH_TOKEN=<tokenvalue>

`*Note:`*

Mandatory parameters are marked in redand placeholder content is marked in <green>.

add

URL:http://<Citrix-ADC-IP-address(NSIP)/nitro/v1/config/authenticationvserver_tmsessionpolicy_binding

HTTP Method:PUT

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Content-Type:application/json

Request Payload:

```
{

"authenticationvserver_tmsessionpolicy_binding":{

<b>"name":<String_value>,

</b>"policy":<String_value>,

"priority":<Double_value>,

"secondary":<Boolean_value>,

"groupextraction":<Boolean_value>,

"nextfactor":<String_value>,

"gotopriorityexpression":<String_value>

}}
```

Response:

HTTP Status Code on Success: 201 Created HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error

delete

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/authenticationvserver_tmsessionpolicy_binding/name_value<String>

Query-parameters:

args

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/authenticationvserver_tmsessionpolicy_binding/name_value<String>?**args=policy:<String_value>,secondary:<Boolean_value>,groupextraction:<Boolean_value>**

HTTP Method:DELETE

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue>

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error

get

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/authenticationvserver_tmsessionpolicy_binding/name_value<String>

Query-parameters:

filter

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/authenticationvserver_tmsessionpolicy_binding/name_value<String>?**filter=property-name1:property-value1,property-name2:property-value2**

Use this query-parameter to get the filtered set of authenticationvserver_tmsessionpolicy_binding resources configured on Citrix ADC. Filtering can be done on any of the properties of the resource.

pagination

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/authenticationvserver_tmsessionpolicy_binding/name_value<String>?**pagesize=#no;pageno=#no**

Use this query-parameter to get the authenticationvserver_tmsessionpolicy_binding resources in chunks.

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "authenticationvserver_tmsessionpolicy_binding": [ {

"priority":<Double_value>,

"name":<String_value>,

"nextfactor":<String_value>,

"gotopriorityexpression":<String_value>,

"secondary":<Boolean_value>,

"policy":<String_value>,

"groupextraction":<Boolean_value>

}]}
```

get (all)

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/authenticationvserver_tmsessionpolicy_binding

Query-parameters:

bulkbindings

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/authenticationvserver_tmsessionpolicy_binding?**bulkbindings=yes**

NITRO allows you to fetch bindings in bulk.

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "authenticationvserver_tmsessionpolicy_binding": [ {

"priority":<Double_value>,

"name":<String_value>,

"nextfactor":<String_value>,

"gotopriorityexpression":<String_value>,

"secondary":<Boolean_value>,

"policy":<String_value>,

"groupextraction":<Boolean_value>

}]}
```

count

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/authenticationvserver_tmsessionpolicy_binding/name_value<String>?**count=yes**

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{"authenticationvserver_tmsessionpolicy_binding": [ { "__count":
"#no"} ] }
```

authenticationvserver_vpnportaltheme_binding

Binding object showing the vpnportaltheme that can be bound to authenticationvserver.

Properties

(click to see Operations)

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
name	<String>	Read-write	Name of the authentication virtual server to which to bind the policy. Minimum length = 1
portaltheme	<String>	Read-write	Theme for Authentication virtual server Login portal.
acttype	<Double>	Read-only	.
__count	<Double>	Read-only	count parameter

Operations

(click to see Properties)

ADD:| DELETE:| GET| GET (ALL)| COUNT

Some options that you can use for each operations:

- **Getting warnings in response:**NITRO allows you to get warnings in an operation by specifying the "warning" query parameter as "yes". For example, to get warnings while connecting to the Citrix ADC appliance, the URL is as follows:

  http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/login?warning=yes

  If any, the warnings are displayed in the response payload with the HTTP code "209 X-NITRO-WARNING".

- **Authenticated access for individual NITRO operations:**NITRO allows you to logon to the Citrix ADC appliance to perform individual operations. You can use this option instead of creating a NITRO session (using the login object) and then using that session to perform all operations,

  To do this, you must specify the username and password in the request header of the NITRO request as follows:

  X-NITRO-USER:<username>

  X-NITRO-PASS:<password>

  Note: In such cases, make sure that the request header DOES not include the following:

  Cookie:NITRO_AUTH_TOKEN=<tokenvalue>

`*Note:`*

Mandatory parameters are marked in redand placeholder content is marked in <green>.

add

URL:http://<Citrix-ADC-IP-address(NSIP)/nitro/v1/config/authenticationvserver_vpnportaltheme_binding

HTTP Method:PUT

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Content-Type:application/json

Request Payload:

```
{

"authenticationvserver_vpnportaltheme_binding":{

<b>"name":<String_value>,

</b>"portaltheme":<String_value>

}}
```

Response:

HTTP Status Code on Success: 201 Created HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error

delete

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/authenticationvserver_vpnportaltheme_binding/name_value<String>

Query-parameters:

args

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/authenticationvserver_vpnportaltheme_binding/name_value<String>?**args=portaltheme:<String_value>**

HTTP Method:DELETE

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue>

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error

get

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/authenticationvserver_vpnportaltheme_binding/name_value<String>

Query-parameters:

filter

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/authenticationvserver_vpnportaltheme_binding/name_value<String>?**filter=property-name1:property-value1,property-name2:property-value2**

Use this query-parameter to get the filtered set of authenticationvserver_vpnportaltheme_binding resources configured on Citrix ADC. Filtering can be done on any of the properties of the resource.

pagination

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/authenticationvserver_vpnportaltheme_binding/name_value<String>?**pagesize=#no;pageno=#no**

Use this query-parameter to get the authenticationvserver_vpnportaltheme_binding resources in chunks.

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "authenticationvserver_vpnportaltheme_binding": [ {

"name":<String_value>,
```

```
"portaltheme":<String_value>,

"acttype":<Double_value>

}]}
```

get (all)

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/authenticationvserver_vpnportaltheme_binding

Query-parameters:

bulkbindings

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/authenticationvserver_vpnportaltheme_binding?**bulkbindings=yes**

NITRO allows you to fetch bindings in bulk.

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "authenticationvserver_vpnportaltheme_binding": [ {

"name":<String_value>,

"portaltheme":<String_value>,
```

```
"acttype":<Double_value>

}]}
```

count

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/authenticationvserver_vpnportaltheme_binding/name_value<String>?**count=yes**

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{"authenticationvserver_vpnportaltheme_binding": [ { "__count": "#no"} ] }
```

authenticationwebauthaction

Configuration for Web authentication action resource.

Properties

(click to see Operations)

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
name	<String>	Read-write	Name for the Web Authentication action. Must begin with a letter, number, or the underscore character (_), and must contain only letters, numbers, and the hyphen (-), period (.) pound (#), space ( ), at (@), equals (=), colon (:), and underscore characters. Cannot be changed after the profile is created. The following requirement applies only to the Citrix ADC CLI: If the name includes one or more spaces, enclose the name in double or single quotation marks (for example, "my authentication action" or 'my authentication action'). Minimum length = 1

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
serverip	<String>	Read-write	IP address of the web server to be used for authentication. Minimum length = 1
serverport	<Integer>	Read-write	Port on which the web server accepts connections. Minimum value = 1 Range 1 - 65535 * in CLI is represented as 65535 in NITRO API
fullreqexpr	<String>	Read-write	Exact HTTP request, in the form of an expression, which the Citrix ADC sends to the authentication server. The Citrix ADC does not check the validity of this request. One must manually validate the request.
scheme	<String>	Read-write	Type of scheme for the web server. Possible values = http, https
successrule	<String>	Read-write	Expression, that checks to see if

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
			authentication is successful.
defaultauthentication group	<String>	Read-write	This is the default group that is chosen when the authentication succeeds in addition to extracted groups.
attribute1	<String>	Read-write	Expression that would be evaluated to extract attribute1 from the webauth response. Maximum length = 128
attribute2	<String>	Read-write	Expression that would be evaluated to extract attribute2 from the webauth response. Maximum length = 128
attribute3	<String>	Read-write	Expression that would be evaluated to extract attribute3 from the webauth response. Maximum length = 128
attribute4	<String>	Read-write	Expression that would be evaluated to extract attribute4 from the webauth response.

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
			Maximum length = 128
attribute5	<String>	Read-write	Expression that would be evaluated to extract attribute5 from the webauth response. Maximum length = 128
attribute6	<String>	Read-write	Expression that would be evaluated to extract attribute6 from the webauth response. Maximum length = 128
attribute7	<String>	Read-write	Expression that would be evaluated to extract attribute7 from the webauth response. Maximum length = 128
attribute8	<String>	Read-write	Expression that would be evaluated to extract attribute8 from the webauth response. Maximum length = 128
attribute9	<String>	Read-write	Expression that would be evaluated to extract attribute9 from the webauth response.

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
			Maximum length = 128
attribute10	<String>	Read-write	Expression that would be evaluated to extract attribute10 from the webauth response. Maximum length = 128
attribute11	<String>	Read-write	Expression that would be evaluated to extract attribute11 from the webauth response. Maximum length = 128
attribute12	<String>	Read-write	Expression that would be evaluated to extract attribute12 from the webauth response. Maximum length = 128
attribute13	<String>	Read-write	Expression that would be evaluated to extract attribute13 from the webauth response. Maximum length = 128
attribute14	<String>	Read-write	Expression that would be evaluated to extract attribute14 from the webauth response.

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
			Maximum length = 128
attribute15	<String>	Read-write	Expression that would be evaluated to extract attribute15 from the webauth response. Maximum length = 128
attribute16	<String>	Read-write	Expression that would be evaluated to extract attribute16 from the webauth response. Maximum length = 128
__count	<Double>	Read-only	count parameter

Operations

(click to see Properties)

ADD| DELETE| UPDATE| UNSET| GET (ALL)| GET| COUNT

Some options that you can use for each operations:

- **Getting warnings in response:**NITRO allows you to get warnings in an operation by specifying the "warning" query parameter as "yes". For example, to get warnings while connecting to the Citrix ADC appliance, the URL is as follows:

  http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/login?warning=yes

  If any, the warnings are displayed in the response payload with the HTTP code "209 X-NITRO-WARNING".

- **Authenticated access for individual NITRO operations:**NITRO allows you to logon to the Citrix ADC appliance to perform individual operations. You can use this option instead of

creating a NITRO session (using the login object) and then using that session to perform all operations,

To do this, you must specify the username and password in the request header of the NITRO request as follows:

X-NITRO-USER:<username>

X-NITRO-PASS:<password>

Note: In such cases, make sure that the request header DOES not include the following:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue>

`*Note:`*

Mandatory parameters are marked in redand placeholder content is marked in <green>.

add

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/authenticationwebauthaction

HTTP Method:POST

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Content-Type:application/json

Request Payload:

```
{"authenticationwebauthaction":{

<b>"name":<String_value>,

</b><b>"serverip":<String_value>,

</b><b>"serverport":<Integer_value>,

</b>"fullreqexpr":<String_value>,

<b>"scheme":<String_value>,

</b><b>"successrule":<String_value>,
```

```
</b>"defaultauthenticationgroup":<String_value>,

"attribute1":<String_value>,

"attribute2":<String_value>,

"attribute3":<String_value>,

"attribute4":<String_value>,

"attribute5":<String_value>,

"attribute6":<String_value>,

"attribute7":<String_value>,

"attribute8":<String_value>,

"attribute9":<String_value>,

"attribute10":<String_value>,

"attribute11":<String_value>,

"attribute12":<String_value>,

"attribute13":<String_value>,

"attribute14":<String_value>,

"attribute15":<String_value>,

"attribute16":<String_value>

}}
```

Response:

HTTP Status Code on Success: 201 Created HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error

delete

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/authenticationwebauthaction/name_value<String>

HTTP Method:DELETE

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue>

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error

update

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/authenticationwebauthaction

HTTP Method:PUT

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Content-Type:application/json

Request Payload:

```
{"authenticationwebauthaction":{

<b>"name":<String_value>,

</b>"serverip":<String_value>,

"serverport":<Integer_value>,

"fullreqexpr":<String_value>,

"scheme":<String_value>,

"successrule":<String_value>,

"defaultauthenticationgroup":<String_value>,
```

```
"attribute1":<String_value>,

"attribute2":<String_value>,

"attribute3":<String_value>,

"attribute4":<String_value>,

"attribute5":<String_value>,

"attribute6":<String_value>,

"attribute7":<String_value>,

"attribute8":<String_value>,

"attribute9":<String_value>,

"attribute10":<String_value>,

"attribute11":<String_value>,

"attribute12":<String_value>,

"attribute13":<String_value>,

"attribute14":<String_value>,

"attribute15":<String_value>,

"attribute16":<String_value>

}}
```

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error

unset

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/authenticationwebauthaction?**action=unset**

HTTP Method:POST

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Content-Type:application/json

Request Payload:

```
{"authenticationwebauthaction":{

<b>"name":<String_value>,

</b>"serverip":true,

"serverport":true,

"fullreqexpr":true,

"defaultauthenticationgroup":true,

"attribute1":true,

"attribute2":true,

"attribute3":true,

"attribute4":true,

"attribute5":true,

"attribute6":true,

"attribute7":true,

"attribute8":true,

"attribute9":true,
```

```
"attribute10":true,

"attribute11":true,

"attribute12":true,

"attribute13":true,

"attribute14":true,

"attribute15":true,

"attribute16":true

}}
```

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error

get (all)

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/authenticationwebauthaction

Query-parameters:

attrs

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/authenticationwebauthaction?**attrs=property-name1,property-name2**

Use this query parameter to specify the resource details that you want to retrieve.

filter

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/authenticationwebauthaction?**filter=property-name1:property-val1,property-name2:property-val2**

Use this query-parameter to get the filtered set of authenticationwebauthaction resources configured on Citrix ADC. Filtering can be done on any of the properties of the resource.

view

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/authenticationwebauthaction?**view=summary**

Note: By default, the retrieved results are displayed in detail view (?view=detail).

pagination

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/authenticationwebauthaction?**pagesize=#no;pageno=#no**

Use this query-parameter to get the authenticationwebauthaction resources in chunks.

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "authenticationwebauthaction": [ {

"name":<String_value>,

"serverip":<String_value>,

"serverport":<Integer_value>,

"fullreqexpr":<String_value>,

"scheme":<String_value>,

"successrule":<String_value>,
```

```
"defaultauthenticationgroup":<String_value>,

"attribute1":<String_value>,

"attribute2":<String_value>,

"attribute3":<String_value>,

"attribute4":<String_value>,

"attribute5":<String_value>,

"attribute6":<String_value>,

"attribute7":<String_value>,

"attribute8":<String_value>,

"attribute9":<String_value>,

"attribute10":<String_value>,

"attribute11":<String_value>,

"attribute12":<String_value>,

"attribute13":<String_value>,

"attribute14":<String_value>,

"attribute15":<String_value>,

"attribute16":<String_value>

}]}
```

get

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/authenticationwebauthaction/name_value<String>

Query-parameters:

attrs

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/authenticationwebauthaction/name_value<String>?**attrs=property-name1,property-name2**

Use this query parameter to specify the resource details that you want to retrieve.

view

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/authenticationwebauthaction/name_value<String>?**view=summary**

Note: By default, the retrieved results are displayed in detail view (?view=detail).

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "authenticationwebauthaction": [ {

"name":<String_value>,

"serverip":<String_value>,

"serverport":<Integer_value>,

"fullreqexpr":<String_value>,

"scheme":<String_value>,
```

```
"successrule":<String_value>,

"defaultauthenticationgroup":<String_value>,

"attribute1":<String_value>,

"attribute2":<String_value>,

"attribute3":<String_value>,

"attribute4":<String_value>,

"attribute5":<String_value>,

"attribute6":<String_value>,

"attribute7":<String_value>,

"attribute8":<String_value>,

"attribute9":<String_value>,

"attribute10":<String_value>,

"attribute11":<String_value>,

"attribute12":<String_value>,

"attribute13":<String_value>,

"attribute14":<String_value>,

"attribute15":<String_value>,

"attribute16":<String_value>

}]}
```

count

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/authenticationwebauthaction?**count=yes**

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "authenticationwebauthaction": [ { "__count": "#no"} ] }
```

authenticationwebauthpolicy

Configuration for Web authentication policy resource.

Properties

(click to see Operations)

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
name	<String>	Read-write	Name for the WebAuth policy. Must begin with a letter, number, or the underscore character (_), and must contain only letters, numbers, and the hyphen (-), period (.) pound (#), space ( ), at (@),

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
			equals (=), colon (:), and underscore characters. Cannot be changed after LDAP policy is created. The following requirement applies only to the Citrix ADC CLI: If the name includes one or more spaces, enclose the name in double or single quotation marks (for example, "my authentication policy" or 'my authentication policy'). Minimum length = 1
rule	<String>	Read-write	Name of the Citrix ADC named rule, or an expression, that the policy uses to determine whether to attempt to authenticate the user with the Web server. Minimum length = 1
action	<String>	Read-write	Name of the WebAuth action to perform if the policy matches.

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
			Minimum length = 1
__count	<Double>	Read-only	count parameter

Operations

(click to see Properties)

ADD| DELETE| UPDATE| GET (ALL)| GET| COUNT

Some options that you can use for each operations:

- **Getting warnings in response:**NITRO allows you to get warnings in an operation by specifying the "warning" query parameter as "yes". For example, to get warnings while connecting to the Citrix ADC appliance, the URL is as follows:

  http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/login?warning=yes

  If any, the warnings are displayed in the response payload with the HTTP code "209 X-NITRO-WARNING".

- **Authenticated access for individual NITRO operations:**NITRO allows you to logon to the Citrix ADC appliance to perform individual operations. You can use this option instead of creating a NITRO session (using the login object) and then using that session to perform all operations,

  To do this, you must specify the username and password in the request header of the NITRO request as follows:

  X-NITRO-USER:<username>

  X-NITRO-PASS:<password>

  Note: In such cases, make sure that the request header DOES not include the following:

  Cookie:NITRO_AUTH_TOKEN=<tokenvalue>

`*Note:`*

Mandatory parameters are marked in redand placeholder content is marked in <green>.

add

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/authenticationwebauthpolicy

HTTP Method:POST

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Content-Type:application/json

Request Payload:

```
{"authenticationwebauthpolicy":{

<b>"name":<String_value>,

</b><b>"rule":<String_value>,

</b><b>"action":<String_value>

</b>}}
```

Response:

HTTP Status Code on Success: 201 Created HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error

delete

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/authenticationwebauthpolicy/name_value<String>

HTTP Method:DELETE

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue>

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error

update

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/authenticationwebauthpolicy

HTTP Method:PUT

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Content-Type:application/json

Request Payload:

```
{"authenticationwebauthpolicy":{

<b>"name":<String_value>,

</b>"rule":<String_value>,

"action":<String_value>

}}
```

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error

get (all)

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/authenticationwebauthpolicy

Query-parameters:

attrs

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/authenticationwebauthpolicy?**attrs=property-name1,property-name2**

Use this query parameter to specify the resource details that you want to retrieve.

filter

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/authenticationwebauthpolicy?**filter=property-name1:property-val1,property-name2:property-val2**

Use this query-parameter to get the filtered set of authenticationwebauthpolicy resources configured on Citrix ADC. Filtering can be done on any of the properties of the resource.

view

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/authenticationwebauthpolicy?**view=summary**

Note: By default, the retrieved results are displayed in detail view (?view=detail).

pagination

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/authenticationwebauthpolicy?**pagesize=#no;pageno=#no**

Use this query-parameter to get the authenticationwebauthpolicy resources in chunks.

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "authenticationwebauthpolicy": [ {

"name":<String_value>,

"rule":<String_value>,

"action":<String_value>
```

```
}]}
```

get

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/authenticationwebauthpolicy/name_value<String>

Query-parameters:

attrs

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/authenticationwebauthpolicy/name_value<String>?**attrs=property-name1,property-name2**

Use this query parameter to specify the resource details that you want to retrieve.

view

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/authenticationwebauthpolicy/name_value<String>?**view=summary**

Note: By default, the retrieved results are displayed in detail view (?view=detail).

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "authenticationwebauthpolicy": [ {
```

```
"name":<String_value>,

"rule":<String_value>,

"action":<String_value>

}]}
```

count

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/authenticationwebauthpolicy?**count=yes**

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "authenticationwebauthpolicy": [ { "__count": "#no"} ] }
```

authenticationwebauthpolicy_authenticationvserver_binding

Binding object showing the authenticationvserver that can be bound to authenticationwebauthpolicy.

Properties

(click to see Operations)

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
boundto	<String>	Read-write	The entity name to which policy is bound.
name	<String>	Read-write	Name of the WebAuth policy. Minimum length = 1
priority	<Double>	Read-only	.
activepolicy	<Double>	Read-only	.
__count	<Double>	Read-only	count parameter

Operations

(click to see Properties)

GET| GET (ALL)| COUNT

Some options that you can use for each operations:

- **Getting warnings in response:**NITRO allows you to get warnings in an operation by specifying the "warning" query parameter as "yes". For example, to get warnings while connecting to the Citrix ADC appliance, the URL is as follows:

  http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/login?warning=yes

  If any, the warnings are displayed in the response payload with the HTTP code "209 X-NITRO-WARNING".

- **Authenticated access for individual NITRO operations:**NITRO allows you to logon to the Citrix ADC appliance to perform individual operations. You can use this option instead of creating a NITRO session (using the login object) and then using that session to perform all operations,

  To do this, you must specify the username and password in the request header of the NITRO request as follows:

  X-NITRO-USER:<username>

  X-NITRO-PASS:<password>

Note: In such cases, make sure that the request header DOES not include the following:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue>

`*Note:`*

Mandatory parameters are marked in redand placeholder content is marked in <green>.

get

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/authenticationwebauthpolicy_authenticationvserver_binding/name_value<String>

Query-parameters:

filter

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/authenticationwebauthpolicy_authenticationvserver_binding/name_value<String>?**filter=property-name1:property-value1,property-name2:property-value2**

Use this query-parameter to get the filtered set of authenticationwebauthpolicy_authenticationvserver_binding resources configured on Citrix ADC. Filtering can be done on any of the properties of the resource.

pagination

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/authenticationwebauthpolicy_authenticationvserver_binding/name_value<String>?**pagesize=#no;pageno=#no**

Use this query-parameter to get the authenticationwebauthpolicy_authenticationvserver_binding resources in chunks.

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "authenticationwebauthpolicy_authenticationvserver_binding": [ {

"boundto":<String_value>,

"name":<String_value>,

"priority":<Double_value>,

"activepolicy":<Double_value>

}]}
```

get (all)

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/authenticationwebauthpolicy_authenticationvserver_binding

Query-parameters:

bulkbindings

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/authenticationwebauthpolicy_authenticationvserver_binding?**bulkbindings=yes**

NITRO allows you to fetch bindings in bulk.

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "authenticationwebauthpolicy_authenticationvserver_binding": [ {

"boundto":<String_value>,

"name":<String_value>,

"priority":<Double_value>,

"activepolicy":<Double_value>

}]}
```

count

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/authenticationwebauthpolicy_authenticationvserver_binding/name_value<String>?**count=yes**

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{"authenticationwebauthpolicy_authenticationvserver_binding":
[ { "__count": "#no"} ] }
```

authenticationwebauthpolicy_binding

Binding object which returns the resources bound to authenticationwebauthpolicy.

Properties

(click to see Operations)

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
name	<String>	Read-write	Name of the WebAuth policy. Minimum length = 1
authenticationwebauthpolicy_vpnvserver_binding	<authenticationwebauthpolicy_vpnvserver_binding[]>	Read-only	vpnvserver that can be bound to authenticationwebauthpolicy.
authenticationwebauthpolicy_authenticationvserver_binding	<authenticationwebauthpolicy_authenticationvserver_binding[]>	Read-only	authenticationvserver that can be bound to authenticationwebauthpolicy.
authenticationwebauthpolicy_systemglobal_binding	<authenticationwebauthpolicy_systemglobal_binding[]>	Read-only	systemglobal that can be bound to authenticationwebauthpolicy.
authenticationwebauthpolicy_vpnglobal_binding	<authenticationwebauthpolicy_vpnglobal_binding[]>	Read-only	vpnglobal that can be bound to authenticationwebauthpolicy.

Operations

(click to see Properties)

GET| GET (ALL)

Some options that you can use for each operations:

- **Getting warnings in response:**NITRO allows you to get warnings in an operation by specifying the "warning" query parameter as "yes". For example, to get warnings while connecting to the Citrix ADC appliance, the URL is as follows:

  http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/login?warning=yes

  If any, the warnings are displayed in the response payload with the HTTP code "209 X-NITRO-WARNING".

- **Authenticated access for individual NITRO operations:**NITRO allows you to logon to the Citrix ADC appliance to perform individual operations. You can use this option instead of creating a NITRO session (using the login object) and then using that session to perform all operations,

  To do this, you must specify the username and password in the request header of the NITRO request as follows:

  X-NITRO-USER:<username>

  X-NITRO-PASS:<password>

  Note: In such cases, make sure that the request header DOES not include the following:

  Cookie:NITRO_AUTH_TOKEN=<tokenvalue>

`*Note:`*

Mandatory parameters are marked in redand placeholder content is marked in <green>.

get

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/authenticationwebauthpolicy_binding/name_value<String>

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "authenticationwebauthpolicy_binding": [ {

"name":<String_value>,

"authenticationwebauthpolicy_authenticationvserver_binding":<authenticatio
nwebauthpolicy_authenticationvserver_binding[]_value>,

"authenticationwebauthpolicy_systemglobal_binding":<authenticationwebauthp
olicy_systemglobal_binding[]_value>,

"authenticationwebauthpolicy_vpnvserver_binding":<authenticationwebauthpol
icy_vpnvserver_binding[]_value>,

"authenticationwebauthpolicy_vpnglobal_binding":<authenticationwebauthpoli
cy_vpnglobal_binding[]_value>

}]}
```

get (all)

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/authenticationwebauthpolicy_binding

Query-parameters:

bulkbindings

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/authenticationwebauthpolicy_binding?**bulkbindings=yes**

NITRO allows you to fetch bindings in bulk.

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "authenticationwebauthpolicy_binding": [ {

"name":<String_value>,

"authenticationwebauthpolicy_authenticationvserver_binding":<authenticatio
nwebauthpolicy_authenticationvserver_binding[]_value>,

"authenticationwebauthpolicy_systemglobal_binding":<authenticationwebauthp
olicy_systemglobal_binding[]_value>,

"authenticationwebauthpolicy_vpnvserver_binding":<authenticationwebauthpol
icy_vpnvserver_binding[]_value>,

"authenticationwebauthpolicy_vpnglobal_binding":<authenticationwebauthpoli
cy_vpnglobal_binding[]_value>

}]}
```

authenticationwebauthpolicy_systemglobal_binding

Binding object showing the systemglobal that can be bound to authenticationwebauthpolicy.

Properties

(click to see Operations)

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
boundto	<String>	Read-write	The entity name to which policy is bound.

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
name	<String>	Read-write	Name of the WebAuth policy. Minimum length = 1
priority	<Double>	Read-only	.
activepolicy	<Double>	Read-only	.
__count	<Double>	Read-only	count parameter

Operations

(click to see Properties)

GET| GET (ALL)| COUNT

Some options that you can use for each operations:

- **Getting warnings in response:**NITRO allows you to get warnings in an operation by specifying the "warning" query parameter as "yes". For example, to get warnings while connecting to the Citrix ADC appliance, the URL is as follows:

  http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/login?warning=yes

  If any, the warnings are displayed in the response payload with the HTTP code "209 X-NITRO-WARNING".

- **Authenticated access for individual NITRO operations:**NITRO allows you to logon to the Citrix ADC appliance to perform individual operations. You can use this option instead of creating a NITRO session (using the login object) and then using that session to perform all operations,

  To do this, you must specify the username and password in the request header of the NITRO request as follows:

  X-NITRO-USER:<username>

  X-NITRO-PASS:<password>

  Note: In such cases, make sure that the request header DOES not include the following:

  Cookie:NITRO_AUTH_TOKEN=<tokenvalue>

`*Note:`*

Mandatory parameters are marked in redand placeholder content is marked in <green>.

get

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/authenticationwebauthpolicy_systemglobal_binding/name_value<String>

Query-parameters:

filter

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/authenticationwebauthpolicy_systemglobal_binding/name_value<String>?**filter=property-name1:property-value1,property-name2:property-value2**

Use this query-parameter to get the filtered set of authenticationwebauthpolicy_systemglobal_binding resources configured on Citrix ADC. Filtering can be done on any of the properties of the resource.

pagination

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/authenticationwebauthpolicy_systemglobal_binding/name_value<String>?**pagesize=#no;pageno=#no**

Use this query-parameter to get the authenticationwebauthpolicy_systemglobal_binding resources in chunks.

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "authenticationwebauthpolicy_systemglobal_binding": [ {

"boundto":<String_value>,

"name":<String_value>,

"priority":<Double_value>,

"activepolicy":<Double_value>

}]}
```

get (all)

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/authenticationwebauthpolicy_systemglobal_binding

Query-parameters:

bulkbindings

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/authenticationwebauthpolicy_systemglobal_binding?**bulkbindings=yes**

NITRO allows you to fetch bindings in bulk.

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "authenticationwebauthpolicy_systemglobal_binding": [ {

"boundto":<String_value>,

"name":<String_value>,

"priority":<Double_value>,

"activepolicy":<Double_value>

}]}
```

count

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/authenticationwebauthpolicy_systemglobal_binding/name_value<String>?**count=yes**

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{"authenticationwebauthpolicy_systemglobal_binding": [ { "__count":
"#no"} ] }
```

authenticationwebauthpolicy_vpnglobal_binding

Binding object showing the vpnglobal that can be bound to authenticationwebauthpolicy.

Properties

(click to see Operations)

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
boundto	<String>	Read-write	The entity name to which policy is bound.
name	<String>	Read-write	Name of the WebAuth policy. Minimum length = 1
priority	<Double>	Read-only	.
activepolicy	<Double>	Read-only	.
__count	<Double>	Read-only	count parameter

Operations

(click to see Properties)

GET| GET (ALL)| COUNT

Some options that you can use for each operations:

- **Getting warnings in response:**NITRO allows you to get warnings in an operation by specifying the "warning" query parameter as "yes". For example, to get warnings while connecting to the Citrix ADC appliance, the URL is as follows:

  http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/login?warning=yes

  If any, the warnings are displayed in the response payload with the HTTP code "209 X-NITRO-WARNING".

- **Authenticated access for individual NITRO operations:**NITRO allows you to logon to the Citrix ADC appliance to perform individual operations. You can use this option instead of

creating a NITRO session (using the login object) and then using that session to perform all operations,

To do this, you must specify the username and password in the request header of the NITRO request as follows:

X-NITRO-USER:<username>

X-NITRO-PASS:<password>

Note: In such cases, make sure that the request header DOES not include the following:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue>

`*Note:`*

Mandatory parameters are marked in redand placeholder content is marked in <green>.

get

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/authenticationwebauthpolicy_vpnglobal_binding/name_value<String>

Query-parameters:

filter

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/authenticationwebauthpolicy_vpnglobal_binding/name_value<String>?**filter=property-name1:property-value1,property-name2:property-value2**

Use this query-parameter to get the filtered set of authenticationwebauthpolicy_vpnglobal_binding resources configured on Citrix ADC. Filtering can be done on any of the properties of the resource.

pagination

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/authenticationwebauthpolicy_vpnglobal_binding/name_value<String>?**pagesize=#no;pageno=#no**

Use this query-parameter to get the authenticationwebauthpolicy_vpnglobal_binding resources in chunks.

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "authenticationwebauthpolicy_vpnglobal_binding": [ {

"boundto":<String_value>,

"name":<String_value>,

"priority":<Double_value>,

"activepolicy":<Double_value>

}]}
```

get (all)

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/authenticationwebauthpolicy_vpnglobal_binding

Query-parameters:

bulkbindings

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/authenticationwebauthpolicy_vpnglobal_binding?**bulkbindings=yes**

NITRO allows you to fetch bindings in bulk.

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "authenticationwebauthpolicy_vpnglobal_binding": [ {

"boundto":<String_value>,

"name":<String_value>,

"priority":<Double_value>,

"activepolicy":<Double_value>

}]}
```

count

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/authenticationwebauthpolicy_vpnglobal_binding/name_value<String>?**count=yes**

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{"authenticationwebauthpolicy_vpnglobal_binding": [ { "__count":
"#no"} ] }
```

authenticationwebauthpolicy_vpnvserver_binding

Binding object showing the vpnvserver that can be bound to authenticationwebauthpolicy.

Properties

(click to see Operations)

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
boundto	<String>	Read-write	The entity name to which policy is bound.
name	<String>	Read-write	Name of the WebAuth policy. Minimum length = 1
priority	<Double>	Read-only	.
activepolicy	<Double>	Read-only	.
__count	<Double>	Read-only	count parameter

Operations

(click to see Properties)

GET| GET (ALL)| COUNT

Some options that you can use for each operations:

- **Getting warnings in response:**NITRO allows you to get warnings in an operation by specifying the "warning" query parameter as "yes". For example, to get warnings while connecting to the Citrix ADC appliance, the URL is as follows:

  http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/login?warning=yes

  If any, the warnings are displayed in the response payload with the HTTP code "209 X-NITRO-WARNING".

- **Authenticated access for individual NITRO operations:**NITRO allows you to logon to the Citrix ADC appliance to perform individual operations. You can use this option instead of creating a NITRO session (using the login object) and then using that session to perform all operations,

  To do this, you must specify the username and password in the request header of the NITRO request as follows:

  X-NITRO-USER:<username>

  X-NITRO-PASS:<password>

  Note: In such cases, make sure that the request header DOES not include the following:

  Cookie:NITRO_AUTH_TOKEN=<tokenvalue>

`*Note:`*

Mandatory parameters are marked in redand placeholder content is marked in <green>.

get

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/authenticationwebauthpolicy_vpnvserver_binding/name_value<String>

Query-parameters:

filter

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/authenticationwebauthpolicy_vpnvserver_binding/name_value<String>?**filter=property-name1:property-value1,property-name2:property-value2**

Use this query-parameter to get the filtered set of authenticationwebauthpolicy_vpnvserver_binding resources configured on Citrix ADC. Filtering can be done on any of the properties of the resource.

pagination

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/authenticationwebauthpolicy_vpnvserver_binding/name_value<String>?**pagesize=#no;pageno=#no**

Use this query-parameter to get the authenticationwebauthpolicy_vpnvserver_binding resources in chunks.

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "authenticationwebauthpolicy_vpnvserver_binding": [ {

"boundto":<String_value>,

"name":<String_value>,

"priority":<Double_value>,

"activepolicy":<Double_value>

}]}
```

get (all)

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/authenticationwebauthpolicy_vpnvserver_binding

Query-parameters:

bulkbindings

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/authenticationwebauthpolicy_vpnvserver_binding?**bulkbindings=yes**

NITRO allows you to fetch bindings in bulk.

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "authenticationwebauthpolicy_vpnvserver_binding": [ {

"boundto":<String_value>,

"name":<String_value>,

"priority":<Double_value>,

"activepolicy":<Double_value>

}]}
```

count

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/authenticationwebauthpolicy_vpnvserver_binding/name_value<String>?**count=yes**

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{"authenticationwebauthpolicy_vpnvserver_binding": [ { "__count":
"#no"} ] }
```

authenticationazurekeyvault

Configuration for Azure Key Vault entity resource.

Properties

(click to see Operations)

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
name	<String>	Read-write	Name for the new Azure Key Vault profile. Must begin with an ASCII alphanumeric or underscore (_) character, and must

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
			contain only ASCII alphanumeric, underscore, hash (#), period (.), space, colon (:), at (@), equals (=), and hyphen (-) characters. Cannot be changed after an action is created. The following requirement applies only to the Citrix ADC CLI: If the name includes one or more spaces, enclose the name in double or single quotation marks (for example, "my action" or 'my action'). Minimum length = 1
vaultname	<String>	Read-write	Name of the Azure vault account as configured in azure portal. Minimum length = 1 Maximum length = 63
clientid	<String>	Read-write	Unique identity of the relying party requesting for authentication.

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
			Minimum length = 1
clientsecret	<String>	Read-write	Unique secret string to authorize relying party at authorization server. Minimum length = 1
servicekeyname	<String>	Read-write	Friendly name of the Key to be used to compute signature. Minimum length = 1
signaturealg	<String>	Read-write	Algorithm to be used to sign/verify transactions. Default value: RS256 Possible values = RS256
tokenendpoint	<String>	Read-write	URL endpoint on relying party to which the OAuth token is to be sent. Minimum length = 1
pushservice	<String>	Read-write	Name of the service used to send push notifications. Minimum length = 1
defaultauthentication group	<String>	Read-write	This is the group that is added to user sessions that match current IdP policy. It can be used in policies to identify relying party trust.

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
refreshinterval	<Double>	Read-write	Interval at which access token in obtained. Default value: 50
tenantid	<String>	Read-write	TenantID of the application. This is usually specific to providers such as Microsoft and usually refers to the deployment identifier.
authentication	<String>	Read-write	If authentication is disabled, otp checks are not performed after azure vault keys are obtained. This is useful to distinguish whether user has registered devices. . Default value: ENABLED Possible values = ENABLED, DISABLED
__count	<Double>	Read-only	count parameter

Operations

(click to see Properties)

ADD| DELETE| UPDATE| UNSET| GET (ALL)| GET| COUNT

Some options that you can use for each operations:

- **Getting warnings in response:**NITRO allows you to get warnings in an operation by specifying the "warning" query parameter as "yes". For example, to get warnings while connecting to the Citrix ADC appliance, the URL is as follows:

  http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/login?warning=yes

  If any, the warnings are displayed in the response payload with the HTTP code "209 X-NITRO-WARNING".

- **Authenticated access for individual NITRO operations:**NITRO allows you to logon to the Citrix ADC appliance to perform individual operations. You can use this option instead of creating a NITRO session (using the login object) and then using that session to perform all operations,

  To do this, you must specify the username and password in the request header of the NITRO request as follows:

  X-NITRO-USER:<username>

  X-NITRO-PASS:<password>

  Note: In such cases, make sure that the request header DOES not include the following:

  Cookie:NITRO_AUTH_TOKEN=<tokenvalue>

`*Note:`*

Mandatory parameters are marked in redand placeholder content is marked in <green>.

add

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/authenticationazurekeyvault

HTTP Method:POST

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Content-Type:application/json

Request Payload:

```
{"authenticationazurekeyvault":{
```

```
<b>"name":<String_value>,

</b><b>"vaultname":<String_value>,

</b><b>"clientid":<String_value>,

</b><b>"clientsecret":<String_value>,

</b><b>"servicekeyname":<String_value>,

</b>"signaturealg":<String_value>,

"tokenendpoint":<String_value>,

"pushservice":<String_value>,

"defaultauthenticationgroup":<String_value>,

"refreshinterval":<Double_value>,

"tenantid":<String_value>,

"authentication":<String_value>

}}
```

Response:

HTTP Status Code on Success: 201 Created HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error

delete

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/authenticationazurekeyvault/name_value<String>

HTTP Method:DELETE

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue>

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error

update

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/authenticationazurekeyvault

HTTP Method:PUT

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Content-Type:application/json

Request Payload:

```
{"authenticationazurekeyvault":{

<b>"name":<String_value>,

</b>"vaultname":<String_value>,

"clientid":<String_value>,

"clientsecret":<String_value>,

"servicekeyname":<String_value>,

"signaturealg":<String_value>,

"tokenendpoint":<String_value>,

"pushservice":<String_value>,

"defaultauthenticationgroup":<String_value>,

"refreshinterval":<Double_value>,

"tenantid":<String_value>,
```

```
"authentication":<String_value>

}}
```

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error

unset

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/authenticationazurekeyvault?**action=unset**

HTTP Method:POST

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Content-Type:application/json

Request Payload:

```
{"authenticationazurekeyvault":{

<b>"name":<String_value>,

</b>"signaturealg":true,

"pushservice":true,

"defaultauthenticationgroup":true,

"refreshinterval":true,

"authentication":true

}}
```

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error

get (all)

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/authenticationazurekeyvault

Query-parameters:

attrs

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/authenticationazurekeyvault?**attrs=property-name1,property-name2**

Use this query parameter to specify the resource details that you want to retrieve.

filter

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/authenticationazurekeyvault?**filter=property-name1:property-val1,property-name2:property-val2**

Use this query-parameter to get the filtered set of authenticationazurekeyvault resources configured on Citrix ADC. Filtering can be done on any of the properties of the resource.

view

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/authenticationazurekeyvault?**view=summary**

Note: By default, the retrieved results are displayed in detail view (?view=detail).

pagination

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/authenticationazurekeyvault?**pagesize=#no;pageno=#no**

Use this query-parameter to get the authenticationazurekeyvault resources in chunks.

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "authenticationazurekeyvault": [ {

"name":<String_value>,

"vaultname":<String_value>,

"clientid":<String_value>,

"clientsecret":<String_value>,

"servicekeyname":<String_value>,

"signaturealg":<String_value>,

"tokenendpoint":<String_value>,

"pushservice":<String_value>,

"defaultauthenticationgroup":<String_value>,

"refreshinterval":<Double_value>,

"tenantid":<String_value>,

"authentication":<String_value>

}]}
```

get

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/authenticationazurekeyvault/name_value<String>

Query-parameters:

attrs

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/authenticationazurekeyvault/name_value<String>?**attrs=property-name1,property-name2**

Use this query parameter to specify the resource details that you want to retrieve.

view

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/authenticationazurekeyvault/name_value<String>?**view=summary**

Note: By default, the retrieved results are displayed in detail view (?view=detail).

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "authenticationazurekeyvault": [ {

"name":<String_value>,

"vaultname":<String_value>,
```

```
"clientid":<String_value>,

"clientsecret":<String_value>,

"servicekeyname":<String_value>,

"signaturealg":<String_value>,

"tokenendpoint":<String_value>,

"pushservice":<String_value>,

"defaultauthenticationgroup":<String_value>,

"refreshinterval":<Double_value>,

"tenantid":<String_value>,

"authentication":<String_value>

}]}
```

count

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/authenticationazurekeyvault?**count=yes**

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "authenticationazurekeyvault": [ { "__count": "#no"} ] }
```

authenticationcaptchaaction

Configuration for Captcha Action resource.

Properties

(click to see Operations)

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
name	<String>	Read-write	Name for the new captcha action. Must begin with an ASCII alphanumeric or underscore (_) character, and must contain only ASCII alphanumeric, underscore, hash (#), period (.), space, colon (:), at (@), equals (=), and hyphen (-) characters. Cannot be changed after an action is created. The following requirement applies only to the Citrix ADC CLI: If the name includes one or more spaces, enclose the name in double or single quotation

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
			marks (for example, "my action" or 'my action'). Minimum length = 1
serverurl	<String>	Read-write	This is the endpoint at which captcha response is validated.
secretkey	<String>	Read-write	Secret of gateway as established at the captcha source.
sitekey	<String>	Read-write	Sitekey to identify gateway fqdn while loading captcha.
defaultauthentication group	<String>	Read-write	This is the group that is added to user sessions that match current policy.
__count	<Double>	Read-only	count parameter

Operations

(click to see Properties)

ADD| DELETE| UPDATE| UNSET| GET (ALL)| GET| COUNT

Some options that you can use for each operations:

- **Getting warnings in response:**NITRO allows you to get warnings in an operation by specifying the "warning" query parameter as "yes". For example, to get warnings while connecting to the Citrix ADC appliance, the URL is as follows:

  http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/login?warning=yes

  If any, the warnings are displayed in the response payload with the HTTP code "209 X-NITRO-WARNING".

- **Authenticated access for individual NITRO operations:**NITRO allows you to logon to the Citrix ADC appliance to perform individual operations. You can use this option instead of creating a NITRO session (using the login object) and then using that session to perform all operations,

  To do this, you must specify the username and password in the request header of the NITRO request as follows:

  X-NITRO-USER:<username>

  X-NITRO-PASS:<password>

  Note: In such cases, make sure that the request header DOES not include the following:

  Cookie:NITRO_AUTH_TOKEN=<tokenvalue>

`*Note:`*

Mandatory parameters are marked in redand placeholder content is marked in <green>.

add

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/authenticationcaptchaaction

HTTP Method:POST

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Content-Type:application/json

Request Payload:

```
{"authenticationcaptchaaction":{

<b>"name":<String_value>,

</b>"serverurl":<String_value>,

<b>"secretkey":<String_value>,

</b><b>"sitekey":<String_value>,

</b>"defaultauthenticationgroup":<String_value>
```

```
}}
```

Response:

HTTP Status Code on Success: 201 Created HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error

delete

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/authenticationcaptchaaction/name_value<String>

HTTP Method:DELETE

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue>

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error

update

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/authenticationcaptchaaction

HTTP Method:PUT

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Content-Type:application/json

Request Payload:

```
{"authenticationcaptchaaction":{

<b>"name":<String_value>,
```

```
</b>"serverurl":<String_value>,

"secretkey":<String_value>,

"sitekey":<String_value>,

"defaultauthenticationgroup":<String_value>

}}
```

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error

unset

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/authenticationcaptchaaction?**action=unset**

HTTP Method:POST

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Content-Type:application/json

Request Payload:

```
{"authenticationcaptchaaction":{

<b>"name":<String_value>,

</b>"serverurl":true,

"defaultauthenticationgroup":true

}}
```

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error

get (all)

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/authenticationcaptchaaction

Query-parameters:

attrs

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/authenticationcaptchaaction?**attrs=property-name1,property-name2**

Use this query parameter to specify the resource details that you want to retrieve.

filter

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/authenticationcaptchaaction?**filter=property-name1:property-val1,property-name2:property-val2**

Use this query-parameter to get the filtered set of authenticationcaptchaaction resources configured on Citrix ADC. Filtering can be done on any of the properties of the resource.

view

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/authenticationcaptchaaction?**view=summary**

Note: By default, the retrieved results are displayed in detail view (?view=detail).

pagination

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/authenticationcaptchaaction?**pagesize=#no;pageno=#no**

Use this query-parameter to get the authenticationcaptchaaction resources in chunks.

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "authenticationcaptchaaction": [ {

"name":<String_value>,

"serverurl":<String_value>,

"secretkey":<String_value>,

"sitekey":<String_value>,

"defaultauthenticationgroup":<String_value>

}]}
```

get

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/authenticationcaptchaaction/name_value<String>

Query-parameters:

attrs

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/authenticationcaptchaaction/name_value<String>?**attrs=property-name1,property-name2**

Use this query parameter to specify the resource details that you want to retrieve.

view

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/authenticationcaptchaaction/name_value<String>?**view=summary**

Note: By default, the retrieved results are displayed in detail view (?view=detail).

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "authenticationcaptchaaction": [ {

"name":<String_value>,

"serverurl":<String_value>,

"secretkey":<String_value>,

"sitekey":<String_value>,

"defaultauthenticationgroup":<String_value>

}]}
```

count

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/authenticationcaptchaaction?**count=yes**

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "authenticationcaptchaaction": [ { "__count": "#no"} ] }
```

authenticationemailaction

Configuration for Email entity resource.

Properties

(click to see Operations)

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
name	<String>	Read-write	Name for the new email action. Must begin with an ASCII alphanumeric or underscore (_) character, and must contain only ASCII alphanumeric, underscore, hash (#), period (.), space, colon (:), at (@), equals (=), and hyphen (-) characters. Cannot

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
			be changed after an action is created. The following requirement applies only to the Citrix ADC CLI: If the name includes one or more spaces, enclose the name in double or single quotation marks (for example, "my action" or 'my action'). Minimum length = 1
username	<String>	Read-write	Username/Clientid/ EmailID to be used to authenticate to the server. Minimum length = 1
password	<String>	Read-write	Password/ Clientsecret to use when authenticating to the server. Minimum length = 1
serverurl	<String>	Read-write	Address of the server that delivers the message. It is fully qualified fqdn such as http(s):// or smtp(s):// for http and smtp protocols respectively. For SMTP, the port number is mandatory

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
			like smtps://smtp.example.com:25. Minimum length = 1
content	<String>	Read-write	Content to be delivered to the user. "$code" string within the content will be replaced with the actual one-time-code to be sent.
defaultauthenticationgroup	<String>	Read-write	This is the group that is added to user sessions that match current IdP policy. It can be used in policies to identify relying party trust.
timeout	<Double>	Read-write	Time after which the code expires. Default value: 30
type	<String>	Read-write	Type of the email action. Default type is SMTP. Default value: SMTP Possible values = SMTP, ATHENA
emailaddress	<String>	Read-write	An optional expression that yields user's email. When not configured, user's default mail address would be used.

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
			When configured, result of this expression is used as destination email address.
__count	<Double>	Read-only	count parameter

Operations

(click to see Properties)

ADD| DELETE| UPDATE| UNSET| GET (ALL)| GET| COUNT

Some options that you can use for each operations:

- **Getting warnings in response:**NITRO allows you to get warnings in an operation by specifying the "warning" query parameter as "yes". For example, to get warnings while connecting to the Citrix ADC appliance, the URL is as follows:

  http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/login?warning=yes

  If any, the warnings are displayed in the response payload with the HTTP code "209 X-NITRO-WARNING".

- **Authenticated access for individual NITRO operations:**NITRO allows you to logon to the Citrix ADC appliance to perform individual operations. You can use this option instead of creating a NITRO session (using the login object) and then using that session to perform all operations,

  To do this, you must specify the username and password in the request header of the NITRO request as follows:

  X-NITRO-USER:<username>

  X-NITRO-PASS:<password>

  Note: In such cases, make sure that the request header DOES not include the following:

  Cookie:NITRO_AUTH_TOKEN=<tokenvalue>

`*Note:` *

Mandatory parameters are marked in redand placeholder content is marked in <green>.

add

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/authenticationemailaction

HTTP Method:POST

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Content-Type:application/json

Request Payload:

```
{"authenticationemailaction":{

<b>"name":<String_value>,

</b><b>"username":<String_value>,

</b><b>"password":<String_value>,

</b><b>"serverurl":<String_value>,

</b>"content":<String_value>,

"defaultauthenticationgroup":<String_value>,

"timeout":<Double_value>,

"type":<String_value>,

"emailaddress":<String_value>

}}
```

Response:

HTTP Status Code on Success: 201 Created HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error

delete

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/authenticationemailaction/name_value<String>

HTTP Method:DELETE

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue>

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error

update

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/authenticationemailaction

HTTP Method:PUT

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Content-Type:application/json

Request Payload:

```
{"authenticationemailaction":{

<b>"name":<String_value>,

</b>"username":<String_value>,

"password":<String_value>,

"serverurl":<String_value>,

"content":<String_value>,

"defaultauthenticationgroup":<String_value>,

"timeout":<Double_value>,
```

```
"type":<String_value>,

"emailaddress":<String_value>

}}
```

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error

unset

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/authenticationemailaction?**action=unset**

HTTP Method:POST

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Content-Type:application/json

Request Payload:

```
{"authenticationemailaction":{

<b>"name":<String_value>,

</b>"content":true,

"defaultauthenticationgroup":true,

"timeout":true,

"type":true,

"emailaddress":true

}}
```

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error

get (all)

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/authenticationemailaction

Query-parameters:

attrs

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/authenticationemailaction?**attrs=property-name1,property-name2**

Use this query parameter to specify the resource details that you want to retrieve.

filter

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/authenticationemailaction?**filter=property-name1:property-val1,property-name2:property-val2**

Use this query-parameter to get the filtered set of authenticationemailaction resources configured on Citrix ADC. Filtering can be done on any of the properties of the resource.

view

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/authenticationemailaction?**view=summary**

Note: By default, the retrieved results are displayed in detail view (?view=detail).

pagination

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/authenticationemailaction?**pagesize=#no;pageno=#no**

Use this query-parameter to get the authenticationemailaction resources in chunks.

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "authenticationemailaction": [ {

"name":<String_value>,

"username":<String_value>,

"password":<String_value>,

"serverurl":<String_value>,

"content":<String_value>,

"defaultauthenticationgroup":<String_value>,

"timeout":<Double_value>,

"type":<String_value>,

"emailaddress":<String_value>

}]}
```

get

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/authenticationemailaction/name_value<String>

Query-parameters:

attrs

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/authenticationemailaction/name_value<String>?**attrs=property-name1,property-name2**

Use this query parameter to specify the resource details that you want to retrieve.

view

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/authenticationemailaction/name_value<String>?**view=summary**

Note: By default, the retrieved results are displayed in detail view (?view=detail).

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "authenticationemailaction": [ {

"name":<String_value>,

"username":<String_value>,

"password":<String_value>,

"serverurl":<String_value>,

"content":<String_value>,
```

```
"defaultauthenticationgroup":<String_value>,

"timeout":<Double_value>,

"type":<String_value>,

"emailaddress":<String_value>

}]}
```

count

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/authenticationemailaction?**count=yes**

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "authenticationemailaction": [ { "__count": "#no"} ] }
```

authenticationnoauthaction

Configuration for no authentication action resource.

Properties

(click to see Operations)

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
name	<String>	Read-write	Name for the new no-authentication action. Must begin with an ASCII alphanumeric or underscore (_) character, and must contain only ASCII alphanumeric, underscore, hash (#), period (.), space, colon (:), at (@), equals (=), and hyphen (-) characters. Cannot be changed after an action is created. The following requirement applies only to the Citrix ADC CLI: If the name includes one or more spaces, enclose the name in double or single quotation marks (for example, "my action" or 'my action'). Minimum length = 1
defaultauthentication group	<String>	Read-write	This is the group that is added to user sessions that match current policy.
__count	<Double>	Read-only	count parameter

Operations

(click to see Properties)

ADD| DELETE| UPDATE| UNSET| GET (ALL)| GET| COUNT

Some options that you can use for each operations:

- **Getting warnings in response:**NITRO allows you to get warnings in an operation by specifying the "warning" query parameter as "yes". For example, to get warnings while connecting to the Citrix ADC appliance, the URL is as follows:

  http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/login?warning=yes

  If any, the warnings are displayed in the response payload with the HTTP code "209 X-NITRO-WARNING".

- **Authenticated access for individual NITRO operations:**NITRO allows you to logon to the Citrix ADC appliance to perform individual operations. You can use this option instead of creating a NITRO session (using the login object) and then using that session to perform all operations,

  To do this, you must specify the username and password in the request header of the NITRO request as follows:

  X-NITRO-USER:<username>

  X-NITRO-PASS:<password>

  Note: In such cases, make sure that the request header DOES not include the following:

  Cookie:NITRO_AUTH_TOKEN=<tokenvalue>

`*Note:`*

Mandatory parameters are marked in redand placeholder content is marked in <green>.

add

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/authenticationnoauthaction

HTTP Method:POST

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Content-Type:application/json

Request Payload:

```
{"authenticationnoauthaction":{

<b>"name":<String_value>,

</b>"defaultauthenticationgroup":<String_value>

}}
```

Response:

HTTP Status Code on Success: 201 Created HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error

delete

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/authenticationnoauthaction/name_value<String>

HTTP Method:DELETE

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue>

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error

update

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/authenticationnoauthaction

HTTP Method:PUT

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Content-Type:application/json

Request Payload:

```
{"authenticationnoauthaction":{

<b>"name":<String_value>,

</b>"defaultauthenticationgroup":<String_value>

}}
```

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error

unset

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/authenticationnoauthaction?**action=unset**

HTTP Method:POST

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Content-Type:application/json

Request Payload:

```
{"authenticationnoauthaction":{

<b>"name":<String_value>,

</b>"defaultauthenticationgroup":true

}}
```

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error

get (all)

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/authenticationnoauthaction

Query-parameters:

attrs

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/authenticationnoauthaction?**attrs=property-name1,property-name2**

Use this query parameter to specify the resource details that you want to retrieve.

filter

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/authenticationnoauthaction?**filter=property-name1:property-val1,property-name2:property-val2**

Use this query-parameter to get the filtered set of authenticationnoauthaction resources configured on Citrix ADC. Filtering can be done on any of the properties of the resource.

view

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/authenticationnoauthaction?**view=summary**

Note: By default, the retrieved results are displayed in detail view (?view=detail).

pagination

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/authenticationnoauthaction?**pagesize=#no;pageno=#no**

Use this query-parameter to get the authenticationnoauthaction resources in chunks.

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "authenticationnoauthaction": [ {

"name":<String_value>,

"defaultauthenticationgroup":<String_value>

}]}
```

get

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/authenticationnoauthaction/name_value<String>

Query-parameters:

attrs

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/authenticationnoauthaction/name_value<String>?**attrs=property-name1,property-name2**

Use this query parameter to specify the resource details that you want to retrieve.

view

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/authenticationnoauthaction/name_value<String>?**view=summary**

Note: By default, the retrieved results are displayed in detail view (?view=detail).

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "authenticationnoauthaction": [ {

"name":<String_value>,

"defaultauthenticationgroup":<String_value>

}]}
```

count

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/authenticationnoauthaction?**count=yes**

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "authenticationnoauthaction": [ { "__count": "#no"} ] }
```

authenticationpushservice

Configuration for Service details for sending push notifications resource.

Properties

(click to see Operations)

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
name	<String>	Read-write	Name for the push service. Must begin with an ASCII alphanumeric or underscore (_) character, and must contain only ASCII alphanumeric, underscore, hash (#), period (.), space, colon (:), at sign (@), equal sign (=), and hyphen (-) characters. Cannot be changed after the profile is created. CLI Users: If the name includes one or more spaces, enclose the name in double or single quotation marks (for example, "my push service" or 'my push service'). .

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
			Minimum length = 1 Maximum length = 127
clientid	<String>	Read-write	Unique identity for communicating with Citrix Push server in cloud. Minimum length = 1
clientsecret	<String>	Read-write	Unique secret for communicating with Citrix Push server in cloud. Minimum length = 1
customerid	<String>	Read-write	Customer id/name of the account in cloud that is used to create clientid/secret pair. Minimum length = 1
refreshinterval	<Double>	Read-write	Interval at which certificates or idtoken is refreshed. Default value: 50
Namespace	<String>	Read-only	This is fully qualified domain name of the notification service in the cloud. If omitted, namespace defaults to https://mfa.cloud.com/. Minimum length = 1 Maximum length = 63

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
hubname	<String>	Read-only	Name of the hub within a namespace. This is used to classify different identities within a namespace. Minimum length = 1
servicekey	<String>	Read-only	Key to be used to compute signature necessary for registering to notification service. Minimum length = 1
servicekeyname	<String>	Read-only	Friendly name of the Key to be used to compute signature necessary for registering to notification service. Minimum length = 1
certendpoint	<String>	Read-only	URL of the endpoint that contains JWKs (Json Web Key) for JWT (Json Web Token) verification. This is used at cloud instance that offers push service.
pushservicestatus	<String>	Read-only	Describes status of push service. Default value: INIT Possible values = INIT, CERTFETCH, CCTOKEN, COMPLETE

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
trustservice	<String>	Read-only	URL of the service that generates tokens for cloud access. Minimum length = 1
pushcloudserverstatus	<String>	Read-only	Describes status of the cloud service that does push. Possible values = UP, DOWN
signingkeyname	<String>	Read-only	Friendly name of the Key to be used to compute signature necessary for accessing notification service. Minimum length = 1
signingkey	<String>	Read-only	Key to be used to compute signature necessary for accessing notification service. Minimum length = 1
__count	<Double>	Read-only	count parameter

Operations

(click to see Properties)

ADD| DELETE| UPDATE| UNSET| GET (ALL)| GET| COUNT

Some options that you can use for each operations:

- **Getting warnings in response:**NITRO allows you to get warnings in an operation by specifying the "warning" query parameter as "yes". For example, to get warnings while connecting to the Citrix ADC appliance, the URL is as follows:

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/login?warning=yes

If any, the warnings are displayed in the response payload with the HTTP code "209 X-NITRO-WARNING".

- **Authenticated access for individual NITRO operations:**NITRO allows you to logon to the Citrix ADC appliance to perform individual operations. You can use this option instead of creating a NITRO session (using the login object) and then using that session to perform all operations,

  To do this, you must specify the username and password in the request header of the NITRO request as follows:

  X-NITRO-USER:<username>

  X-NITRO-PASS:<password>

  Note: In such cases, make sure that the request header DOES not include the following:

  Cookie:NITRO_AUTH_TOKEN=<tokenvalue>

`*Note:`*

Mandatory parameters are marked in redand placeholder content is marked in <green>.

add

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/authenticationpushservice

HTTP Method:POST

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Content-Type:application/json

Request Payload:

```
{"authenticationpushservice":{

<b>"name":<String_value>,

</b>"clientid":<String_value>,

"clientsecret":<String_value>,
```

```
"customerid":<String_value>,

"refreshinterval":<Double_value>

}}
```

Response:

HTTP Status Code on Success: 201 Created HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error

delete

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/authenticationpushservice/name_value<String>

HTTP Method:DELETE

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue>

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error

update

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/authenticationpushservice

HTTP Method:PUT

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Content-Type:application/json

Request Payload:

```
{"authenticationpushservice":{

<b>"name":<String_value>,

</b>"clientid":<String_value>,

"clientsecret":<String_value>,

"customerid":<String_value>,

"refreshinterval":<Double_value>

}}
```

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error

unset

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/authenticationpushservice?**action=unset**

HTTP Method:POST

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Content-Type:application/json

Request Payload:

```
{"authenticationpushservice":{

<b>"name":<String_value>,

</b>"refreshinterval":true
```

```
}}
```

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error

get (all)

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/authenticationpushservice

Query-parameters:

attrs

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/authenticationpushservice?**attrs=property-name1,property-name2**

Use this query parameter to specify the resource details that you want to retrieve.

filter

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/authenticationpushservice?**filter=property-name1:property-val1,property-name2:property-val2**

Use this query-parameter to get the filtered set of authenticationpushservice resources configured on Citrix ADC. Filtering can be done on any of the properties of the resource.

view

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/authenticationpushservice?**view=summary**

Note: By default, the retrieved results are displayed in detail view (?view=detail).

pagination

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/authenticationpushservice?**pagesize=#no;pageno=#no**

Use this query-parameter to get the authenticationpushservice resources in chunks.

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "authenticationpushservice": [ {

"name":<String_value>,

"Namespace":<String_value>,

"hubname":<String_value>,

"servicekey":<String_value>,

"servicekeyname":<String_value>,

"clientid":<String_value>,

"clientsecret":<String_value>,

"customerid":<String_value>,

"certendpoint":<String_value>,

"refreshinterval":<Double_value>,

"pushservicestatus":<String_value>,

"trustservice":<String_value>,

"pushcloudserverstatus":<String_value>,
```

```
"signingkeyname":<String_value>,

"signingkey":<String_value>

}]}
```

get

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/authenticationpushservice/name_value<String>

Query-parameters:

attrs

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/authenticationpushservice/name_value<String>?**attrs=property-name1,property-name2**

Use this query parameter to specify the resource details that you want to retrieve.

view

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/authenticationpushservice/name_value<String>?**view=summary**

Note: By default, the retrieved results are displayed in detail view (?view=detail).

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "authenticationpushservice": [ {
"name":<String_value>,
"Namespace":<String_value>,
"hubname":<String_value>,
"servicekey":<String_value>,
"servicekeyname":<String_value>,
"clientid":<String_value>,
"clientsecret":<String_value>,
"customerid":<String_value>,
"certendpoint":<String_value>,
"refreshinterval":<Double_value>,
"pushservicestatus":<String_value>,
"trustservice":<String_value>,
"pushcloudserverstatus":<String_value>,
"signingkeyname":<String_value>,
"signingkey":<String_value>
}]}
```

count

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/authenticationpushservice?**count=yes**

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "authenticationpushservice": [ { "__count": "#no"} ] }
```

Authorization

Authorization

Authorization configuration. Authorization services check which resources users are authorized to access, and grant permissions accordingly.

NAME	DESCRIPTION
authorizationaction	Configuration for authorization action
authorizationpolicy	Configuration for authorization policy
authorizationpolicy_aaagroup_binding	Binding object showing the aaagroup that can be bound to authorizationpolicy
authorizationpolicy_aaauser_binding	Binding object showing the aaauser that can be bound to authorizationpolicy
authorizationpolicy_authorizationpolicylabel_binding	Binding object showing the authorizationpolicylabel that can be bound to authorizationpolicy

NAME	DESCRIPTION
authorizationpolicy_binding	Binding object showing the resources that can be bound to authorizationpolicy
authorizationpolicy_csvserver_binding	Binding object showing the csvserver that can be bound to authorizationpolicy
authorizationpolicy_lbvserver_binding	Binding object showing the lbvserver that can be bound to authorizationpolicy
authorizationpolicylabel	Configuration for authorization policy label
authorizationpolicylabel_authorizationpolicy_binding	Binding object showing the authorizationpolicy that can be bound to authorizationpolicylabel
authorizationpolicylabel_binding	Binding object showing the resources that can be bound to authorizationpolicylabel

authorizationaction

Configuration for authorization action resource.

Properties

(click to see Operations)

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
name	<String>	Read-write	Name of authorization action. Minimum length = 1
__count	<Double>	Read-only	count parameter

Operations

(click to see Properties)

GET (ALL)| GET| COUNT

Some options that you can use for each operations:

- **Getting warnings in response:**NITRO allows you to get warnings in an operation by specifying the "warning" query parameter as "yes". For example, to get warnings while connecting to the Citrix ADC appliance, the URL is as follows:

  http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/login?warning=yes

  If any, the warnings are displayed in the response payload with the HTTP code "209 X-NITRO-WARNING".

- **Authenticated access for individual NITRO operations:**NITRO allows you to logon to the Citrix ADC appliance to perform individual operations. You can use this option instead of creating a NITRO session (using the login object) and then using that session to perform all operations,

  To do this, you must specify the username and password in the request header of the NITRO request as follows:

  X-NITRO-USER:<username>

  X-NITRO-PASS:<password>

  Note: In such cases, make sure that the request header DOES not include the following:

  Cookie:NITRO_AUTH_TOKEN=<tokenvalue>

`*Note:`*

Mandatory parameters are marked in redand placeholder content is marked in <green>.

get (all)

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/authorizationaction

Query-parameters:

attrs

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/authorizationaction?**attrs=property-name1,property-name2**

Use this query parameter to specify the resource details that you want to retrieve.

filter

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/authorizationaction?**filter=property-name1:property-val1,property-name2:property-val2**

Use this query-parameter to get the filtered set of authorizationaction resources configured on Citrix ADC. Filtering can be done on any of the properties of the resource.

view

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/authorizationaction?**view=summary**

Note: By default, the retrieved results are displayed in detail view (?view=detail).

pagination

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/authorizationaction?**pagesize=#no;pageno=#no**

Use this query-parameter to get the authorizationaction resources in chunks.

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "authorizationaction": [ {

"name":<String_value>,



}]}
```

get

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/authorizationaction/name_value<String>

Query-parameters:

attrs

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/authorizationaction/name_value<String>?**attrs=property-name1,property-name2**

Use this query parameter to specify the resource details that you want to retrieve.

view

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/authorizationaction/name_value<String>?**view=summary**

Note: By default, the retrieved results are displayed in detail view (?view=detail).

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "authorizationaction": [ {

"name":<String_value>,



}]}
```

count

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/authorizationaction?**count=yes**

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "authorizationaction": [ { "__count": "#no"} ] }
```

authorizationpolicy

Configuration for authorization policy resource.

Properties

(click to see Operations)

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
name	<String>	Read-write	Name for the new authorization policy. Must begin with a letter, number, or the underscore character (_), and

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
			must contain only letters, numbers, and the hyphen (-), period (.) pound (#), space ( ), at (@), equals (=), colon (:), and underscore characters. Cannot be changed after the authorization policy is added. The following requirement applies only to the Citrix ADC CLI: If the name includes one or more spaces, enclose the name in double or single quotation marks (for example, "my authorization policy" or 'my authorization policy'). Minimum length = 1
rule	<String>	Read-write	Name of the Citrix ADC named rule, or an expression, that the policy uses to perform the authentication.

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
action	<String>	Read-write	Action to perform if the policy matches: either allow or deny the request. Minimum length = 1
newname	<String>	Read-write	The new name of the author policy. Minimum length = 1
activepolicy	<Double>	Read-only	.
expressiontype	<String>	Read-only	Type of policy (Classic/Advanced) . Possible values = Classic Policy, Advanced Policy
hits	<Double>	Read-only	Number of hits.
__count	<Double>	Read-only	count parameter

Operations

(click to see Properties)

ADD| DELETE| UPDATE| RENAME| GET (ALL)| GET| COUNT

Some options that you can use for each operations:

- **Getting warnings in response:**NITRO allows you to get warnings in an operation by specifying the "warning" query parameter as "yes". For example, to get warnings while connecting to the Citrix ADC appliance, the URL is as follows:

  http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/login?warning=yes

  If any, the warnings are displayed in the response payload with the HTTP code "209 X-NITRO-WARNING".

- **Authenticated access for individual NITRO operations:**NITRO allows you to logon to the Citrix ADC appliance to perform individual operations. You can use this option instead of

creating a NITRO session (using the login object) and then using that session to perform all operations,

To do this, you must specify the username and password in the request header of the NITRO request as follows:

X-NITRO-USER:<username>

X-NITRO-PASS:<password>

Note: In such cases, make sure that the request header DOES not include the following:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue>

`*Note:`*

Mandatory parameters are marked in redand placeholder content is marked in <green>.

add

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/authorizationpolicy

HTTP Method:POST

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Content-Type:application/json

Request Payload:

```
{"authorizationpolicy":{

<b>"name":<String_value>,

</b><b>"rule":<String_value>,

</b><b>"action":<String_value>

</b>}}
```

Response:

HTTP Status Code on Success: 201 Created HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error

delete

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/authorizationpolicy/name_value<String>

HTTP Method:DELETE

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue>

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error

update

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/authorizationpolicy

HTTP Method:PUT

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Content-Type:application/json

Request Payload:

```
{"authorizationpolicy":{

<b>"name":<String_value>,

</b>"rule":<String_value>,

"action":<String_value>

}}
```

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error

rename

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/authorizationpolicy?**action=rename**

HTTP Method:POST

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Content-Type:application/json

Request Payload:

```
{"authorizationpolicy":{

<b>"name":<String_value>,

</b><b>"newname":<String_value>

</b>}}
```

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error

get (all)

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/authorizationpolicy

Query-parameters:

attrs

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/authorizationpolicy?**attrs=property-name1,property-name2**

Use this query parameter to specify the resource details that you want to retrieve.

filter

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/authorizationpolicy?**filter=property-name1:property-val1,property-name2:property-val2**

Use this query-parameter to get the filtered set of authorizationpolicy resources configured on Citrix ADC. Filtering can be done on any of the properties of the resource.

view

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/authorizationpolicy?**view=summary**

Note: By default, the retrieved results are displayed in detail view (?view=detail).

pagination

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/authorizationpolicy?**pagesize=#no;pageno=#no**

Use this query-parameter to get the authorizationpolicy resources in chunks.

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "authorizationpolicy": [ {

"name":<String_value>,
```

```
"rule":<String_value>,

"action":<String_value>,

"activepolicy":<Double_value>,

"expressiontype":<String_value>,

"hits":<Double_value>

}]}
```

get

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/authorizationpolicy/name_value<String>

Query-parameters:

attrs

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/authorizationpolicy/name_value<String>?**attrs=property-name1,property-name2**

Use this query parameter to specify the resource details that you want to retrieve.

view

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/authorizationpolicy/name_value<String>?**view=summary**

Note: By default, the retrieved results are displayed in detail view (?view=detail).

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "authorizationpolicy": [ {

"name":<String_value>,

"rule":<String_value>,

"action":<String_value>,

"activepolicy":<Double_value>,

"expressiontype":<String_value>,

"hits":<Double_value>

}]}
```

count

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/authorizationpolicy?**count=yes**

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "authorizationpolicy": [ { "__count": "#no"} ] }
```

authorizationpolicylabel

Configuration for authorization policy label resource.

Properties

(click to see Operations)

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
labelname	<String>	Read-write	Name for the new authorization policy label. Must begin with a letter, number, or the underscore character (_), and must contain only letters, numbers, and the hyphen (-), period (.) pound (#), space ( ), at (@), equals (=), colon (:), and underscore characters. Cannot be changed after the authorization policy is created. The following requirement applies only to the Citrix ADC CLI:

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
			If the name includes one or more spaces, enclose the name in double or single quotation marks (for example, "my authorization policy label" or 'authorization policy label').
newname	<String>	Read-write	The new name of the auth policy label. Minimum length = 1
numpol	<Double>	Read-only	Number of polices bound to label.
hits	<Double>	Read-only	Number of times policy label was invoked.
policyname	<String>	Read-only	Name of the authorization policy to bind to the policy label.
priority	<Double>	Read-only	Specifies the priority of the policy.
gotopriorityexpression	<String>	Read-only	Expression specifying the priority of the next policy which will get evaluated if the current policy rule evaluates to TRUE.

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
labeltype	<String>	Read-only	Type of invocation. Available settings function as follows: * reqvserver - Send the request to the specified request virtual server. * resvserver - Send the response to the specified response virtual server. * policylabel - Invoke the specified policy label. Possible values = reqvserver, resvserver, policylabel
invoke_labelname	<String>	Read-only	Name of the policy label to invoke if the current policy evaluates to TRUE, the invoke parameter is set, and Label Type is set to Policy Label.
flowtype	<Double>	Read-only	Flowtype of the bound authorization policy.
description	<String>	Read-only	Description of the policylabel.
__count	<Double>	Read-only	count parameter

Operations

(click to see Properties)

ADD| DELETE| RENAME| GET (ALL)| GET| COUNT

Some options that you can use for each operations:

- **Getting warnings in response:**NITRO allows you to get warnings in an operation by specifying the "warning" query parameter as "yes". For example, to get warnings while connecting to the Citrix ADC appliance, the URL is as follows:

  http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/login?warning=yes

  If any, the warnings are displayed in the response payload with the HTTP code "209 X-NITRO-WARNING".

- **Authenticated access for individual NITRO operations:**NITRO allows you to logon to the Citrix ADC appliance to perform individual operations. You can use this option instead of creating a NITRO session (using the login object) and then using that session to perform all operations,

  To do this, you must specify the username and password in the request header of the NITRO request as follows:

  X-NITRO-USER:<username>

  X-NITRO-PASS:<password>

  Note: In such cases, make sure that the request header DOES not include the following:

  Cookie:NITRO_AUTH_TOKEN=<tokenvalue>

`*Note:`*

Mandatory parameters are marked in redand placeholder content is marked in <green>.

add

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/authorizationpolicylabel

HTTP Method:POST

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Content-Type:application/json

Request Payload:

```
{"authorizationpolicylabel":{

<b>"labelname":<String_value>

</b>}}
```

Response:

HTTP Status Code on Success: 201 Created HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error

delete

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/authorizationpolicylabel/labelname_value<String>

HTTP Method:DELETE

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue>

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error

rename

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/authorizationpolicylabel?**action=rename**

HTTP Method:POST

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Content-Type:application/json

Request Payload:

```
{"authorizationpolicylabel":{

<b>"labelname":<String_value>,

</b><b>"newname":<String_value>

</b>}}
```

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error

get (all)

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/authorizationpolicylabel

Query-parameters:

attrs

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/authorizationpolicylabel?**attrs=property-name1,property-name2**

Use this query parameter to specify the resource details that you want to retrieve.

filter

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/authorizationpolicylabel?**filter=property-name1:property-val1,property-name2:property-val2**

Use this query-parameter to get the filtered set of authorizationpolicylabel resources configured on Citrix ADC. Filtering can be done on any of the properties of the resource.

view

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/authorizationpolicylabel?**view=summary**

Note: By default, the retrieved results are displayed in detail view (?view=detail).

pagination

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/authorizationpolicylabel?**pagesize=#no;pageno=#no**

Use this query-parameter to get the authorizationpolicylabel resources in chunks.

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "authorizationpolicylabel": [ {

"labelname":<String_value>,

"numpol":<Double_value>,

"hits":<Double_value>,

"policyname":<String_value>,

"priority":<Double_value>,

"gotopriorityexpression":<String_value>,

"labeltype":<String_value>,

"invoke_labelname":<String_value>,

"flowtype":<Double_value>,
```

```
"description":<String_value>

}]}
```

get

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/authorizationpolicylabel/labelname_value<String>

Query-parameters:

attrs

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/authorizationpolicylabel/labelname_value<String>?**attrs=property-name1,property-name2**

Use this query parameter to specify the resource details that you want to retrieve.

view

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/authorizationpolicylabel/labelname_value<String>?**view=summary**

Note: By default, the retrieved results are displayed in detail view (?view=detail).

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "authorizationpolicylabel": [ {

"labelname":<String_value>,

"numpol":<Double_value>,

"hits":<Double_value>,

"policyname":<String_value>,

"priority":<Double_value>,

"gotopriorityexpression":<String_value>,

"labeltype":<String_value>,

"invoke_labelname":<String_value>,

"flowtype":<Double_value>,

"description":<String_value>

}]}
```

count

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/authorizationpolicylabel?**count=yes**

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "authorizationpolicylabel": [ { "__count": "#no"} ] }
```

authorizationpolicylabel_authorizationpolicy_binding

Binding object showing the authorizationpolicy that can be bound to authorizationpolicylabel.

Properties

(click to see Operations)

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
priority	<Double>	Read-write	Specifies the priority of the policy.
policyname	<String>	Read-write	Name of the authorization policy to bind to the policy label.
labelname	<String>	Read-write	Name of the authorization policy label to which to bind the policy.
gotopriorityexpression	<String>	Read-write	Expression specifying the priority of the next policy which will get evaluated if the current policy rule evaluates to TRUE.
invoke	<Boolean>	Read-write	If the current policy evaluates to TRUE, terminate evaluation of policies bound to the current policy

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
			label, and then either forward the request or response to the specified virtual server or evaluate the specified policy label.
invoke_labelname	<String>	Read-write	Name of the policy label to invoke if the current policy evaluates to TRUE, the invoke parameter is set, and Label Type is set to Policy Label.
labeltype	<String>	Read-write	Type of invocation. Available settings function as follows: * reqvserver - Send the request to the specified request virtual server. * resvserver - Send the response to the specified response virtual server. * policylabel - Invoke the specified policy label. Possible values = reqvserver, resvserver, policylabel
__count	<Double>	Read-write	count parameter

Operations

(click to see Properties)

ADD:| DELETE:| GET| GET (ALL)| COUNT

Some options that you can use for each operations:

- **Getting warnings in response:**NITRO allows you to get warnings in an operation by specifying the "warning" query parameter as "yes". For example, to get warnings while connecting to the Citrix ADC appliance, the URL is as follows:

  http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/login?warning=yes

  If any, the warnings are displayed in the response payload with the HTTP code "209 X-NITRO-WARNING".

- **Authenticated access for individual NITRO operations:**NITRO allows you to logon to the Citrix ADC appliance to perform individual operations. You can use this option instead of creating a NITRO session (using the login object) and then using that session to perform all operations,

  To do this, you must specify the username and password in the request header of the NITRO request as follows:

  X-NITRO-USER:<username>

  X-NITRO-PASS:<password>

  Note: In such cases, make sure that the request header DOES not include the following:

  Cookie:NITRO_AUTH_TOKEN=<tokenvalue>

`*Note:`*

Mandatory parameters are marked in redand placeholder content is marked in <green>.

add

URL:http://<Citrix-ADC-IP-address(NSIP)/nitro/v1/config/authorizationpolicylabel_authorizationpolicy_binding

HTTP Method:PUT

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Content-Type:application/json

Request Payload:

```
{

"authorizationpolicylabel_authorizationpolicy_binding":{

<b>"labelname":<String_value>,

</b><b>"policyname":<String_value>,

</b><b>"priority":<Double_value>,

</b>"gotopriorityexpression":<String_value>,

"invoke":<Boolean_value>,

"labeltype":<String_value>,

"invoke_labelname":<String_value>

}}
```

Response:

HTTP Status Code on Success: 201 Created HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error

delete

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/authorizationpolicylabel_authorizationpolicy_binding/labelname_value<String>

Query-parameters:

args

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/authorizationpolicylabel_authorizationpolicy_binding/labelname_value<String>?**args=policyname:<String_value>,priority:<Double_value>**

HTTP Method:DELETE

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue>

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error

get

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/authorizationpolicylabel_authorizationpolicy_binding/labelname_value<String>

Query-parameters:

filter

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/authorizationpolicylabel_authorizationpolicy_binding/labelname_value<String>?**filter=property-name1:property-value1,property-name2:property-value2**

Use this query-parameter to get the filtered set of authorizationpolicylabel_authorizationpolicy_binding resources configured on Citrix ADC. Filtering can be done on any of the properties of the resource.

pagination

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/authorizationpolicylabel_authorizationpolicy_binding/labelname_value<String>?**pagesize=#no;pageno=#no**

Use this query-parameter to get the authorizationpolicylabel_authorizationpolicy_binding resources in chunks.

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "authorizationpolicylabel_authorizationpolicy_binding": [ {

"priority":<Double_value>,

"policyname":<String_value>,

"labelname":<String_value>,

"gotopriorityexpression":<String_value>,

"invoke":<Boolean_value>,

"invoke_labelname":<String_value>,

"labeltype":<String_value>

}]}
```

get (all)

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/authorizationpolicylabel_authorizationpolicy_binding

Query-parameters:

bulkbindings

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/authorizationpolicylabel_authorizationpolicy_binding?**bulkbindings=yes**

NITRO allows you to fetch bindings in bulk.

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "authorizationpolicylabel_authorizationpolicy_binding": [ {

"priority":<Double_value>,

"policyname":<String_value>,

"labelname":<String_value>,

"gotopriorityexpression":<String_value>,

"invoke":<Boolean_value>,

"invoke_labelname":<String_value>,

"labeltype":<String_value>

}]}
```

count

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/authorizationpolicylabel_authorizationpolicy_binding/labelname_value<String>?**count=yes**

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{"authorizationpolicylabel_authorizationpolicy_binding": [ { "__count":
"#no"} ] }
```

authorizationpolicylabel_binding

Binding object which returns the resources bound to authorizationpolicylabel.

Properties

(click to see Operations)

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
labelname	<String>	Read-write	Name of the authorization policy label.
authorizationpolicylabel_authorizationpolicy_binding	<authorizationpolicylabel_authorizationpolicy_binding[]>	Read-only	authorizationpolicy that can be bound to authorizationpolicylabel.

Operations

(click to see Properties)

GET| GET (ALL)

Some options that you can use for each operations:

- **Getting warnings in response:**NITRO allows you to get warnings in an operation by specifying the "warning" query parameter as "yes". For example, to get warnings while connecting to the Citrix ADC appliance, the URL is as follows:

  http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/login?warning=yes

  If any, the warnings are displayed in the response payload with the HTTP code "209 X-NITRO-WARNING".

- **Authenticated access for individual NITRO operations:**NITRO allows you to logon to the Citrix ADC appliance to perform individual operations. You can use this option instead of creating a NITRO session (using the login object) and then using that session to perform all operations,

  To do this, you must specify the username and password in the request header of the NITRO request as follows:

  X-NITRO-USER:<username>

  X-NITRO-PASS:<password>

  Note: In such cases, make sure that the request header DOES not include the following:

  Cookie:NITRO_AUTH_TOKEN=<tokenvalue>

`*Note:`*

Mandatory parameters are marked in redand placeholder content is marked in <green>.

get

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/authorizationpolicylabel_binding/labelname_value<String>

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "authorizationpolicylabel_binding": [ {

"labelname":<String_value>,

"authorizationpolicylabel_authorizationpolicy_binding":<authorizationpolic
ylabel_authorizationpolicy_binding[]_value>

}]}
```

get (all)

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/authorizationpolicylabel_binding

Query-parameters:

bulkbindings

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/authorizationpolicylabel_binding?**bulkbindings=yes**

NITRO allows you to fetch bindings in bulk.

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "authorizationpolicylabel_binding": [ {

"labelname":<String_value>,

"authorizationpolicylabel_authorizationpolicy_binding":<authorizationpolic
ylabel_authorizationpolicy_binding[]_value>

}]}
```

authorizationpolicy_aaagroup_binding

Binding object showing the aaagroup that can be bound to authorizationpolicy.

Properties

(click to see Operations)

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
boundto	<String>	Read-write	The entity name to which policy is bound.
name	<String>	Read-write	Name of the authorization policy. Minimum length = 1
priority	<Double>	Read-only	.
__count	<Double>	Read-only	count parameter

Operations

(click to see Properties)

GET| GET (ALL)| COUNT

Some options that you can use for each operations:

- **Getting warnings in response:**NITRO allows you to get warnings in an operation by specifying the "warning" query parameter as "yes". For example, to get warnings while connecting to the Citrix ADC appliance, the URL is as follows:

  http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/login?warning=yes

  If any, the warnings are displayed in the response payload with the HTTP code "209 X-NITRO-WARNING".

- **Authenticated access for individual NITRO operations:**NITRO allows you to logon to the Citrix ADC appliance to perform individual operations. You can use this option instead of creating a NITRO session (using the login object) and then using that session to perform all operations,

  To do this, you must specify the username and password in the request header of the NITRO request as follows:

  X-NITRO-USER:<username>

  X-NITRO-PASS:<password>

  Note: In such cases, make sure that the request header DOES not include the following:

  Cookie:NITRO_AUTH_TOKEN=<tokenvalue>

`*Note:`*

Mandatory parameters are marked in redand placeholder content is marked in <green>.

get

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/authorizationpolicy_aaagroup_binding/name_value<String>

Query-parameters:

filter

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/authorizationpolicy_aaagroup_binding/name_value<String>?**filter=property-name1:property-value1,property-name2:property-value2**

Use this query-parameter to get the filtered set of authorizationpolicy_aaagroup_binding resources configured on Citrix ADC. Filtering can be done on any of the properties of the resource.

pagination

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/authorizationpolicy_aaagroup_binding/name_value<String>?**pagesize=#no;pageno=#no**

Use this query-parameter to get the authorizationpolicy_aaagroup_binding resources in chunks.

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "authorizationpolicy_aaagroup_binding": [ {

"boundto":<String_value>,

"name":<String_value>,

"priority":<Double_value>

}]}
```

get (all)

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/authorizationpolicy_aaagroup_binding

Query-parameters:

bulkbindings

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/authorizationpolicy_aaagroup_binding?**bulkbindings=yes**

NITRO allows you to fetch bindings in bulk.

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "authorizationpolicy_aaagroup_binding": [ {

"boundto":<String_value>,

"name":<String_value>,

"priority":<Double_value>

}]}
```

count

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/authorizationpolicy_aaagroup_binding/name_value<String>?**count=yes**

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{"authorizationpolicy_aaagroup_binding": [ { "__count": "#no"} ] }
```

authorizationpolicy_aaauser_binding

Binding object showing the aaauser that can be bound to authorizationpolicy.

Properties

(click to see Operations)

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
boundto	<String>	Read-write	The entity name to which policy is bound.
name	<String>	Read-write	Name of the authorization policy. Minimum length = 1
priority	<Double>	Read-only	.
__count	<Double>	Read-only	count parameter

Operations

(click to see Properties)

GET| GET (ALL)| COUNT

Some options that you can use for each operations:

- **Getting warnings in response:**NITRO allows you to get warnings in an operation by specifying the "warning" query parameter as "yes". For example, to get warnings while connecting to the Citrix ADC appliance, the URL is as follows:

  http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/login?warning=yes

  If any, the warnings are displayed in the response payload with the HTTP code "209 X-NITRO-WARNING".

- **Authenticated access for individual NITRO operations:**NITRO allows you to logon to the Citrix ADC appliance to perform individual operations. You can use this option instead of creating a NITRO session (using the login object) and then using that session to perform all operations,

  To do this, you must specify the username and password in the request header of the NITRO request as follows:

  X-NITRO-USER:<username>

  X-NITRO-PASS:<password>

  Note: In such cases, make sure that the request header DOES not include the following:

  Cookie:NITRO_AUTH_TOKEN=<tokenvalue>

`*Note:`*

Mandatory parameters are marked in redand placeholder content is marked in <green>.

get

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/authorizationpolicy_aaauser_binding/name_value<String>

Query-parameters:

filter

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/authorizationpolicy_aaauser_binding/name_value<String>?**filter=property-name1:property-value1,property-name2:property-value2**

Use this query-parameter to get the filtered set of authorizationpolicy_aaauser_binding resources configured on Citrix ADC. Filtering can be done on any of the properties of the resource.

pagination

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/authorizationpolicy_aaauser_binding/name_value<String>?**pagesize=#no;pageno=#no**

Use this query-parameter to get the authorizationpolicy_aaauser_binding resources in chunks.

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "authorizationpolicy_aaauser_binding": [ {

"boundto":<String_value>,

"name":<String_value>,

"priority":<Double_value>

}]}
```

get (all)

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/authorizationpolicy_aaauser_binding

Query-parameters:

bulkbindings

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/authorizationpolicy_aaauser_binding?**bulkbindings=yes**

NITRO allows you to fetch bindings in bulk.

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "authorizationpolicy_aaauser_binding": [ {

"boundto":<String_value>,

"name":<String_value>,

"priority":<Double_value>

}]}
```

count

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/authorizationpolicy_aaauser_binding/name_value<String>?**count=yes**

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{"authorizationpolicy_aaauser_binding": [ { "__count": "#no"} ] }
```

authorizationpolicy_authorizationpolicylabel_binding

Binding object showing the authorizationpolicylabel that can be bound to authorizationpolicy.

Properties

(click to see Operations)

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
boundto	<String>	Read-write	The entity name to which policy is bound.
name	<String>	Read-write	Name of the authorization policy. Minimum length = 1
priority	<Double>	Read-only	.
__count	<Double>	Read-only	count parameter

Operations

(click to see Properties)

GET| GET (ALL)| COUNT

Some options that you can use for each operations:

- **Getting warnings in response:**NITRO allows you to get warnings in an operation by specifying the "warning" query parameter as "yes". For example, to get warnings while connecting to the Citrix ADC appliance, the URL is as follows:

  http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/login?warning=yes

  If any, the warnings are displayed in the response payload with the HTTP code "209 X-NITRO-WARNING".

- **Authenticated access for individual NITRO operations:**NITRO allows you to logon to the Citrix ADC appliance to perform individual operations. You can use this option instead of creating a NITRO session (using the login object) and then using that session to perform all operations,

  To do this, you must specify the username and password in the request header of the NITRO request as follows:

  X-NITRO-USER:<username>

  X-NITRO-PASS:<password>

  Note: In such cases, make sure that the request header DOES not include the following:

  Cookie:NITRO_AUTH_TOKEN=<tokenvalue>

`*Note:`*

Mandatory parameters are marked in redand placeholder content is marked in <green>.

get

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/authorizationpolicy_authorizationpolicylabel_binding/name_value<String>

Query-parameters:

filter

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/authorizationpolicy_authorizationpolicylabel_binding/name_value<String>?**filter=property-name1:property-value1,property-name2:property-value2**

Use this query-parameter to get the filtered set of authorizationpolicy_authorizationpolicylabel_binding resources configured on Citrix ADC. Filtering can be done on any of the properties of the resource.

pagination

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/authorizationpolicy_authorizationpolicylabel_binding/name_value<String>?**pagesize=#no;pageno=#no**

Use this query-parameter to get the authorizationpolicy_authorizationpolicylabel_binding resources in chunks.

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "authorizationpolicy_authorizationpolicylabel_binding": [ {

"boundto":<String_value>,

"name":<String_value>,

"priority":<Double_value>

}]}
```

get (all)

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/authorizationpolicy_authorizationpolicylabel_binding

Query-parameters:

bulkbindings

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/authorizationpolicy_authorizationpolicylabel_binding?**bulkbindings=yes**

NITRO allows you to fetch bindings in bulk.

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "authorizationpolicy_authorizationpolicylabel_binding": [ {

"boundto":<String_value>,

"name":<String_value>,

"priority":<Double_value>

}]}
```

count

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/authorizationpolicy_authorizationpolicylabel_binding/name_value<String>?**count=yes**

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{"authorizationpolicy_authorizationpolicylabel_binding": [ { "__count":
"#no"} ] }
```

authorizationpolicy_binding

Binding object which returns the resources bound to authorizationpolicy.

Properties

(click to see Operations)

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
name	<String>	Read-write	Name of the authorization policy. Minimum length = 1
authorizationpolicy_csvserver_binding	<authorizationpolicy_csvserver_binding[]>	Read-only	csvserver that can be bound to authorizationpolicy.
authorizationpolicy_lbvserver_binding	<authorizationpolicy_lbvserver_binding[]>	Read-only	lbvserver that can be bound to authorizationpolicy.
authorizationpolicy_aaauser_binding	<authorizationpolicy_aaauser_binding[]>	Read-only	aaauser that can be bound to authorizationpolicy.

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
authorizationpolicy_authorizationpolicylabel_binding	<authorizationpolicy_authorizationpolicylabel_binding[]>	Read-only	authorizationpolicylabel that can be bound to authorizationpolicy.
authorizationpolicy_aaagroup_binding	<authorizationpolicy_aaagroup_binding[]>	Read-only	aaagroup that can be bound to authorizationpolicy.

Operations

(click to see Properties)

GET| GET (ALL)

Some options that you can use for each operations:

- **Getting warnings in response:**NITRO allows you to get warnings in an operation by specifying the "warning" query parameter as "yes". For example, to get warnings while connecting to the Citrix ADC appliance, the URL is as follows:

  http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/login?warning=yes

  If any, the warnings are displayed in the response payload with the HTTP code "209 X-NITRO-WARNING".

- **Authenticated access for individual NITRO operations:**NITRO allows you to logon to the Citrix ADC appliance to perform individual operations. You can use this option instead of creating a NITRO session (using the login object) and then using that session to perform all operations,

  To do this, you must specify the username and password in the request header of the NITRO request as follows:

  X-NITRO-USER:<username>

  X-NITRO-PASS:<password>

  Note: In such cases, make sure that the request header DOES not include the following:

  Cookie:NITRO_AUTH_TOKEN=<tokenvalue>

`*Note:` *

Mandatory parameters are marked in redand placeholder content is marked in <green>.

get

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/authorizationpolicy_binding/name_value<String>

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "authorizationpolicy_binding": [ {

"name":<String_value>,

"authorizationpolicy_lbvserver_binding":<authorizationpolicy_lbvserver_binding[]_value>,

"authorizationpolicy_aaagroup_binding":<authorizationpolicy_aaagroup_binding[]_value>,

"authorizationpolicy_csvserver_binding":<authorizationpolicy_csvserver_binding[]_value>,

"authorizationpolicy_authorizationpolicylabel_binding":<authorizationpolicy_authorizationpolicylabel_binding[]_value>,

"authorizationpolicy_aaauser_binding":<authorizationpolicy_aaauser_binding[]_value>

}]}
```

get (all)

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/authorizationpolicy_binding

Query-parameters:

bulkbindings

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/authorizationpolicy_binding?**bulkbindings=yes**

NITRO allows you to fetch bindings in bulk.

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "authorizationpolicy_binding": [ {

"name":<String_value>,

"authorizationpolicy_lbvserver_binding":<authorizationpolicy_lbvserver_bin
ding[]_value>,

"authorizationpolicy_aaagroup_binding":<authorizationpolicy_aaagroup_bindi
ng[]_value>,

"authorizationpolicy_csvserver_binding":<authorizationpolicy_csvserver_bin
ding[]_value>,
```

```
"authorizationpolicy_authorizationpolicylabel_binding":<authorizationpolic
y_authorizationpolicylabel_binding[]_value>,

"authorizationpolicy_aaauser_binding":<authorizationpolicy_aaauser_binding
[]_value>

}]}
```

authorizationpolicy_csvserver_binding

Binding object showing the csvserver that can be bound to authorizationpolicy.

Properties

(click to see Operations)

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
boundto	<String>	Read-write	The entity name to which policy is bound.
name	<String>	Read-write	Name of the authorization policy. Minimum length = 1
priority	<Double>	Read-only	.
__count	<Double>	Read-only	count parameter

Operations

(click to see Properties)

GET| GET (ALL)| COUNT

Some options that you can use for each operations:

- **Getting warnings in response:**NITRO allows you to get warnings in an operation by specifying the "warning" query parameter as "yes". For example, to get warnings while connecting to the Citrix ADC appliance, the URL is as follows:

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/login?warning=yes

If any, the warnings are displayed in the response payload with the HTTP code "209 X-NITRO-WARNING".

- **Authenticated access for individual NITRO operations:**NITRO allows you to logon to the Citrix ADC appliance to perform individual operations. You can use this option instead of creating a NITRO session (using the login object) and then using that session to perform all operations,

  To do this, you must specify the username and password in the request header of the NITRO request as follows:

  X-NITRO-USER:<username>

  X-NITRO-PASS:<password>

  Note: In such cases, make sure that the request header DOES not include the following:

  Cookie:NITRO_AUTH_TOKEN=<tokenvalue>

`*Note:`*

Mandatory parameters are marked in redand placeholder content is marked in <green>.

get

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/authorizationpolicy_csvserver_binding/name_value<String>

Query-parameters:

filter

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/authorizationpolicy_csvserver_binding/name_value<String>?**filter=property-name1:property-value1,property-name2:property-value2**

Use this query-parameter to get the filtered set of authorizationpolicy_csvserver_binding resources configured on Citrix ADC. Filtering can be done on any of the properties of the resource.

pagination

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/authorizationpolicy_csvserver_binding/name_value<String>?**pagesize=#no;pageno=#no**

Use this query-parameter to get the authorizationpolicy_csvserver_binding resources in chunks.

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "authorizationpolicy_csvserver_binding": [ {

"boundto":<String_value>,

"name":<String_value>,

"priority":<Double_value>

}]}
```

get (all)

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/authorizationpolicy_csvserver_binding

Query-parameters:

bulkbindings

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/authorizationpolicy_csvserver_binding?**bulkbindings=yes**

NITRO allows you to fetch bindings in bulk.

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "authorizationpolicy_csvserver_binding": [ {

"boundto":<String_value>,

"name":<String_value>,

"priority":<Double_value>

}]}
```

count

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/authorizationpolicy_csvserver_binding/name_value<String>?**count=yes**

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{"authorizationpolicy_csvserver_binding": [ { "__count": "#no"} ] }
```

authorizationpolicy_lbvserver_binding

Binding object showing the lbvserver that can be bound to authorizationpolicy.

Properties

(click to see Operations)

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
boundto	<String>	Read-write	The entity name to which policy is bound.
name	<String>	Read-write	Name of the authorization policy. Minimum length = 1
priority	<Double>	Read-only	.
__count	<Double>	Read-only	count parameter

Operations

(click to see Properties)

GET| GET (ALL)| COUNT

Some options that you can use for each operations:

- **Getting warnings in response:**NITRO allows you to get warnings in an operation by specifying the "warning" query parameter as "yes". For example, to get warnings while connecting to the Citrix ADC appliance, the URL is as follows:

  http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/login?warning=yes

If any, the warnings are displayed in the response payload with the HTTP code "209 X-NITRO-WARNING".

- **Authenticated access for individual NITRO operations:**NITRO allows you to logon to the Citrix ADC appliance to perform individual operations. You can use this option instead of creating a NITRO session (using the login object) and then using that session to perform all operations,

  To do this, you must specify the username and password in the request header of the NITRO request as follows:

  X-NITRO-USER:<username>

  X-NITRO-PASS:<password>

  Note: In such cases, make sure that the request header DOES not include the following:

  Cookie:NITRO_AUTH_TOKEN=<tokenvalue>

`*Note:`*

Mandatory parameters are marked in redand placeholder content is marked in <green>.

get

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/authorizationpolicy_lbvserver_binding/name_value<String>

Query-parameters:

filter

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/authorizationpolicy_lbvserver_binding/name_value<String>?**filter=property-name1:property-value1,property-name2:property-value2**

Use this query-parameter to get the filtered set of authorizationpolicy_lbvserver_binding resources configured on Citrix ADC. Filtering can be done on any of the properties of the resource.

pagination

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/authorizationpolicy_lbvserver_binding/name_value<String>?**pagesize=#no;pageno=#no**

Use this query-parameter to get the authorizationpolicy_lbvserver_binding resources in chunks.

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "authorizationpolicy_lbvserver_binding": [ {

"boundto":<String_value>,

"name":<String_value>,

"priority":<Double_value>

}]}
```

get (all)

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/authorizationpolicy_lbvserver_binding

Query-parameters:

bulkbindings

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/authorizationpolicy_lbvserver_binding?**bulkbindings=yes**

NITRO allows you to fetch bindings in bulk.

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "authorizationpolicy_lbvserver_binding": [ {

"boundto":<String_value>,

"name":<String_value>,

"priority":<Double_value>

}]}
```

count

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/authorizationpolicy_lbvserver_binding/name_value<String>?**count=yes**

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{"authorizationpolicy_lbvserver_binding": [ { "__count": "#no"} ] }
```

Autoscale

Autoscale

AutoScale configuration.

NAME	DESCRIPTION
autoscaleaction	Configuration for autoscale action
autoscalepolicy	Configuration for Autoscale policy
autoscalepolicy_binding	Binding object showing the resources that can be bound to autoscalepolicy
autoscalepolicy_nstimer_binding	Binding object showing the nstimer that can be bound to autoscalepolicy
autoscaleprofile	Configuration for autoscale profile

autoscaleaction

Configuration for autoscale action resource.

Properties

(click to see Operations)

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
name	<String>	Read-write	ActionScale action name. Minimum length = 1
type	<String>	Read-write	The type of action.

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
			Possible values = SCALE_UP, SCALE_DOWN
profilename	<String>	Read-write	AutoScale profile name. Minimum length = 1
parameters	<String>	Read-write	Parameters to use in the action. Minimum length = 1
vmdestroygraceperiod	<Double>	Read-write	Time in minutes a VM is kept in inactive state before destroying. Default value: 10
quiettime	<Double>	Read-write	Time in seconds no other policy is evaluated or action is taken. Default value: 300
vserver	<String>	Read-write	Name of the vserver on which autoscale action has to be taken.
__count	<Double>	Read-only	count parameter

Operations

(click to see Properties)

ADD| DELETE| UPDATE| UNSET| GET (ALL)| GET| COUNT

Some options that you can use for each operations:

- **Getting warnings in response:**NITRO allows you to get warnings in an operation by specifying the "warning" query parameter as "yes". For example, to get warnings while connecting to the Citrix ADC appliance, the URL is as follows:

  http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/login?warning=yes

  If any, the warnings are displayed in the response payload with the HTTP code "209 X-NITRO-WARNING".

- **Authenticated access for individual NITRO operations:**NITRO allows you to logon to the Citrix ADC appliance to perform individual operations. You can use this option instead of creating a NITRO session (using the login object) and then using that session to perform all operations,

  To do this, you must specify the username and password in the request header of the NITRO request as follows:

  X-NITRO-USER:<username>

  X-NITRO-PASS:<password>

  Note: In such cases, make sure that the request header DOES not include the following:

  Cookie:NITRO_AUTH_TOKEN=<tokenvalue>

`*Note:`*

Mandatory parameters are marked in redand placeholder content is marked in <green>.

add

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/autoscaleaction

HTTP Method:POST

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Content-Type:application/json

Request Payload:

```
{"autoscaleaction":{
```

```
<b>"name":<String_value>,

</b><b>"type":<String_value>,

</b><b>"profilename":<String_value>,

</b><b>"parameters":<String_value>,

</b>"vmdestroygraceperiod":<Double_value>,

"quiettime":<Double_value>,

<b>"vserver":<String_value>

</b>}}
```

Response:

HTTP Status Code on Success: 201 Created HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error

delete

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/autoscaleaction/name_value<String>

HTTP Method:DELETE

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue>

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error

update

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/autoscaleaction

HTTP Method:PUT

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Content-Type:application/json

Request Payload:

```
{"autoscaleaction":{

<b>"name":<String_value>,

</b>"profilename":<String_value>,

"parameters":<String_value>,

"vmdestroygraceperiod":<Double_value>,

"quiettime":<Double_value>,

"vserver":<String_value>

}}
```

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error

unset

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/autoscaleaction?**action=unset**

HTTP Method:POST

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Content-Type:application/json

Request Payload:

```
{"autoscaleaction":{

<b>"name":<String_value>,

</b>"vmdestroygraceperiod":true,

"quiettime":true

}}
```

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error

get (all)

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/autoscaleaction

Query-parameters:

attrs

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/autoscaleaction?**attrs=property-name1,property-name2**

Use this query parameter to specify the resource details that you want to retrieve.

filter

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/autoscaleaction?**filter=property-name1:property-val1,property-name2:property-val2**

Use this query-parameter to get the filtered set of autoscaleaction resources configured on Citrix ADC. Filtering can be done on any of the properties of the resource.

view

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/autoscaleaction?**view=summary**

Note: By default, the retrieved results are displayed in detail view (?view=detail).

pagination

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/autoscaleaction?**pagesize=#no;pageno=#no**

Use this query-parameter to get the autoscaleaction resources in chunks.

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "autoscaleaction": [ {

"name":<String_value>,

"type":<String_value>,

"profilename":<String_value>,

"parameters":<String_value>,

"vmdestroygraceperiod":<Double_value>,

"quiettime":<Double_value>,

"vserver":<String_value>

}]}
```

get

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/autoscaleaction/name_value<String>

Query-parameters:

attrs

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/autoscaleaction/name_value<String>?**attrs=property-name1,property-name2**

Use this query parameter to specify the resource details that you want to retrieve.

view

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/autoscaleaction/name_value<String>?**view=summary**

Note: By default, the retrieved results are displayed in detail view (?view=detail).

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "autoscaleaction": [ {

"name":<String_value>,

"type":<String_value>,
```

```
"profilename":<String_value>,

"parameters":<String_value>,

"vmdestroygraceperiod":<Double_value>,

"quiettime":<Double_value>,

"vserver":<String_value>

}]}
```

count

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/autoscaleaction?**count=yes**

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "autoscaleaction": [ { "__count": "#no"} ] }
```

autoscalepolicy

Configuration for Autoscale policy resource.

Properties

(click to see Operations)

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
name	<String>	Read-write	The name of the autoscale policy. Minimum length = 1
rule	<String>	Read-write	The rule associated with the policy.
action	<String>	Read-write	The autoscale profile associated with the policy. Minimum length = 1
comment	<String>	Read-write	Comments associated with this autoscale policy.
logaction	<String>	Read-write	The log action associated with the autoscale policy.
newname	<String>	Read-write	The new name of the autoscale policy. Minimum length = 1
hits	<Double>	Read-only	Number of hits.
undefhits	<Double>	Read-only	Number of Undef hits.
priority	<Double>	Read-only	Specifies the priority of the policy.
__count	<Double>	Read-only	count parameter

Operations

(click to see Properties)

ADD| DELETE| UPDATE| UNSET| RENAME| GET (ALL)| GET| COUNT

Some options that you can use for each operations:

- **Getting warnings in response:**NITRO allows you to get warnings in an operation by specifying the "warning" query parameter as "yes". For example, to get warnings while connecting to the Citrix ADC appliance, the URL is as follows:

  http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/login?warning=yes

  If any, the warnings are displayed in the response payload with the HTTP code "209 X-NITRO-WARNING".

- **Authenticated access for individual NITRO operations:**NITRO allows you to logon to the Citrix ADC appliance to perform individual operations. You can use this option instead of creating a NITRO session (using the login object) and then using that session to perform all operations,

  To do this, you must specify the username and password in the request header of the NITRO request as follows:

  X-NITRO-USER:<username>

  X-NITRO-PASS:<password>

  Note: In such cases, make sure that the request header DOES not include the following:

  Cookie:NITRO_AUTH_TOKEN=<tokenvalue>

`*Note:`*

Mandatory parameters are marked in redand placeholder content is marked in <green>.

add

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/autoscalepolicy

HTTP Method:POST

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Content-Type:application/json

Request Payload:

```
{"autoscalepolicy":{

<b>"name":<String_value>,

</b><b>"rule":<String_value>,

</b><b>"action":<String_value>,

</b>"comment":<String_value>,

"logaction":<String_value>

}}
```

Response:

HTTP Status Code on Success: 201 Created HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error

delete

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/autoscalepolicy/name_value<String>

HTTP Method:DELETE

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue>

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error

update

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/autoscalepolicy

HTTP Method:PUT

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Content-Type:application/json

Request Payload:

```
{"autoscalepolicy":{

<b>"name":<String_value>,

</b>"rule":<String_value>,

"action":<String_value>,

"comment":<String_value>,

"logaction":<String_value>

}}
```

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error

unset

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/autoscalepolicy?**action=unset**

HTTP Method:POST

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Content-Type:application/json

Request Payload:

```
{"autoscalepolicy":{
```

```
<b>"name":<String_value>,

</b>"rule":true,

"action":true,

"comment":true,

"logaction":true

}}
```

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error

rename

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/autoscalepolicy?**action=rename**

HTTP Method:POST

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Content-Type:application/json

Request Payload:

```
{"autoscalepolicy":{

<b>"name":<String_value>,

</b><b>"newname":<String_value>

</b>}}
```

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error

get (all)

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/autoscalepolicy

Query-parameters:

attrs

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/autoscalepolicy?**attrs=property-name1,property-name2**

Use this query parameter to specify the resource details that you want to retrieve.

filter

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/autoscalepolicy?**filter=property-name1:property-val1,property-name2:property-val2**

Use this query-parameter to get the filtered set of autoscalepolicy resources configured on Citrix ADC. Filtering can be done on any of the properties of the resource.

view

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/autoscalepolicy?**view=summary**

Note: By default, the retrieved results are displayed in detail view (?view=detail).

pagination

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/autoscalepolicy?**pagesize=#no;pageno=#no**

Use this query-parameter to get the autoscalepolicy resources in chunks.

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "autoscalepolicy": [ {

"name":<String_value>,

"rule":<String_value>,

"action":<String_value>,

"comment":<String_value>,

"logaction":<String_value>,

"hits":<Double_value>,

"undefhits":<Double_value>,

"priority":<Double_value>

}]}
```

get

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/autoscalepolicy/name_value<String>

Query-parameters:

attrs

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/autoscalepolicy/name_value<String>?**attrs=property-name1,property-name2**

Use this query parameter to specify the resource details that you want to retrieve.

view

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/autoscalepolicy/name_value<String>?**view=summary**

Note: By default, the retrieved results are displayed in detail view (?view=detail).

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "autoscalepolicy": [ {

"name":<String_value>,

"rule":<String_value>,

"action":<String_value>,

"comment":<String_value>,

"logaction":<String_value>,

"hits":<Double_value>,

"undefhits":<Double_value>,

"priority":<Double_value>

}]}
```

count

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/autoscalepolicy?**count=yes**

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "autoscalepolicy": [ { "__count": "#no"} ] }
```

autoscalepolicy_binding

Binding object which returns the resources bound to autoscalepolicy.

Properties

(click to see Operations)

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
name	<String>	Read-write	The name of the autoscale policy. Minimum length = 1
autoscalepolicy_nstimer_binding	<autoscalepolicy_nstimer_binding[]>	Read-only	nstimer that can be bound to autoscalepolicy.

Operations

(click to see Properties)

GET| GET (ALL)

Some options that you can use for each operations:

- **Getting warnings in response:**NITRO allows you to get warnings in an operation by specifying the "warning" query parameter as "yes". For example, to get warnings while connecting to the Citrix ADC appliance, the URL is as follows:

  http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/login?warning=yes

  If any, the warnings are displayed in the response payload with the HTTP code "209 X-NITRO-WARNING".

- **Authenticated access for individual NITRO operations:**NITRO allows you to logon to the Citrix ADC appliance to perform individual operations. You can use this option instead of creating a NITRO session (using the login object) and then using that session to perform all operations,

  To do this, you must specify the username and password in the request header of the NITRO request as follows:

  X-NITRO-USER:<username>

  X-NITRO-PASS:<password>

  Note: In such cases, make sure that the request header DOES not include the following:

  Cookie:NITRO_AUTH_TOKEN=<tokenvalue>

`*Note:`*

Mandatory parameters are marked in redand placeholder content is marked in <green>.

get

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/autoscalepolicy_binding/name_value<String>

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "autoscalepolicy_binding": [ {

"name":<String_value>,

"autoscalepolicy_nstimer_binding":<autoscalepolicy_nstimer_binding[]_value
>

}]}
```

get (all)

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/autoscalepolicy_binding

Query-parameters:

bulkbindings

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/autoscalepolicy_binding?**bulkbindings=yes**

NITRO allows you to fetch bindings in bulk.

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "autoscalepolicy_binding": [ {

"name":<String_value>,

"autoscalepolicy_nstimer_binding":<autoscalepolicy_nstimer_binding[]_value
>

}]}
```

autoscalepolicy_nstimer_binding

Binding object showing the nstimer that can be bound to autoscalepolicy.

Properties

(click to see Operations)

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
boundto	<String>	Read-write	Location where policy is bound.
name	<String>	Read-write	The name of the autoscale policy. Minimum length = 1
priority	<Double>	Read-only	Specifies the priority of the policy.
gotopriorityexpression	<String>	Read-only	Expression specifying the priority of the next policy

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
			which will get evaluated if the current policy rule evaluates to TRUE.
__count	<Double>	Read-only	count parameter

Operations

(click to see Properties)

GET| GET (ALL)| COUNT

Some options that you can use for each operations:

- **Getting warnings in response:**NITRO allows you to get warnings in an operation by specifying the "warning" query parameter as "yes". For example, to get warnings while connecting to the Citrix ADC appliance, the URL is as follows:

  http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/login?warning=yes

  If any, the warnings are displayed in the response payload with the HTTP code "209 X-NITRO-WARNING".

- **Authenticated access for individual NITRO operations:**NITRO allows you to logon to the Citrix ADC appliance to perform individual operations. You can use this option instead of creating a NITRO session (using the login object) and then using that session to perform all operations,

  To do this, you must specify the username and password in the request header of the NITRO request as follows:

  X-NITRO-USER:<username>

  X-NITRO-PASS:<password>

  Note: In such cases, make sure that the request header DOES not include the following:

  Cookie:NITRO_AUTH_TOKEN=<tokenvalue>

`*Note:`*

Mandatory parameters are marked in redand placeholder content is marked in <green>.

get

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/autoscalepolicy_nstimer_binding/name_value<String>

Query-parameters:

filter

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/autoscalepolicy_nstimer_binding/name_value<String>?**filter=property-name1:property-value1,property-name2:property-value2**

Use this query-parameter to get the filtered set of autoscalepolicy_nstimer_binding resources configured on Citrix ADC. Filtering can be done on any of the properties of the resource.

pagination

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/autoscalepolicy_nstimer_binding/name_value<String>?**pagesize=#no;pageno=#no**

Use this query-parameter to get the autoscalepolicy_nstimer_binding resources in chunks.

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "autoscalepolicy_nstimer_binding": [ {

"boundto":<String_value>,

"name":<String_value>,
```

```
"priority":<Double_value>,

"gotopriorityexpression":<String_value>

}]}
```

get (all)

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/autoscalepolicy_nstimer_binding

Query-parameters:

bulkbindings

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/autoscalepolicy_nstimer_binding?**bulkbindings=yes**

NITRO allows you to fetch bindings in bulk.

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "autoscalepolicy_nstimer_binding": [ {

"boundto":<String_value>,

"name":<String_value>,
```

```
"priority":<Double_value>,

"gotopriorityexpression":<String_value>

}]}
```

count

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/autoscalepolicy_nstimer_binding/name_value<String>?**count=yes**

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{"autoscalepolicy_nstimer_binding": [ { "__count": "#no"} ] }
```

autoscaleprofile

Configuration for autoscale profile resource.

Properties

(click to see Operations)

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
name	<String>	Read-write	AutoScale profile name. Minimum length = 1
type	<String>	Read-write	The type of profile. Possible values = CLOUDSTACK
url	<String>	Read-write	URL providing the service. Minimum length = 1
apikey	<String>	Read-write	api key for authentication with service. Minimum length = 1
sharedsecret	<String>	Read-write	shared secret for authentication with service. Minimum length = 1
__count	<Double>	Read-only	count parameter

Operations

(click to see Properties)

ADD| DELETE| UPDATE| GET (ALL)| GET| COUNT

Some options that you can use for each operations:

- **Getting warnings in response:**NITRO allows you to get warnings in an operation by specifying the "warning" query parameter as "yes". For example, to get warnings while connecting to the Citrix ADC appliance, the URL is as follows:

  http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/login?warning=yes

  If any, the warnings are displayed in the response payload with the HTTP code "209 X-NITRO-WARNING".

- **Authenticated access for individual NITRO operations:**NITRO allows you to logon to the Citrix ADC appliance to perform individual operations. You can use this option instead of creating a NITRO session (using the login object) and then using that session to perform all operations,

  To do this, you must specify the username and password in the request header of the NITRO request as follows:

  X-NITRO-USER:<username>

  X-NITRO-PASS:<password>

  Note: In such cases, make sure that the request header DOES not include the following:

  Cookie:NITRO_AUTH_TOKEN=<tokenvalue>

`*Note:`*

Mandatory parameters are marked in redand placeholder content is marked in <green>.

add

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/autoscaleprofile

HTTP Method:POST

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Content-Type:application/json

Request Payload:

```
{"autoscaleprofile":{

<b>"name":<String_value>,

</b><b>"type":<String_value>,

</b><b>"url":<String_value>,

</b><b>"apikey":<String_value>,

</b><b>"sharedsecret":<String_value>
```

```
</b>}}
```

Response:

HTTP Status Code on Success: 201 Created HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error

delete

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/autoscaleprofile/name_value<String>

HTTP Method:DELETE

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue>

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error

update

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/autoscaleprofile

HTTP Method:PUT

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Content-Type:application/json

Request Payload:

```
{"autoscaleprofile":{

<b>"name":<String_value>,

</b>"url":<String_value>,
```

```
"apikey":<String_value>,

"sharedsecret":<String_value>

}}
```

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error

get (all)

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/autoscaleprofile

Query-parameters:

attrs

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/autoscaleprofile?**attrs=property-name1,property-name2**

Use this query parameter to specify the resource details that you want to retrieve.

filter

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/autoscaleprofile?**filter=property-name1:property-val1,property-name2:property-val2**

Use this query-parameter to get the filtered set of autoscaleprofile resources configured on Citrix ADC. Filtering can be done on any of the properties of the resource.

view

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/autoscaleprofile?**view=summary**

Note: By default, the retrieved results are displayed in detail view (?view=detail).

pagination

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/autoscaleprofile?**pagesize=#no;pageno=#no**

Use this query-parameter to get the autoscaleprofile resources in chunks.

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "autoscaleprofile": [ {

"name":<String_value>,

"type":<String_value>,

"url":<String_value>,

"apikey":<String_value>,

"sharedsecret":<String_value>

}]}
```

get

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/autoscaleprofile/name_value<String>

Query-parameters:

attrs

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/autoscaleprofile/name_value<String>?**attrs=property-name1,property-name2**

Use this query parameter to specify the resource details that you want to retrieve.

view

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/autoscaleprofile/name_value<String>?**view=summary**

Note: By default, the retrieved results are displayed in detail view (?view=detail).

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "autoscaleprofile": [ {

"name":<String_value>,

"type":<String_value>,

"url":<String_value>,

"apikey":<String_value>,

"sharedsecret":<String_value>

}]}
```

count

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/autoscaleprofile?**count=yes**

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "autoscaleprofile": [ { "__count": "#no"} ] }
```

Basic

Basic

Basic system configuration.

NAME	DESCRIPTION
dbsmonitors	Configuration for DB monitors
extendedmemoryparam	Configuration for Parameter for extended memory used by LSN and Subscriber Store
location	Configuration for location
locationdata	Configuration for location data

NAME	DESCRIPTION
locationfile	Configuration for location file
locationfile6	Configuration for location file6
locationparameter	Configuration for location parameter
nstrace	Configuration for nstrace operations
radiusnode	Configuration for RADIUS Node
reporting	Configuration for reporting
server	Configuration for server
server_binding	Binding object showing the resources that can be bound to server
server_gslbservice_binding	Binding object showing the gslbservice that can be bound to server
server_gslbservicegroup_binding	Binding object showing the gslbservicegroup that can be bound to server
server_service_binding	Binding object showing the service that can be bound to server
server_servicegroup_binding	Binding object showing the servicegroup that can be bound to server
service	Configuration for service
service_binding	Binding object showing the resources that can be bound to service
service_dospolicy_binding	Binding object showing the dospolicy that can be bound to service
service_lbmonitor_binding	Binding object showing the lbmonitor that can be bound to service
service_scpolicy_binding	Binding object showing the scpolicy that can be bound to service

NAME	DESCRIPTION
servicegroup	Configuration for service group
servicegroup_binding	Binding object showing the resources that can be bound to servicegroup
servicegroup_lbmonitor_binding	Binding object showing the lbmonitor that can be bound to servicegroup
servicegroup_servicegroupentitymonbindings_binding	Binding object showing the servicegroupentitymonbindings that can be bound to servicegroup
servicegroup_servicegroupmember_binding	Binding object showing the servicegroupmember that can be bound to servicegroup
servicegroupbindings	Configuration for servicegroupbind
svcbindings	Configuration for service bindings
vserver	Configuration for virtual server

dbsmonitors

Configuration for DB monitors resource.

Operations

(click to see Properties)

RESTART

Some options that you can use for each operations:

- **Getting warnings in response:**NITRO allows you to get warnings in an operation by specifying the "warning" query parameter as "yes". For example, to get warnings while connecting to the Citrix ADC appliance, the URL is as follows:

  http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/login?warning=yes

  If any, the warnings are displayed in the response payload with the HTTP code "209 X-NITRO-WARNING".

- **Authenticated access for individual NITRO operations:**NITRO allows you to logon to the Citrix ADC appliance to perform individual operations. You can use this option instead of creating a NITRO session (using the login object) and then using that session to perform all operations,

  To do this, you must specify the username and password in the request header of the NITRO request as follows:

  X-NITRO-USER:<username>

  X-NITRO-PASS:<password>

  Note: In such cases, make sure that the request header DOES not include the following:

  Cookie:NITRO_AUTH_TOKEN=<tokenvalue>

`*Note:`*

Mandatory parameters are marked in redand placeholder content is marked in <green>.

restart

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/dbsmonitors?**action=restart**

HTTP Method:POST

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Content-Type:application/json

Request Payload:

```
{"dbsmonitors":{

}}
```

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error

extendedmemoryparam

Configuration for Parameter for extended memory used by LSN and Subscriber Store resource.

Properties

(click to see Operations)

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
memlimit	<Double>	Read-write	Amount of Citrix ADC memory to reserve for the memory used by LSN and Subscriber Session Store feature, in multiples of 2MB. Note: If you later reduce the value of this parameter, the amount of active memory is not reduced. Changing the configured memory limit can only increase the amount of active memory.
memlimitactive	<Double>	Read-only	The active memory limit for extendedmemory on the system. Active memory limit could be different from configured memory limit. This could happen when memory limit could not be increased due to unavailability, or could not be

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
			decreased as it is already in use. This active memory limit configures the current memory limit for LSN and Subscriber Session Store.
maxmemlimit	<Double>	Read-only	The maximum value of memory limit for extendedmemory on the system. Actual available memory may be less. This is maximum memory that can be utilized by LSN and Subscriber Session Store modules.
minrequiredmemory	<Double>	Read-only	The minimum memory requirement for extendedmemory. This is minimum memory required for LSN and Subscriber Session Store Modules.

Operations

(click to see Properties)

UPDATE| UNSET| GET (ALL)

Some options that you can use for each operations:

- **Getting warnings in response:**NITRO allows you to get warnings in an operation by specifying the "warning" query parameter as "yes". For example, to get warnings while connecting to the Citrix ADC appliance, the URL is as follows:

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/login?warning=yes

If any, the warnings are displayed in the response payload with the HTTP code "209 X-NITRO-WARNING".

- **Authenticated access for individual NITRO operations:**NITRO allows you to logon to the Citrix ADC appliance to perform individual operations. You can use this option instead of creating a NITRO session (using the login object) and then using that session to perform all operations,

  To do this, you must specify the username and password in the request header of the NITRO request as follows:

  X-NITRO-USER:<username>

  X-NITRO-PASS:<password>

  Note: In such cases, make sure that the request header DOES not include the following:

  Cookie:NITRO_AUTH_TOKEN=<tokenvalue>

`*Note:`*

Mandatory parameters are marked in redand placeholder content is marked in <green>.

update

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/extendedmemoryparam

HTTP Method:PUT

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Content-Type:application/json

Request Payload:

```
{"extendedmemoryparam":{

"memlimit":<Double_value>

}}
```

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error

unset

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/extendedmemoryparam?**action=unset**

HTTP Method:POST

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Content-Type:application/json

Request Payload:

```
{"extendedmemoryparam":{

"memlimit":true

}}
```

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error

get (all)

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/extendedmemoryparam

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "extendedmemoryparam": [ {

"memlimitactive":<Double_value>,

"maxmemlimit":<Double_value>,

"memlimit":<Double_value>,

"minrequiredmemory":<Double_value>

}]}
```

location

Configuration for location resource.

Properties

(click to see Operations)

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
ipfrom	<String>	Read-write	First IP address in the range, in dotted decimal notation. Minimum length = 1
ipto	<String>	Read-write	Last IP address in the range, in dotted decimal notation.

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
			Minimum length = 1
preferredlocation	<String>	Read-write	String of qualifiers, in dotted notation, describing the geographical location of the IP address range. Each qualifier is more specific than the one that precedes it, as in continent.country.region.city.isp.organization. For example, "NA.US.CA.San Jose.ATT.citrix". Note: A qualifier that includes a dot (.) or space ( ) must be enclosed in double quotation marks. Minimum length = 1
longitude	<Integer>	Read-write	Numerical value, in degrees, specifying the longitude of the geographical location of the IP address-range. Note: Longitude and latitude parameters are used for selecting a service with the static proximity GSLB method. If they are not specified, selection is based

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
			on the qualifiers specified for the location. Minimum value = -180 Maximum value = 180
latitude	<Integer>	Read-write	Numerical value, in degrees, specifying the latitude of the geographical location of the IP address-range. Note: Longitude and latitude parameters are used for selecting a service with the static proximity GSLB method. If they are not specified, selection is based on the qualifiers specified for the location. Minimum value = -90 Maximum value = 90
q1label	<String>	Read-only	Least specific location qualifier.
q2label	<String>	Read-only	Location qualifier 2.

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
q3label	<String>	Read-only	Location qualifier 3.
q4label	<String>	Read-only	Location qualifier 4.
q5label	<String>	Read-only	Location qualifier 5.
q6label	<String>	Read-only	Most specific location qualifier.
__count	<Double>	Read-only	count parameter

Operations

(click to see Properties)

ADD| DELETE| GET (ALL)| GET| COUNT

Some options that you can use for each operations:

- **Getting warnings in response:**NITRO allows you to get warnings in an operation by specifying the "warning" query parameter as "yes". For example, to get warnings while connecting to the Citrix ADC appliance, the URL is as follows:

  http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/login?warning=yes

  If any, the warnings are displayed in the response payload with the HTTP code "209 X-NITRO-WARNING".

- **Authenticated access for individual NITRO operations:**NITRO allows you to logon to the Citrix ADC appliance to perform individual operations. You can use this option instead of creating a NITRO session (using the login object) and then using that session to perform all operations,

  To do this, you must specify the username and password in the request header of the NITRO request as follows:

  X-NITRO-USER:<username>

  X-NITRO-PASS:<password>

  Note: In such cases, make sure that the request header DOES not include the following:

  Cookie:NITRO_AUTH_TOKEN=<tokenvalue>

`*Note:` *

Mandatory parameters are marked in redand placeholder content is marked in <green>.

add

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/location

HTTP Method:POST

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Content-Type:application/json

Request Payload:

```
{"location":{

<b>"ipfrom":<String_value>,

</b><b>"ipto":<String_value>,

</b><b>"preferredlocation":<String_value>,

</b>"longitude":<Integer_value>,

"latitude":<Integer_value>

}}
```

Response:

HTTP Status Code on Success: 201 Created HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error

delete

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/location/ipfrom_value<String>

HTTP Method:DELETE

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue>

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error

get (all)

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/location

Query-parameters:

attrs

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/location?**attrs=property-name1,property-name2**

Use this query parameter to specify the resource details that you want to retrieve.

filter

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/location?**filter=property-name1:property-val1,property-name2:property-val2**

Use this query-parameter to get the filtered set of location resources configured on Citrix ADC. Filtering can be done on any of the properties of the resource.

view

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/location?**view=summary**

Note: By default, the retrieved results are displayed in detail view (?view=detail).

pagination

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/location?**pagesize=#no;pageno=#no**

Use this query-parameter to get the location resources in chunks.

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "location": [ {

"ipfrom":<String_value>,

"ipto":<String_value>,

"preferredlocation":<String_value>,

"q1label":<String_value>,

"q2label":<String_value>,

"q3label":<String_value>,

"q4label":<String_value>,

"q5label":<String_value>,

"q6label":<String_value>,

"longitude":<Integer_value>,

"latitude":<Integer_value>

}]}
```

get

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/location/ipfrom_value<String>

Query-parameters:

attrs

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/location/ipfrom_value<String>?**attrs=property-name1,property-name2**

Use this query parameter to specify the resource details that you want to retrieve.

view

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/location/ipfrom_value<String>?**view=summary**

Note: By default, the retrieved results are displayed in detail view (?view=detail).

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "location": [ {

"ipfrom":<String_value>,

"ipto":<String_value>,

"preferredlocation":<String_value>,

"q1label":<String_value>,

"q2label":<String_value>,
```

```
"q3label":<String_value>,

"q4label":<String_value>,

"q5label":<String_value>,

"q6label":<String_value>,

"longitude":<Integer_value>,

"latitude":<Integer_value>

}]}
```

count

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/location?**count=yes**

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "location": [ { "__count": "#no"} ] }
```

locationdata

Configuration for location data resource.

Operations

(click to see Properties)

CLEAR

Some options that you can use for each operations:

- **Getting warnings in response:**NITRO allows you to get warnings in an operation by specifying the "warning" query parameter as "yes". For example, to get warnings while connecting to the Citrix ADC appliance, the URL is as follows:

  http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/login?warning=yes

  If any, the warnings are displayed in the response payload with the HTTP code "209 X-NITRO-WARNING".

- **Authenticated access for individual NITRO operations:**NITRO allows you to logon to the Citrix ADC appliance to perform individual operations. You can use this option instead of creating a NITRO session (using the login object) and then using that session to perform all operations,

  To do this, you must specify the username and password in the request header of the NITRO request as follows:

  X-NITRO-USER:<username>

  X-NITRO-PASS:<password>

  Note: In such cases, make sure that the request header DOES not include the following:

  Cookie:NITRO_AUTH_TOKEN=<tokenvalue>

`*Note:`*

Mandatory parameters are marked in redand placeholder content is marked in <green>.

clear

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/locationdata?**action=clear**

HTTP Method:POST

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Content-Type:application/json

Request Payload:

```
{"locationdata":{

}}
```

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error

locationfile

Configuration for location file resource.

Properties

(click to see Operations)

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
Locationfile	<String>	Read-write	Name of the location file, with or without absolute path. If the path is not included, the default path (/var/netscaler/locdb) is assumed. In a high availability setup, the static database must be stored in the same location on both Citrix ADCs. Minimum length = 1
format	<String>	Read-write	Format of the location file. Required for the Citrix ADC to identify

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
			how to read the location file. Default value: Citrix ADC Possible values = netscaler, ip-country, ip-country-isp, ip-country-region-city, ip-country-region-city-isp, geoip-country, geoip-region, geoip-city, geoip-country-org, geoip-country-isp, geoip-city-isp-org
src	<String>	Read-write	URL \(protocol, host, path, and file name\) from where the location file will be imported. NOTE: The import fails if the object to be imported is on an HTTPS server that requires client certificate authentication for access. Minimum length = 1 Maximum length = 2047

Operations

(click to see Properties)

ADD| DELETE| IMPORT| GET (ALL)

Some options that you can use for each operations:

- **Getting warnings in response:**NITRO allows you to get warnings in an operation by specifying the "warning" query parameter as "yes". For example, to get warnings while connecting to the Citrix ADC appliance, the URL is as follows:

  http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/login?warning=yes

  If any, the warnings are displayed in the response payload with the HTTP code "209 X-NITRO-WARNING".

- **Authenticated access for individual NITRO operations:**NITRO allows you to logon to the Citrix ADC appliance to perform individual operations. You can use this option instead of creating a NITRO session (using the login object) and then using that session to perform all operations,

  To do this, you must specify the username and password in the request header of the NITRO request as follows:

  X-NITRO-USER:<username>

  X-NITRO-PASS:<password>

  Note: In such cases, make sure that the request header DOES not include the following:

  Cookie:NITRO_AUTH_TOKEN=<tokenvalue>

`*Note:`*

Mandatory parameters are marked in redand placeholder content is marked in <green>.

add

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/locationfile

HTTP Method:POST

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Content-Type:application/json

Request Payload:

```
{"locationfile":{

<b>"Locationfile":<String_value>,

</b>"format":<String_value>

}}
```

Response:

HTTP Status Code on Success: 201 Created HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error

delete

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/locationfile

HTTP Method:DELETE

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue>

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error

Import

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/locationfile?**action=Import**

HTTP Method:POST

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Content-Type:application/json

Request Payload:

```
{"locationfile":{

<b>"Locationfile":<String_value>,

</b><b>"src":<String_value>

</b>}}
```

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error

get (all)

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/locationfile

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "locationfile": [ {

"Locationfile":<String_value>,
```

```
"format":<String_value>

}]}
```

locationfile6

Configuration for location file6 resource.

Properties

(click to see Operations)

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
Locationfile	<String>	Read-write	Name of the IPv6 location file, with or without absolute path. If the path is not included, the default path (/var/netscaler/locdb) is assumed. In a high availability setup, the static database must be stored in the same location on both Citrix ADCs. Minimum length = 1
format	<String>	Read-write	Format of the IPv6 location file. Required for the Citrix ADC to identify how to read the location file. Default value: netscaler6

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
			Possible values = netscaler6, geoip-country6
src	<String>	Read-write	URL \(protocol, host, path, and file name\) from where the location file will be imported. NOTE: The import fails if the object to be imported is on an HTTPS server that requires client certificate authentication for access. Minimum length = 1 Maximum length = 2047
__count	<Double>	Read-only	count parameter

Operations

(click to see Properties)

ADD| DELETE| IMPORT| GET (ALL)| COUNT

Some options that you can use for each operations:

- **Getting warnings in response:**NITRO allows you to get warnings in an operation by specifying the "warning" query parameter as "yes". For example, to get warnings while connecting to the Citrix ADC appliance, the URL is as follows:

  http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/login?warning=yes

If any, the warnings are displayed in the response payload with the HTTP code "209 X-NITRO-WARNING".

- **Authenticated access for individual NITRO operations:**NITRO allows you to logon to the Citrix ADC appliance to perform individual operations. You can use this option instead of creating a NITRO session (using the login object) and then using that session to perform all operations,

  To do this, you must specify the username and password in the request header of the NITRO request as follows:

  X-NITRO-USER:<username>

  X-NITRO-PASS:<password>

  Note: In such cases, make sure that the request header DOES not include the following:

  Cookie:NITRO_AUTH_TOKEN=<tokenvalue>

`*Note:`*

Mandatory parameters are marked in redand placeholder content is marked in <green>.

add

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/locationfile6

HTTP Method:POST

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Content-Type:application/json

Request Payload:

```
{"locationfile6":{

<b>"Locationfile":<String_value>,

</b>"format":<String_value>

}}
```

Response:

HTTP Status Code on Success: 201 Created HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error

delete

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/locationfile6

HTTP Method:DELETE

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue>

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error

Import

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/locationfile6?**action=Import**

HTTP Method:POST

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Content-Type:application/json

Request Payload:

```
{"locationfile6":{

<b>"Locationfile":<String_value>,

</b><b>"src":<String_value>

</b>}}
```

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error

get (all)

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/locationfile6

Query-parameters:

attrs

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/locationfile6?**attrs=property-name1,property-name2**

Use this query parameter to specify the resource details that you want to retrieve.

filter

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/locationfile6?**filter=property-name1:property-val1,property-name2:property-val2**

Use this query-parameter to get the filtered set of locationfile6 resources configured on Citrix ADC. Filtering can be done on any of the properties of the resource.

view

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/locationfile6?**view=summary**

Note: By default, the retrieved results are displayed in detail view (?view=detail).

pagination

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/locationfile6?**pagesize=#no;pageno=#no**

Use this query-parameter to get the locationfile6 resources in chunks.

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "locationfile6": [ {

"Locationfile":<String_value>,

"format":<String_value>

}]}
```

count

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/locationfile6?**count=yes**

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "locationfile6": [ { "__count": "#no"} ] }
```

locationparameter

Configuration for location parameter resource.

Properties

(click to see Operations)

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
context	<String>	Read-write	Context for describing locations. In geographic context, qualifier labels are assigned by default in the following sequence: Continent.Country.Region.City.ISP.Organization. In custom context, the qualifiers labels can have any meaning that you designate. Possible values = geographic, custom
q1label	<String>	Read-write	Label specifying the meaning of the first qualifier. Can be specified for custom context only. Minimum length = 1
q2label	<String>	Read-write	Label specifying the meaning of the second qualifier. Can be specified for custom context only. Minimum length = 1

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
q3label	<String>	Read-write	Label specifying the meaning of the third qualifier. Can be specified for custom context only. Minimum length = 1
q4label	<String>	Read-write	Label specifying the meaning of the fourth qualifier. Can be specified for custom context only. Minimum length = 1
q5label	<String>	Read-write	Label specifying the meaning of the fifth qualifier. Can be specified for custom context only. Minimum length = 1
q6label	<String>	Read-write	Label specifying the meaning of the sixth qualifier. Can be specified for custom context only. Minimum length = 1
Locationfile	<String>	Read-only	Currently loaded location database file.
format	<String>	Read-only	Location file format. Possible values = netscaler, ip-country, ip-country-isp, ip-country-region-city, ip-country-region-city-isp, geoip-country,

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
			geoip-region, geoip-city, geoip-country-org, geoip-country-isp, geoip-city-isp-org
custom	<Double>	Read-only	Number of configured custom locations.
Static	<Double>	Read-only	Number of configured locations in the database file (static locations).
lines	<Double>	Read-only	Number of lines in the databse files.
errors	<Double>	Read-only	Number of errros encountered while reading the database file.
warnings	<Double>	Read-only	Number of warnings encountered while reading the database file.
entries	<Double>	Read-only	Number of successfully added entries.
locationfile6	<String>	Read-only	Currently loaded location database file.
format6	<String>	Read-only	Location file format. Possible values = netscaler6, geoip-country6

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
custom6	<Double>	Read-only	Number of configured custom locations.
static6	<Double>	Read-only	Number of configured locations in the database file (static locations).
lines6	<Double>	Read-only	Number of lines in the databse files.
errors6	<Double>	Read-only	Number of errros encountered while reading the database file.
warnings6	<Double>	Read-only	Number of warnings encountered while reading the database file.
entries6	<Double>	Read-only	Number of successfully added entries.
flags	<Double>	Read-only	Information needed for display. This argument passes information from the kernel to the user space.
status	<Double>	Read-only	This argument displays when the status (success or failure) of database loading.

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
databasemode	<String>	Read-only	This argument displays the database mode. Possible values = File, Internal, Not applicable
flushing	<String>	Read-only	This argument displays the state of flushing. Possible values = In progress, Idle
loading	<String>	Read-only	This argument displays the state of loading. Possible values = In progress, Idle
matchwildcardtoany	<String>	Read-only	Indicates whether wildcard qualifiers should match any other qualifier including non-wildcard. Option to fallback after default behavior changes to not match. Default value: NO Possible values = YES, NO

Operations

(click to see Properties)

UPDATE| UNSET| GET (ALL)

Some options that you can use for each operations:

- **Getting warnings in response:**NITRO allows you to get warnings in an operation by specifying the "warning" query parameter as "yes". For example, to get warnings while connecting to the Citrix ADC appliance, the URL is as follows:

  http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/login?warning=yes

  If any, the warnings are displayed in the response payload with the HTTP code "209 X-NITRO-WARNING".

- **Authenticated access for individual NITRO operations:**NITRO allows you to logon to the Citrix ADC appliance to perform individual operations. You can use this option instead of creating a NITRO session (using the login object) and then using that session to perform all operations,

  To do this, you must specify the username and password in the request header of the NITRO request as follows:

  X-NITRO-USER:<username>

  X-NITRO-PASS:<password>

  Note: In such cases, make sure that the request header DOES not include the following:

  Cookie:NITRO_AUTH_TOKEN=<tokenvalue>

`*Note:`*

Mandatory parameters are marked in redand placeholder content is marked in <green>.

update

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/locationparameter

HTTP Method:PUT

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Content-Type:application/json

Request Payload:

```
{"locationparameter":{
```

```
"context":<String_value>,

"q1label":<String_value>,

"q2label":<String_value>,

"q3label":<String_value>,

"q4label":<String_value>,

"q5label":<String_value>,

"q6label":<String_value>

}}
```

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error

unset

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/locationparameter?**action=unset**

HTTP Method:POST

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Content-Type:application/json

Request Payload:

```
{"locationparameter":{

"context":true,

"q1label":true,
```

```
"q2label":true,

"q3label":true,

"q4label":true,

"q5label":true,

"q6label":true

}}
```

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error

get (all)

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/locationparameter

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "locationparameter": [ {
```

```
"context":<String_value>,

"q1label":<String_value>,

"q2label":<String_value>,

"q3label":<String_value>,

"q4label":<String_value>,

"q5label":<String_value>,

"q6label":<String_value>,

"Locationfile":<String_value>,

"format":<String_value>,

"custom":<Double_value>,

"Static":<Double_value>,

"lines":<Double_value>,

"errors":<Double_value>,

"warnings":<Double_value>,

"entries":<Double_value>,

"locationfile6":<String_value>,

"format6":<String_value>,

"custom6":<Double_value>,

"static6":<Double_value>,

"lines6":<Double_value>,

"errors6":<Double_value>,

"warnings6":<Double_value>,
```

```
"entries6":<Double_value>,

"flags":<Double_value>,

"status":<Double_value>,

"databasemode":<String_value>,

"flushing":<String_value>,

"loading":<String_value>,

"matchwildcardtoany":<String_value>

}]}
```

nstrace

Configuration for nstrace operations resource.

Properties

(click to see Operations)

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
nf	<Double>	Read-write	Number of files to be generated in cycle. Default value: 24 Minimum value = 1 Maximum value = 100
time	<Double>	Read-write	Time per file (sec). Default value: 3600 Minimum value = 1

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
size	<Double>	Read-write	Size of the captured data. Set 0 for full packet trace. Default value: 164 Minimum value = 0 Maximum value = 1514
mode	<String[]>	Read-write	Capturing mode for trace. Mode can be any of the following values or combination of these values: RX Received packets before NIC pipelining (Filter does not work when RX capturing mode is ON) NEW_RX Received packets after NIC pipelining TX Transmitted packets TXB Packets buffered for transmission IPV6 Translated IPv6 packets C2C Capture C2C message

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
			NS_FR_TX TX/TXB packets are not captured in flow receiver. MPTCP MPTCP master flow Default mode: NEW_RX TXB . Default value: DEFAULT_MODE Possible values = TX, TXB, RX, IPV6, NEW_RX, C2C, NS_FR_TX, APPFW, MPTCP, PolicyBased
pernic	<String>	Read-write	Use separate trace files for each interface. Works only with cap format. Default value: DISABLED Possible values = ENABLED, DISABLED
filename	<String>	Read-write	Name of the trace file.
fileid	<String>	Read-write	ID for the trace file name for uniqueness. Should be used only with -name option.

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
filter	<String>	Read-write	Filter expression for nstrace. Maximum length of filter is 255 and it can be of following format: <expression> [<relop> <expression>] <relop> = (;; | ||) <expression> =: CONNECTION.<qualifier>.<qualifier-method>.(<qualifier-value>) <qualifier> = SRCIP <qualifier-method> = [EQ | NE] <qualifier-value> = A valid IPv4 address. example = CONNECTION.SRCIP.EQ(127.0.0.1) <qualifier> = DSTIP <qualifier-method> = [EQ | NE] <qualifier-value> = A valid IPv4 address.

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
			example = CONNECTION.DSTIP.EQ(127.0.0.1) <qualifier> = IP <qualifier-method> = [EQ | NE] <qualifier-value> = A valid IPv4 address. example = CONNECTION.IP.EQ(127.0.0.1) <qualifier> = SRCIPv6 <qualifier-method> = [EQ | NE] <qualifier-value> = A valid IPv6 address. example = CONNECTION.SRCIPv6.EQ(2001:db8:0:0:1::1) <qualifier> = DSTIPv6 <qualifier-method> = [EQ | NE]

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
			<qualifier-value> = A valid IPv6 address. example = CONNECTION.DSTIPv6.EQ(2001:db8:0:0:1::1) <qualifier> = IPv6 <qualifier-method> = [EQ | NE] <qualifier-value> = A valid IPv6 address. example = CONNECTION.IPv6.EQ(2001:db8:0:0:1::1) <qualifier> = SRCPORT <qualifier-method> = [EQ | NE | GT | GE | LT | LE | BETWEEN] <qualifier-value> = A valid port number. example = CONNECTION.SRCPORT.EQ(80)

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
			<qualifier> = DSTPORT <qualifier-method> = [EQ | NE | GT | GE | LT | LE | BETWEEN] <qualifier-value> = A valid port number. example = CONNECTION.DSTPORT.EQ(80) <qualifier> = PORT <qualifier-method> = [EQ | NE | GT | GE | LT | LE | BETWEEN] <qualifier-value> = A valid port number. example = CONNECTION.PORT.EQ(80) <qualifier> = VLANID <qualifier-method> = [EQ | NE | GT | GE | LT | LE | BETWEEN]

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
			<qualifier-value> = A valid VLAN ID. example = CONNECTION.VLANID.EQ(0) <qualifier> = CONNID <qualifier-method> = [EQ | NE | GT | GE | LT | LE | BETWEEN] <qualifier-value> = A valid PCB dev number. example = CONNECTION.CONNID.EQ(0) <qualifier> = PPEID <qualifier-method> = [EQ | NE | GT | GE | LT | LE | BETWEEN] <qualifier-value> = A valid core ID. example = CONNECTION.PPEID.EQ(0)

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
			<qualifier> = SVCNAME <qualifier-method> = [EQ | NE | CONTAINS | STARTSWITH | ENDSWITH] <qualifier-value> = A valid text string. example = CONNECTION.SVCNAME.EQ("name") <qualifier> = LB_VSERVER.NAME <qualifier-method> = [EQ | NE | CONTAINS | STARTSWITH | ENDSWITH] <qualifier-value> = LB vserver name. example = CONNECTION.LB_VSERVER.NAME.EQ("name")

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
			<qualifier> = CS_VSERVER.NAME <qualifier-method> = [EQ | NE | CONTAINS | STARTSWITH | ENDSWITH] <qualifier-value> = CS vserver name. example = CONNECTION.CS_VSERVER.NAME.EQ("name") <qualifier> = INTF <qualifier-method> = [EQ | NE] <qualifier-value> = A valid interface id in the form of x/y. example = CONNECTION.INTF.EQ("x/y") <qualifier> = SERVICE_TYPE <qualifier-method> = [EQ | NE]

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
			<qualifier-value> = (SVC_HTTP | FTP | TCP | UDP | SSL | SSL_BRIDGE | SSL_TCP | NNTP | RPCSVR | RPCSVRS | RPCCLNT | SVC_DNS | ADNS | SNMP | RTSP | DHCPRA | ANY| MONITOR | MONITOR_UDP | MONITOR_PING | SIP_UDP | SVC_MYSQL | SVC_MSSQL | FIX | SSL_FIX | PKTSTEER | SVC_AAA | SERVICE_UNKNOWN) example = CONNECTION.SERVICE_TYPE.EQ(ANY) <qualifier> = TRAFFIC_DOMAIN_ID

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
			<qualifier-method> = [EQ | NE | GT | GE | LT | LE | BETWEEN] <qualifier-value> = A valid traffic domain ID. example = CONNECTION.TRAFFIC_DOMAIN_ID.EQ(0) eg: start nstrace -filter "CONNECTION.SRCIP.EQ(127.0.0.1) || (CONNECTION.SVCNAME.NE("s1") ;; CONNECTION.SRCPORT.EQ(80))" The filter expression should be given in double quotes. common use cases: Trace capturing full sized traffic from/to ip 10.102.44.111, excluding loopback traffic start nstrace -size 0 -filter "CONNECTION.IP.

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
			NE(127.0.0.1) ;; CONNECTION.IP.EQ(10.102.44.111)" Trace capturing all traffic to (terminating at) port 80 or 443 start nstrace -size 0 -filter "CONNECTION.DSTPORT.EQ(443) || CONNECTION.DSTPORT.EQ(80)" Trace capturing all backend traffic specific to service service1 along with corresponding client side traffic start nstrace -size 0 -filter "CONNECTION.SVCNAME.EQ("service1")" -link ENABLED Trace capturing all traffic through Citrix ADC interface 1/1 start nstrace -filter "CONNECTION.INTF.EQ("1/1")"

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
			Trace capturing all traffic specific through vlan 2 start nstrace -filter "CONNECTION.VLANID.EQ(2)" Trace capturing all frontend (client side) traffic specific to lb vserver vserver1 along with corresponding server side traffic start nstrace -size 0 -filter "CONNECTION.LB_VSERVER.NAME.EQ("vserver1")" -link ENABLED .
link	<String>	Read-write	Includes filtered connection's peer traffic. Default value: DISABLED Possible values = ENABLED, DISABLED
nodes	<Double[]>	Read-write	Nodes on which tracing is started. Minimum value = 0 Maximum value = 32

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
filesize	<Double>	Read-write	File size, in MB, treshold for rollover. If free disk space is less than 2GB at the time of rollover, trace will stop . Default value: 1024 Minimum value = 0 Maximum value = 10240
traceformat	<String>	Read-write	Format in which trace will be generated . Default value: 0 Possible values = NSCAP, PCAP
merge	<String>	Read-write	Specify how traces across PE's are merged . Default value: 0 Possible values = ONSTOP, ONTHEFLY, NOMERGE
doruntimecleanup	<String>	Read-write	Enable or disable runtime temp file cleanup.

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
			Default value: ENABLED Possible values = ENABLED, DISABLED
tracebuffers	<Double>	Read-write	Number of 16KB trace buffers. Default value: 5000 Minimum value = 1000
skiprpc	<String>	Read-write	skip RPC packets. Default value: DISABLED Possible values = ENABLED, DISABLED
skiplocalssh	<String>	Read-write	skip local SSH packets. Default value: DISABLED Possible values = ENABLED, DISABLED
capsslkeys	<String>	Read-write	Capture SSL Master keys. Master keys will not be captured on FIPS machine. Warning: The captured keys can be used to decrypt information that

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
			may be confidential. The captured key files have to be stored in a secure environment. Default value: DISABLED Possible values = ENABLED, DISABLED
capdroppkt	<String>	Read-write	Captures Dropped Packets if set to ENABLED. Default value: DISABLED Possible values = ENABLED, DISABLED
inmemorytrace	<String>	Read-write	Logs packets in appliance's memory and dumps the trace file on stopping the nstrace operation. Default value: DISABLED Possible values = ENABLED, DISABLED
nodeid	<Double>	Read-write	Unique number that identifies the cluster node. Minimum value = 0

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
			Maximum value = 31
state	<String>	Read-only	Current running state of trace. Default value: 0 Possible values = RUNNING, STOPPED
scope	<String>	Read-only	Scope of started trace, local or cluster level. Default value: 0 Possible values = CLUSTER, LOCAL
tracelocation	<String>	Read-only	Directory where current trace files are saved.

Operations

(click to see Properties)

GET (ALL)

Some options that you can use for each operations:

- **Getting warnings in response:**NITRO allows you to get warnings in an operation by specifying the "warning" query parameter as "yes". For example, to get warnings while connecting to the Citrix ADC appliance, the URL is as follows:

  http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/login?warning=yes

  If any, the warnings are displayed in the response payload with the HTTP code "209 X-NITRO-WARNING".

- **Authenticated access for individual NITRO operations:**NITRO allows you to logon to the Citrix ADC appliance to perform individual operations. You can use this option instead of creating a NITRO session (using the login object) and then using that session to perform all operations,

  To do this, you must specify the username and password in the request header of the NITRO request as follows:

  X-NITRO-USER:<username>

  X-NITRO-PASS:<password>

  Note: In such cases, make sure that the request header DOES not include the following:

  Cookie:NITRO_AUTH_TOKEN=<tokenvalue>

`*Note:`*

Mandatory parameters are marked in redand placeholder content is marked in <green>.

get (all)

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/nstrace

Query-parameters:

args

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/nstrace?**args=nodeid:<Double_value>**

Use this query-parameter to get nstrace resources based on additional properties.

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "nstrace": [ {

nodeid:<Double_value>"state":<String_value>,

"scope":<String_value>,

"tracelocation":<String_value>,

"nf":<Double_value>,

"time":<Double_value>,

"size":<Double_value>,

"mode":<String[]_value>,

"traceformat":<String_value>,

"pernic":<String_value>,

"filename":<String_value>,

"fileid":<String_value>,

"filter":<String_value>,

"link":<String_value>,

"nodes":<Double[]_value>,

"merge":<String_value>,

"doruntimecleanup":<String_value>,

"tracebuffers":<Double_value>,

"skiprpc":<String_value>,

"skiplocalssh":<String_value>,
```

```
"capsslkeys":<String_value>,

"capdroppkt":<String_value>,

"inmemorytrace":<String_value>

}]}
```

radiusnode

Configuration for RADIUS Node resource.

Properties

(click to see Operations)

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
nodeprefix	<String>	Read-write	IP address/IP prefix of radius node in CIDR format.
radkey	<String>	Read-write	The key shared between the RADIUS server and clients. Required for Citrix ADC to communicate with the RADIUS nodes.
__count	<Double>	Read-only	count parameter

Operations

(click to see Properties)

ADD| UPDATE| DELETE| GET (ALL)| GET| COUNT

Some options that you can use for each operations:

- **Getting warnings in response:**NITRO allows you to get warnings in an operation by specifying the "warning" query parameter as "yes". For example, to get warnings while connecting to the Citrix ADC appliance, the URL is as follows:

  http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/login?warning=yes

  If any, the warnings are displayed in the response payload with the HTTP code "209 X-NITRO-WARNING".

- **Authenticated access for individual NITRO operations:**NITRO allows you to logon to the Citrix ADC appliance to perform individual operations. You can use this option instead of creating a NITRO session (using the login object) and then using that session to perform all operations,

  To do this, you must specify the username and password in the request header of the NITRO request as follows:

  X-NITRO-USER:<username>

  X-NITRO-PASS:<password>

  Note: In such cases, make sure that the request header DOES not include the following:

  Cookie:NITRO_AUTH_TOKEN=<tokenvalue>

`*Note:`*

Mandatory parameters are marked in redand placeholder content is marked in <green>.

add

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/radiusnode

HTTP Method:POST

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Content-Type:application/json

Request Payload:

```
{"radiusnode":{
```

```
<b>"nodeprefix":<String_value>,

</b><b>"radkey":<String_value>

</b>}}
```

Response:

HTTP Status Code on Success: 201 Created HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error

update

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/radiusnode

HTTP Method:PUT

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Content-Type:application/json

Request Payload:

```
{"radiusnode":{

<b>"nodeprefix":<String_value>,

</b>"radkey":<String_value>

}}
```

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error

delete

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/radiusnode/nodeprefix_value<String>

HTTP Method:DELETE

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue>

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error

get (all)

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/radiusnode

Query-parameters:

attrs

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/radiusnode?**attrs=property-name1,property-name2**

Use this query parameter to specify the resource details that you want to retrieve.

filter

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/radiusnode?**filter=property-name1:property-val1,property-name2:property-val2**

Use this query-parameter to get the filtered set of radiusnode resources configured on Citrix ADC. Filtering can be done on any of the properties of the resource.

view

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/radiusnode?**view=summary**

Note: By default, the retrieved results are displayed in detail view (?view=detail).

pagination

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/radiusnode?**pagesize=#no;pageno=#no**

Use this query-parameter to get the radiusnode resources in chunks.

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "radiusnode": [ {

"nodeprefix":<String_value>,

"radkey":<String_value>

}]}
```

get

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/radiusnode/nodeprefix_value<String>

Query-parameters:

attrs

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/radiusnode/nodeprefix_value<String>?**attrs=property-name1,property-name2**

Use this query parameter to specify the resource details that you want to retrieve.

view

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/radiusnode/nodeprefix_value<String>?**view=summary**

Note: By default, the retrieved results are displayed in detail view (?view=detail).

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "radiusnode": [ {

"nodeprefix":<String_value>,

"radkey":<String_value>

}]}
```

count

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/radiusnode?**count=yes**

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "radiusnode": [ { "__count": "#no"} ] }
```

reporting

Configuration for reporting resource.

Properties

(click to see Operations)

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
state	<String>	Read-only	The rule associated with the entity. Possible values = ENABLED, DISABLED

Operations

(click to see Properties)

ENABLE| DISABLE| GET (ALL)

Some options that you can use for each operations:

- **Getting warnings in response:**NITRO allows you to get warnings in an operation by specifying the "warning" query parameter as "yes". For example, to get warnings while connecting to the Citrix ADC appliance, the URL is as follows:

  http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/login?warning=yes

If any, the warnings are displayed in the response payload with the HTTP code "209 X-NITRO-WARNING".

- **Authenticated access for individual NITRO operations:**NITRO allows you to logon to the Citrix ADC appliance to perform individual operations. You can use this option instead of creating a NITRO session (using the login object) and then using that session to perform all operations,

  To do this, you must specify the username and password in the request header of the NITRO request as follows:

  X-NITRO-USER:<username>

  X-NITRO-PASS:<password>

  Note: In such cases, make sure that the request header DOES not include the following:

  Cookie:NITRO_AUTH_TOKEN=<tokenvalue>

`*Note:`*

Mandatory parameters are marked in redand placeholder content is marked in <green>.

enable

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/reporting?**action=enable**

HTTP Method:POST

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Content-Type:application/json

Request Payload:

```
{"reporting":{

}}
```

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error

disable

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/reporting?**action=disable**

HTTP Method:POST

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Content-Type:application/json

Request Payload:

```
{"reporting":{

}}
```

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error

get (all)

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/reporting

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "reporting": [ {

"state":<String_value>

}]}
```

server

Configuration for server resource.

Properties

(click to see Operations)

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
name	<String>	Read-write	Name for the server. Must begin with an ASCII alphabetic or underscore (_) character, and must contain only ASCII alphanumeric, underscore, hash (#), period (.), space, colon (:), at (@), equals (=), and hyphen (-) characters. Can be changed after the name is created.

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
			Minimum length = 1
ipaddress	<String>	Read-write	IPv4 or IPv6 address of the server. If you create an IP address based server, you can specify the name of the server, instead of its IP address, when creating a service. Note: If you do not create a server entry, the server IP address that you enter when you create a service becomes the name of the server.
domain	<String>	Read-write	Domain name of the server. For a domain based configuration, you must create the server first. Minimum length = 1
translationip	<String>	Read-write	IP address used to transform the server's DNS-resolved IP address.
translationmask	<String>	Read-write	The netmask of the translation ip.
domainresolveretry	<Integer>	Read-write	Time, in seconds, for which the Citrix ADC must wait, after DNS resolution fails, before sending the next DNS query to

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
			resolve the domain name. Default value: 5 Minimum value = 5 Maximum value = 20939
state	<String>	Read-write	Initial state of the server. Default value: ENABLED Possible values = ENABLED, DISABLED
ipv6address	<String>	Read-write	Support IPv6 addressing mode. If you configure a server with the IPv6 addressing mode, you cannot use the server in the IPv4 addressing mode. Default value: NO Possible values = YES, NO
comment	<String>	Read-write	Any information about the server.
td	<Double>	Read-write	Integer value that uniquely identifies the traffic domain in which you want to configure the entity. If you do not specify an ID, the entity

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
			becomes part of the default traffic domain, which has an ID of 0. Minimum value = 0 Maximum value = 4094
querytype	<String>	Read-write	Specify the type of DNS resolution to be done on the configured domain to get the backend services. Valid query types are A, AAAA and SRV with A being the default querytype. The type of DNS resolution done on the domains in SRV records is inherited from ipv6 argument. Default value: A Possible values = A, AAAA, SRV
domainresolvenow	<Boolean>	Read-write	Immediately send a DNS query to resolve the server's domain name.
delay	<Double>	Read-write	Time, in seconds, after which all the services configured on the server are disabled.

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
graceful	<String>	Read-write	Shut down gracefully, without accepting any new connections, and disabling each service when all of its connections are closed. Default value: NO Possible values = YES, NO
Internal	<Boolean>	Read-write	Display names of the servers that have been created for internal use.
newname	<String>	Read-write	New name for the server. Must begin with an ASCII alphabetic or underscore (_) character, and must contain only ASCII alphanumeric, underscore, hash (#), period (.), space, colon (:), at (@), equals (=), and hyphen (-) characters. Minimum length = 1
statechangetimesec	<String>	Read-only	Time when last state change happened. Seconds part.
tickssincelaststatechange	<Double>	Read-only	Time in 10 millisecond ticks

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
			since the last state change.
autoscale	<String>	Read-only	Auto scale option for a servicegroup. Default value: DISABLED Possible values = DISABLED, DNS, POLICY, CLOUD
usip	<String>	Read-only	Use the client's IP address as the source IP address when initiating a connection to the server. When creating a service, if you do not set this parameter, the service inherits the global Use Source IP setting (available in the enable ns mode and disable ns mode CLI commands, or in the System > Settings > Configure modes > Configure Modes dialog box). However, you can override this setting after you create the service. Possible values = YES, NO

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
cka	<String>	Read-only	Enable client keep-alive for the service group. Possible values = YES, NO
tcpb	<String>	Read-only	Enable TCP buffering for the service group. Possible values = YES, NO
cmp	<String>	Read-only	Enable compression for the specified service. Possible values = YES, NO
cacheable	<String>	Read-only	Use the transparent cache redirection virtual server to forward the request to the cache server. Default value: NO Possible values = YES, NO
sc	<String>	Read-only	State of the SureConnect feature for the service group. Default value: OFF Possible values = ON, OFF
sp	<String>	Read-only	Enable surge protection for the service group.

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
			Default value: OFF Possible values = ON, OFF
__count	<Double>	Read-only	count parameter

Operations

(click to see Properties)

ADD| DELETE| UPDATE| UNSET| ENABLE| DISABLE| RENAME| GET (ALL)| GET| COUNT

Some options that you can use for each operations:

- **Getting warnings in response:**NITRO allows you to get warnings in an operation by specifying the "warning" query parameter as "yes". For example, to get warnings while connecting to the Citrix ADC appliance, the URL is as follows:

  http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/login?warning=yes

  If any, the warnings are displayed in the response payload with the HTTP code "209 X-NITRO-WARNING".

- **Authenticated access for individual NITRO operations:**NITRO allows you to logon to the Citrix ADC appliance to perform individual operations. You can use this option instead of creating a NITRO session (using the login object) and then using that session to perform all operations,

  To do this, you must specify the username and password in the request header of the NITRO request as follows:

  X-NITRO-USER:<username>

  X-NITRO-PASS:<password>

  Note: In such cases, make sure that the request header DOES not include the following:

  Cookie:NITRO_AUTH_TOKEN=<tokenvalue>

`*Note:`*

Mandatory parameters are marked in redand placeholder content is marked in <green>.

add

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/server

HTTP Method:POST

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Content-Type:application/json

Request Payload:

```
{"server":{

<b>"name":<String_value>,

</b>"ipaddress":<String_value>,

"domain":<String_value>,

"translationip":<String_value>,

"translationmask":<String_value>,

"domainresolveretry":<Integer_value>,

"state":<String_value>,

"ipv6address":<String_value>,

"comment":<String_value>,

"td":<Double_value>,

"querytype":<String_value>

}}
```

Response:

HTTP Status Code on Success: 201 Created HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error

delete

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/server/name_value<String>

HTTP Method:DELETE

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue>

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error

update

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/server

HTTP Method:PUT

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Content-Type:application/json

Request Payload:

```
{"server":{

<b>"name":<String_value>,

</b>"ipaddress":<String_value>,

"domainresolveretry":<Integer_value>,

"translationip":<String_value>,

"translationmask":<String_value>,
```

```
"domainresolvenow":<Boolean_value>,

"comment":<String_value>

}}
```

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error

unset

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/server?**action=unset**

HTTP Method:POST

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Content-Type:application/json

Request Payload:

```
{"server":{

<b>"name":<String_value>,

</b>"comment":true

}}
```

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error

enable

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/server?**action=enable**

HTTP Method:POST

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Content-Type:application/json

Request Payload:

```
{"server":{

<b>"name":<String_value>

</b>}}
```

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error

disable

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/server?**action=disable**

HTTP Method:POST

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Content-Type:application/json

Request Payload:

```
{"server":{

<b>"name":<String_value>,
```

```
</b>"delay":<Double_value>,

"graceful":<String_value>

}}
```

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error

rename

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/server?**action=rename**

HTTP Method:POST

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Content-Type:application/json

Request Payload:

```
{"server":{

<b>"name":<String_value>,

</b><b>"newname":<String_value>

</b>}}
```

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error

get (all)

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/server

Query-parameters:

args

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/server?**args=name:<String_value>,Internal:<Boolean_value>**

Use this query-parameter to get server resources based on additional properties.

attrs

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/server?**attrs=property-name1,property-name2**

Use this query parameter to specify the resource details that you want to retrieve.

filter

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/server?**filter=property-name1:property-val1,property-name2:property-val2**

Use this query-parameter to get the filtered set of server resources configured on Citrix ADC. Filtering can be done on any of the properties of the resource.

view

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/server?**view=summary**

Note: By default, the retrieved results are displayed in detail view (?view=detail).

pagination

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/server?**pagesize=#no;pageno=#no**

Use this query-parameter to get the server resources in chunks.

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "server": [ {

name:<String_value>,Internal:<Boolean_value>"ipaddress":<String_value>,

"state":<String_value>,

"domain":<String_value>,

"domainresolveretry":<Integer_value>,

"translationip":<String_value>,

"translationmask":<String_value>,

"comment":<String_value>,

"statechangetimesec":<String_value>,

"tickssincelaststatechange":<Double_value>,

"ipv6address":<String_value>,

"td":<Double_value>,

"autoscale":<String_value>,

"usip":<String_value>,

"cka":<String_value>,

"tcpb":<String_value>,

"cmp":<String_value>,
```

```
"cacheable":<String_value>,

"sc":<String_value>,

"sp":<String_value>,

"querytype":<String_value>

}]}
```

get

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/server/name_value<String>

Query-parameters:

attrs

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/server/name_value<String>?**attrs=property-name1,property-name2**

Use this query parameter to specify the resource details that you want to retrieve.

view

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/server/name_value<String>?**view=summary**

Note: By default, the retrieved results are displayed in detail view (?view=detail).

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "server": [ {

name:<String_value>,Internal:<Boolean_value>"ipaddress":<String_value>,

"state":<String_value>,

"domain":<String_value>,

"domainresolveretry":<Integer_value>,

"translationip":<String_value>,

"translationmask":<String_value>,

"comment":<String_value>,

"statechangetimesec":<String_value>,

"tickssincelaststatechange":<Double_value>,

"ipv6address":<String_value>,

"td":<Double_value>,

"autoscale":<String_value>,

"usip":<String_value>,

"cka":<String_value>,

"tcpb":<String_value>,

"cmp":<String_value>,

"cacheable":<String_value>,

"sc":<String_value>,

"sp":<String_value>,

"querytype":<String_value>
```

```
}]}
```

count

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/server?**count=yes**

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "server": [ { "__count": "#no"} ] }
```

server_binding

Binding object which returns the resources bound to server.

Properties

(click to see Operations)

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
name	<String>	Read-write	Name of the server for which to display parameters. Minimum length = 1

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
server_gslbservicegroup_binding	<server_gslbservicegroup_binding[]>	Read-only	gslbservicegroup that can be bound to server.
server_service_binding	<server_service_binding[]>	Read-only	service that can be bound to server.
server_gslbservice_binding	<server_gslbservice_binding[]>	Read-only	gslbservice that can be bound to server.
server_servicegroup_binding	<server_servicegroup_binding[]>	Read-only	servicegroup that can be bound to server.

Operations

(click to see Properties)

GET| GET (ALL)

Some options that you can use for each operations:

- **Getting warnings in response:**NITRO allows you to get warnings in an operation by specifying the "warning" query parameter as "yes". For example, to get warnings while connecting to the Citrix ADC appliance, the URL is as follows:

  http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/login?warning=yes

  If any, the warnings are displayed in the response payload with the HTTP code "209 X-NITRO-WARNING".

- **Authenticated access for individual NITRO operations:**NITRO allows you to logon to the Citrix ADC appliance to perform individual operations. You can use this option instead of creating a NITRO session (using the login object) and then using that session to perform all operations,

  To do this, you must specify the username and password in the request header of the NITRO request as follows:

  X-NITRO-USER:<username>

  X-NITRO-PASS:<password>

Note: In such cases, make sure that the request header DOES not include the following:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue>

`*Note:` *

Mandatory parameters are marked in redand placeholder content is marked in <green>.

get

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/server_binding/name_value<String>

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "server_binding": [ {

"name":<String_value>,

"server_gslbservicegroup_binding":<server_gslbservicegroup_binding[]_value
>,

"server_servicegroup_binding":<server_servicegroup_binding[]_value>,

"server_service_binding":<server_service_binding[]_value>,

"server_gslbservice_binding":<server_gslbservice_binding[]_value>

}]}
```

get (all)

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/server_binding

Query-parameters:

bulkbindings

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/server_binding?**bulkbindings=yes**

NITRO allows you to fetch bindings in bulk.

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "server_binding": [ {

"name":<String_value>,

"server_gslbservicegroup_binding":<server_gslbservicegroup_binding[]_value
>,

"server_servicegroup_binding":<server_servicegroup_binding[]_value>,

"server_service_binding":<server_service_binding[]_value>,

"server_gslbservice_binding":<server_gslbservice_binding[]_value>
```

```
}]}
```

server_gslbservice_binding

Binding object showing the gslbservice that can be bound to server.

Properties

(click to see Operations)

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
name	<String>	Read-write	Name of the server for which to display parameters. Minimum length = 1
servicename	<String>	Read-write	The services attatched to the server.
serviceipstr	<String>	Read-only	This field has been intorduced to show the dbs services ip.
svctype	<String>	Read-only	The type of bound service. Possible values = HTTP, FTP, TCP, UDP, SSL, SSL_BRIDGE, SSL_TCP, DTLS, NNTP, RPCSVR, DNS, ADNS, SNMP, RTSP, DHCPRA, ANY, SIP_UDP, SIP_TCP, SIP_SSL, DNS_TCP, ADNS_TCP,

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
			MYSQL, MSSQL, ORACLE, RADIUS, RADIUSListener, RDP, DIAMETER, SSL_DIAMETER, TFTP, SMPP, PPTP, GRE, SYSLOGTCP, SYSLOGUDP, FIX, SSL_FIX, USER_TCP, USER_SSL_TCP, QUIC, IPFIX, LOGSTREAM
svrstate	<String>	Read-only	The state of the bound service. Possible values = UP, DOWN, UNKNOWN, BUSY, OUT OF SERVICE, GOING OUT OF SERVICE, DOWN WHEN GOING OUT OF SERVICE, NS_EMPTY_STR, Unknown, DISABLED
port	<Integer>	Read-only	The port number to be used for the bound service. Range 1 - 65535 * in CLI is represented as 65535 in NITRO API

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
serviceipaddress	<String>	Read-only	The IP address of the bound service.
__count	<Double>	Read-only	count parameter

Operations

(click to see Properties)

GET| GET (ALL)| COUNT

Some options that you can use for each operations:

- **Getting warnings in response:**NITRO allows you to get warnings in an operation by specifying the "warning" query parameter as "yes". For example, to get warnings while connecting to the Citrix ADC appliance, the URL is as follows:

  http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/login?warning=yes

  If any, the warnings are displayed in the response payload with the HTTP code "209 X-NITRO-WARNING".

- **Authenticated access for individual NITRO operations:**NITRO allows you to logon to the Citrix ADC appliance to perform individual operations. You can use this option instead of creating a NITRO session (using the login object) and then using that session to perform all operations,

  To do this, you must specify the username and password in the request header of the NITRO request as follows:

  X-NITRO-USER:<username>

  X-NITRO-PASS:<password>

  Note: In such cases, make sure that the request header DOES not include the following:

  Cookie:NITRO_AUTH_TOKEN=<tokenvalue>

`*Note:`*

Mandatory parameters are marked in redand placeholder content is marked in <green>.

get

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/server_gslbservice_binding/name_value<String>

Query-parameters:

filter

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/server_gslbservice_binding/name_value<String>?**filter=property-name1:property-value1,property-name2:property-value2**

Use this query-parameter to get the filtered set of server_gslbservice_binding resources configured on Citrix ADC. Filtering can be done on any of the properties of the resource.

pagination

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/server_gslbservice_binding/name_value<String>?**pagesize=#no;pageno=#no**

Use this query-parameter to get the server_gslbservice_binding resources in chunks.

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "server_gslbservice_binding": [ {

"name":<String_value>,

"servicename":<String_value>,
```

```
"serviceipstr":<String_value>,

"svctype":<String_value>,

"svrstate":<String_value>,

"port":<Integer_value>,

"serviceipaddress":<String_value>

}]}
```

get (all)

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/server_gslbservice_binding

Query-parameters:

bulkbindings

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/server_gslbservice_binding?**bulkbindings=yes**

NITRO allows you to fetch bindings in bulk.

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "server_gslbservice_binding": [ {

"name":<String_value>,

"servicename":<String_value>,

"serviceipstr":<String_value>,

"svctype":<String_value>,

"svrstate":<String_value>,

"port":<Integer_value>,

"serviceipaddress":<String_value>

}]}
```

count

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/server_gslbservice_binding/name_value<String>?**count=yes**

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{"server_gslbservice_binding": [ { "__count": "#no"} ] }
```

server_gslbservicegroup_binding

Binding object showing the gslbservicegroup that can be bound to server.

Properties

(click to see Operations)

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
name	<String>	Read-write	Name of the server for which to display parameters. Minimum length = 1
servicegroupname	<String>	Read-write	servicegroups bind to this server.
svrtimeout	<Double>	Read-only	Time, in seconds, after which to terminate an idle server connection. Minimum value = 0 Maximum value = 31536000
serviceipstr	<String>	Read-only	This field has been intorduced to show the dbs services ip.
cip	<String>	Read-only	Before forwarding a request to the service, insert an HTTP header with the client's IPv4 or IPv6 address as its value. Used if the

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
			server needs the client's IP address for security, accounting, or other purposes, and setting the Use Source IP parameter is not a viable option. Possible values = ENABLED, DISABLED
dup_port	<Integer>	Read-only	port of the service. Range 1 - 65535 * in CLI is represented as 65535 in NITRO API
customserverid	<String>	Read-only	A positive integer to identify the service. Used when the persistency type is set to Custom Server ID. Default value: "None"
boundtd	<Double>	Read-only	Integer value that uniquely identifies the traffic domain in which you want to configure the entity. If you do not specify an ID, the entity becomes part of the default traffic domain, which has an ID of 0.

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
			Minimum value = 0 Maximum value = 4094
downstateflush	<String>	Read-only	Perform delayed clean-up of connections to all services in the service group. Default value: ENABLED Possible values = ENABLED, DISABLED
svctype	<String>	Read-only	The type of bound service. Possible values = HTTP, FTP, TCP, UDP, SSL, SSL_BRIDGE, SSL_TCP, DTLS, NNTP, RPCSVR, DNS, ADNS, SNMP, RTSP, DHCPRA, ANY, SIP_UDP, SIP_TCP, SIP_SSL, DNS_TCP, ADNS_TCP, MYSQL, MSSQL, ORACLE, RADIUS, RADIUSListener, RDP, DIAMETER, SSL_DIAMETER, TFTP, SMPP, PPTP, GRE, SYSLOGTCP,

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
			SYSLOGUDP, FIX, SSL_FIX, USER_TCP, USER_SSL_TCP, QUIC, IPFIX, LOGSTREAM
svrstate	<String>	Read-only	The state of the bound service. Possible values = UP, DOWN, UNKNOWN, BUSY, OUT OF SERVICE, GOING OUT OF SERVICE, DOWN WHEN GOING OUT OF SERVICE, NS_EMPTY_STR, Unknown, DISABLED
appflowlog	<String>	Read-only	Enable logging of AppFlow information for the specified service group. Default value: ENABLED Possible values = ENABLED, DISABLED
cipheader	<String>	Read-only	Name of the HTTP header whose value must be set to the IP address of the client. Used with the Client IP parameter. If client

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
			IP insertion is enabled, and the client IP header is not specified, the value of Client IP Header parameter or the value set by the set ns config command is used as client's IP header name. Minimum length = 1
maxclient	<Double>	Read-only	Maximum number of simultaneous open connections for the service group. Minimum value = 0 Maximum value = 4294967294
clttimeout	<Double>	Read-only	Time, in seconds, after which to terminate an idle client connection. Minimum value = 0 Maximum value = 31536000
port	<Integer>	Read-only	The port number to be used for the bound service. Range 1 - 65535 * in CLI is represented as 65535 in NITRO API

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
dup_svctype	<String>	Read-only	service type of the service. Possible values = HTTP, FTP, TCP, UDP, SSL, SSL_BRIDGE, SSL_TCP, DTLS, NNTP, RPCSVR, DNS, ADNS, SNMP, RTSP, DHCPRA, ANY, SIP_UDP, SIP_TCP, SIP_SSL, DNS_TCP, ADNS_TCP, MYSQL, MSSQL, ORACLE, RADIUS, RADIUSListener, RDP, DIAMETER, SSL_DIAMETER, TFTP, SMPP, PPTP, GRE, SYSLOGTCP, SYSLOGUDP, FIX, SSL_FIX, USER_TCP, USER_SSL_TCP, QUIC, IPFIX, LOGSTREAM
monthreshold	<Double>	Read-only	Minimum sum of weights of the monitors that are bound to this service. Used to determine whether to mark a service as UP or DOWN.

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
			Minimum value = 0 Maximum value = 65535
maxreq	<Double>	Read-only	Maximum number of requests that can be sent on a persistent connection to the service group. Note: Connection requests beyond this value are rejected. Minimum value = 0 Maximum value = 65535
serviceipaddress	<String>	Read-only	The IP address of the bound service.
maxbandwidth	<Double>	Read-only	Maximum bandwidth, in Kbps, allocated for all the services in the service group. Minimum value = 0 Maximum value = 4294967287
svrcfgflags	<Double>	Read-only	service flags to denote its a db enabled.
__count	<Double>	Read-only	count parameter

Operations

(click to see Properties)

GET| GET (ALL)| COUNT

Some options that you can use for each operations:

- **Getting warnings in response:**NITRO allows you to get warnings in an operation by specifying the "warning" query parameter as "yes". For example, to get warnings while connecting to the Citrix ADC appliance, the URL is as follows:

  http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/login?warning=yes

  If any, the warnings are displayed in the response payload with the HTTP code "209 X-NITRO-WARNING".

- **Authenticated access for individual NITRO operations:**NITRO allows you to logon to the Citrix ADC appliance to perform individual operations. You can use this option instead of creating a NITRO session (using the login object) and then using that session to perform all operations,

  To do this, you must specify the username and password in the request header of the NITRO request as follows:

  X-NITRO-USER:<username>

  X-NITRO-PASS:<password>

  Note: In such cases, make sure that the request header DOES not include the following:

  Cookie:NITRO_AUTH_TOKEN=<tokenvalue>

`*Note:`*

Mandatory parameters are marked in redand placeholder content is marked in <green>.

get

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/server_gslbservicegroup_binding/name_value<String>

Query-parameters:

filter

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/server_gslbservicegroup_binding/name_value<String>?**filter=property-name1:property-value1,property-name2:property-value2**

Use this query-parameter to get the filtered set of server_gslbservicegroup_binding resources configured on Citrix ADC. Filtering can be done on any of the properties of the resource.

pagination

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/server_gslbservicegroup_binding/name_value<String>?**pagesize=#no;pageno=#no**

Use this query-parameter to get the server_gslbservicegroup_binding resources in chunks.

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "server_gslbservicegroup_binding": [ {

"name":<String_value>,

"servicegroupname":<String_value>,

"svrtimeout":<Double_value>,

"serviceipstr":<String_value>,

"cip":<String_value>,

"dup_port":<Integer_value>,

"customserverid":<String_value>,

"boundtd":<Double_value>,

"downstateflush":<String_value>,
```

```
"svctype":<String_value>,

"svrstate":<String_value>,

"appflowlog":<String_value>,

"cipheader":<String_value>,

"maxclient":<Double_value>,

"clttimeout":<Double_value>,

"port":<Integer_value>,

"dup_svctype":<String_value>,

"monthreshold":<Double_value>,

"maxreq":<Double_value>,

"serviceipaddress":<String_value>,

"maxbandwidth":<Double_value>,

"svrcfgflags":<Double_value>

}]}
```

get (all)

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/server_gslbservicegroup_binding

Query-parameters:

bulkbindings

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/server_gslbservicegroup_binding?**bulkbindings=yes**

NITRO allows you to fetch bindings in bulk.

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "server_gslbservicegroup_binding": [ {

"name":<String_value>,

"servicegroupname":<String_value>,

"svrtimeout":<Double_value>,

"serviceipstr":<String_value>,

"cip":<String_value>,

"dup_port":<Integer_value>,

"customserverid":<String_value>,

"boundtd":<Double_value>,

"downstateflush":<String_value>,

"svctype":<String_value>,

"svrstate":<String_value>,

"appflowlog":<String_value>,

"cipheader":<String_value>,
```

```
"maxclient":<Double_value>,

"clttimeout":<Double_value>,

"port":<Integer_value>,

"dup_svctype":<String_value>,

"monthreshold":<Double_value>,

"maxreq":<Double_value>,

"serviceipaddress":<String_value>,

"maxbandwidth":<Double_value>,

"svrcfgflags":<Double_value>

}]}
```

count

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/server_gslbservicegroup_binding/name_value<String>?**count=yes**

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{"server_gslbservicegroup_binding": [ { "__count": "#no"} ] }
```

server_servicegroup_binding

Binding object showing the servicegroup that can be bound to server.

Properties

(click to see Operations)

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
name	<String>	Read-write	Name of the server for which to display parameters. Minimum length = 1
servicegroupname	<String>	Read-write	servicegroups bind to this server.
sp	<String>	Read-only	Enable surge protection for the service group. Default value: OFF Possible values = ON, OFF
sc	<String>	Read-only	State of the SureConnect feature for the service group. Default value: OFF Possible values = ON, OFF
servicegroupentname2	<String>	Read-only	Used for show server of SRV type to

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
			indicate target FQDNs.
svcitmpriority	<Double>	Read-only	Indicates the Priority of the SRV target FQDN.
svrtimeout	<Double>	Read-only	Time, in seconds, after which to terminate an idle server connection. Minimum value = 0 Maximum value = 31536000
serviceipstr	<String>	Read-only	This field has been intorduced to show the dbs services ip.
tcpb	<String>	Read-only	Enable TCP buffering for the service group. Possible values = YES, NO
cip	<String>	Read-only	Before forwarding a request to the service, insert an HTTP header with the client's IPv4 or IPv6 address as its value. Used if the server needs the client's IP address for security, accounting, or other purposes, and setting the Use Source IP parameter is not a viable option.

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
			Possible values = ENABLED, DISABLED
dup_port	<Integer>	Read-only	port of the service. Range 1 - 65535 * in CLI is represented as 65535 in NITRO API
svcitmboundsvcs	<Double>	Read-only	Indicates the total number of bound IPs for the SRV target FQDN.
customserverid	<String>	Read-only	A positive integer to identify the service. Used when the persistency type is set to Custom Server ID. Default value: "None"
weight	<Double>	Read-only	Indicates the weight of bound IPs.
boundtd	<Double>	Read-only	Integer value that uniquely identifies the traffic domain in which you want to configure the entity. If you do not specify an ID, the entity becomes part of the default traffic domain, which has an ID of 0.

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
			Minimum value = 0 Maximum value = 4094
downstateflush	<String>	Read-only	Perform delayed clean-up of connections to all services in the service group. Default value: ENABLED Possible values = ENABLED, DISABLED
svctype	<String>	Read-only	The type of bound service. Possible values = HTTP, FTP, TCP, UDP, SSL, SSL_BRIDGE, SSL_TCP, DTLS, NNTP, RPCSVR, DNS, ADNS, SNMP, RTSP, DHCPRA, ANY, SIP_UDP, SIP_TCP, SIP_SSL, DNS_TCP, ADNS_TCP, MYSQL, MSSQL, ORACLE, RADIUS, RADIUSListener, RDP, DIAMETER, SSL_DIAMETER, TFTP, SMPP, PPTP, GRE, SYSLOGTCP,

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
			SYSLOGUDP, FIX, SSL_FIX, USER_TCP, USER_SSL_TCP, QUIC, IPFIX, LOGSTREAM
cka	<String>	Read-only	Enable client keep-alive for the service group. Possible values = YES, NO
svrstate	<String>	Read-only	The state of the bound service. Possible values = UP, DOWN, UNKNOWN, BUSY, OUT OF SERVICE, GOING OUT OF SERVICE, DOWN WHEN GOING OUT OF SERVICE, NS_EMPTY_STR, Unknown, DISABLED
appflowlog	<String>	Read-only	Enable logging of AppFlow information for the specified service group. Default value: ENABLED Possible values = ENABLED, DISABLED

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
cipheader	<String>	Read-only	Name of the HTTP header whose value must be set to the IP address of the client. Used with the Client IP parameter. If client IP insertion is enabled, and the client IP header is not specified, the value of Client IP Header parameter or the value set by the set ns config command is used as client's IP header name. Minimum length = 1
maxclient	<Double>	Read-only	Maximum number of simultaneous open connections for the service group. Minimum value = 0 Maximum value = 4294967294
clttimeout	<Double>	Read-only	Time, in seconds, after which to terminate an idle client connection. Minimum value = 0 Maximum value = 31536000
port	<Integer>	Read-only	The port number to be used for the bound service.

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
			Range 1 - 65535 * in CLI is represented as 65535 in NITRO API
dup_svctype	<String>	Read-only	service type of the service. Possible values = HTTP, FTP, TCP, UDP, SSL, SSL_BRIDGE, SSL_TCP, DTLS, NNTP, RPCSVR, DNS, ADNS, SNMP, RTSP, DHCPRA, ANY, SIP_UDP, SIP_TCP, SIP_SSL, DNS_TCP, ADNS_TCP, MYSQL, MSSQL, ORACLE, RADIUS, RADIUSListener, RDP, DIAMETER, SSL_DIAMETER, TFTP, SMPP, PPTP, GRE, SYSLOGTCP, SYSLOGUDP, FIX, SSL_FIX, USER_TCP, USER_SSL_TCP, QUIC, IPFIX, LOGSTREAM

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
cmp	<String>	Read-only	Enable compression for the specified service. Possible values = YES, NO
monthreshold	<Double>	Read-only	Minimum sum of weights of the monitors that are bound to this service. Used to determine whether to mark a service as UP or DOWN. Minimum value = 0 Maximum value = 65535
maxreq	<Double>	Read-only	Maximum number of requests that can be sent on a persistent connection to the service group. Note: Connection requests beyond this value are rejected. Minimum value = 0 Maximum value = 65535
serviceipaddress	<String>	Read-only	The IP address of the bound service.
svcitmactsvcs	<Double>	Read-only	Indicates the total number of active IPs for SRV target FQDN.

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
usip	<String>	Read-only	Use the client's IP address as the source IP address when initiating a connection to the server. When creating a service, if you do not set this parameter, the service inherits the global Use Source IP setting (available in the enable ns mode and disable ns mode CLI commands, or in the System > Settings > Configure modes > Configure Modes dialog box). However, you can override this setting after you create the service. Possible values = YES, NO
cacheable	<String>	Read-only	Use the transparent cache redirection virtual server to forward the request to the cache server. Default value: NO Possible values = YES, NO
maxbandwidth	<Double>	Read-only	Maximum bandwidth, in Kbps, allocated for all the services in the service group.

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
			Minimum value = 0 Maximum value = 4294967287
svrcfgflags	<Double>	Read-only	service flags to denote its a db enabled.
__count	<Double>	Read-only	count parameter

Operations

(click to see Properties)

GET| GET (ALL)| COUNT

Some options that you can use for each operations:

- **Getting warnings in response:**NITRO allows you to get warnings in an operation by specifying the "warning" query parameter as "yes". For example, to get warnings while connecting to the Citrix ADC appliance, the URL is as follows:

  http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/login?warning=yes

  If any, the warnings are displayed in the response payload with the HTTP code "209 X-NITRO-WARNING".

- **Authenticated access for individual NITRO operations:**NITRO allows you to logon to the Citrix ADC appliance to perform individual operations. You can use this option instead of creating a NITRO session (using the login object) and then using that session to perform all operations,

  To do this, you must specify the username and password in the request header of the NITRO request as follows:

  X-NITRO-USER:<username>

  X-NITRO-PASS:<password>

  Note: In such cases, make sure that the request header DOES not include the following:

  Cookie:NITRO_AUTH_TOKEN=<tokenvalue>

`*Note:`*

Mandatory parameters are marked in redand placeholder content is marked in <green>.

get

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/server_servicegroup_binding/name_value<String>

Query-parameters:

filter

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/server_servicegroup_binding/name_value<String>?**filter=property-name1:property-value1,property-name2:property-value2**

Use this query-parameter to get the filtered set of server_servicegroup_binding resources configured on Citrix ADC. Filtering can be done on any of the properties of the resource.

pagination

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/server_servicegroup_binding/name_value<String>?**pagesize=#no;pageno=#no**

Use this query-parameter to get the server_servicegroup_binding resources in chunks.

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "server_servicegroup_binding": [ {
```

```
"name":<String_value>,

"servicegroupname":<String_value>,

"sp":<String_value>,

"sc":<String_value>,

"servicegroupentname2":<String_value>,

"svcitmpriority":<Double_value>,

"svrtimeout":<Double_value>,

"serviceipstr":<String_value>,

"tcpb":<String_value>,

"cip":<String_value>,

"dup_port":<Integer_value>,

"svcitmboundsvcs":<Double_value>,

"customserverid":<String_value>,

"weight":<Double_value>,

"boundtd":<Double_value>,

"downstateflush":<String_value>,

"svctype":<String_value>,

"cka":<String_value>,

"svrstate":<String_value>,

"appflowlog":<String_value>,

"cipheader":<String_value>,

"maxclient":<Double_value>,
```

```
"clttimeout":<Double_value>,

"port":<Integer_value>,

"dup_svctype":<String_value>,

"cmp":<String_value>,

"monthreshold":<Double_value>,

"maxreq":<Double_value>,

"serviceipaddress":<String_value>,

"svcitmactsvcs":<Double_value>,

"usip":<String_value>,

"cacheable":<String_value>,

"maxbandwidth":<Double_value>,

"svrcfgflags":<Double_value>

}]}
```

get (all)

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/server_servicegroup_binding

Query-parameters:

bulkbindings

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/server_servicegroup_binding?**bulkbindings=yes**

NITRO allows you to fetch bindings in bulk.

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "server_servicegroup_binding": [ {

"name":<String_value>,

"servicegroupname":<String_value>,

"sp":<String_value>,

"sc":<String_value>,

"servicegroupentname2":<String_value>,

"svcitmpriority":<Double_value>,

"svrtimeout":<Double_value>,

"serviceipstr":<String_value>,

"tcpb":<String_value>,

"cip":<String_value>,

"dup_port":<Integer_value>,

"svcitmboundsvcs":<Double_value>,

"customserverid":<String_value>,

"weight":<Double_value>,
```

```
"boundtd":<Double_value>,

"downstateflush":<String_value>,

"svctype":<String_value>,

"cka":<String_value>,

"svrstate":<String_value>,

"appflowlog":<String_value>,

"cipheader":<String_value>,

"maxclient":<Double_value>,

"clttimeout":<Double_value>,

"port":<Integer_value>,

"dup_svctype":<String_value>,

"cmp":<String_value>,

"monthreshold":<Double_value>,

"maxreq":<Double_value>,

"serviceipaddress":<String_value>,

"svcitmactsvcs":<Double_value>,

"usip":<String_value>,

"cacheable":<String_value>,

"maxbandwidth":<Double_value>,

"svrcfgflags":<Double_value>

}]}
```

count

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/server_servicegroup_binding/name_value<String>?**count=yes**

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{"server_servicegroup_binding": [ { "__count": "#no"} ] }
```

server_service_binding

Binding object showing the service that can be bound to server.

Properties

(click to see Operations)

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
name	<String>	Read-write	Name of the server for which to display parameters. Minimum length = 1

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
servicename	<String>	Read-write	The services attatched to the server.
serviceipstr	<String>	Read-only	This field has been intorduced to show the dbs services ip.
svctype	<String>	Read-only	The type of bound service. Possible values = HTTP, FTP, TCP, UDP, SSL, SSL_BRIDGE, SSL_TCP, DTLS, NNTP, RPCSVR, DNS, ADNS, SNMP, RTSP, DHCPRA, ANY, SIP_UDP, SIP_TCP, SIP_SSL, DNS_TCP, ADNS_TCP, MYSQL, MSSQL, ORACLE, RADIUS, RADIUSListener, RDP, DIAMETER, SSL_DIAMETER, TFTP, SMPP, PPTP, GRE, SYSLOGTCP, SYSLOGUDP, FIX, SSL_FIX, USER_TCP, USER_SSL_TCP, QUIC, IPFIX, LOGSTREAM

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
svrstate	<String>	Read-only	The state of the bound service. Possible values = UP, DOWN, UNKNOWN, BUSY, OUT OF SERVICE, GOING OUT OF SERVICE, DOWN WHEN GOING OUT OF SERVICE, NS_EMPTY_STR, Unknown, DISABLED
port	<Integer>	Read-only	The port number to be used for the bound service. Range 1 - 65535 * in CLI is represented as 65535 in NITRO API
serviceipaddress	<String>	Read-only	The IP address of the bound service.
__count	<Double>	Read-only	count parameter

Operations

(click to see Properties)

GET| GET (ALL)| COUNT

Some options that you can use for each operations:

- **Getting warnings in response:**NITRO allows you to get warnings in an operation by specifying the "warning" query parameter as "yes". For example, to get warnings while connecting to the Citrix ADC appliance, the URL is as follows:

  http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/login?warning=yes

  If any, the warnings are displayed in the response payload with the HTTP code "209 X-NITRO-WARNING".

- **Authenticated access for individual NITRO operations:**NITRO allows you to logon to the Citrix ADC appliance to perform individual operations. You can use this option instead of creating a NITRO session (using the login object) and then using that session to perform all operations,

  To do this, you must specify the username and password in the request header of the NITRO request as follows:

  X-NITRO-USER:<username>

  X-NITRO-PASS:<password>

  Note: In such cases, make sure that the request header DOES not include the following:

  Cookie:NITRO_AUTH_TOKEN=<tokenvalue>

`*Note:`*

Mandatory parameters are marked in redand placeholder content is marked in <green>.

get

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/server_service_binding/name_value<String>

Query-parameters:

filter

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/server_service_binding/name_value<String>?**filter=property-name1:property-value1,property-name2:property-value2**

Use this query-parameter to get the filtered set of server_service_binding resources configured on Citrix ADC. Filtering can be done on any of the properties of the resource.

pagination

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/server_service_binding/name_value<String>?**pagesize=#no;pageno=#no**

Use this query-parameter to get the server_service_binding resources in chunks.

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "server_service_binding": [ {

"name":<String_value>,

"servicename":<String_value>,

"serviceipstr":<String_value>,

"svctype":<String_value>,

"svrstate":<String_value>,

"port":<Integer_value>,

"serviceipaddress":<String_value>

}]}
```

get (all)

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/server_service_binding

Query-parameters:

bulkbindings

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/server_service_binding?**bulkbindings=yes**

NITRO allows you to fetch bindings in bulk.

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "server_service_binding": [ {

"name":<String_value>,

"servicename":<String_value>,

"serviceipstr":<String_value>,

"svctype":<String_value>,

"svrstate":<String_value>,

"port":<Integer_value>,

"serviceipaddress":<String_value>
```

```
}]}
```

count

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/server_service_binding/name_value<String>?**count=yes**

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{"server_service_binding": [ { "__count": "#no"} ] }
```

service

Configuration for service resource.

Properties

(click to see Operations)

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
name	<String>	Read-write	Name for the service. Must begin with an ASCII alphabetic or

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
			underscore (_) character, and must contain only ASCII alphanumeric, underscore, hash (#), period (.), space, colon (:), at (@), equals (=), and hyphen (-) characters. Cannot be changed after the service has been created. Minimum length = 1
ip	<String>	Read-write	IP to assign to the service. Minimum length = 1
servername	<String>	Read-write	Name of the server that hosts the service. Minimum length = 1
servicetype	<String>	Read-write	Protocol in which data is exchanged with the service. Possible values = HTTP, FTP, TCP, UDP, SSL, SSL_BRIDGE, SSL_TCP, DTLS, NNTP, RPCSVR, DNS, ADNS, SNMP, RTSP, DHCPRA, ANY, SIP_UDP, SIP_TCP, SIP_SSL, DNS_TCP, ADNS_TCP,

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
			MYSQL, MSSQL, ORACLE, RADIUS, RADIUSListener, RDP, DIAMETER, SSL_DIAMETER, TFTP, SMPP, PPTP, GRE, SYSLOGTCP, SYSLOGUDP, FIX, SSL_FIX, USER_TCP, USER_SSL_TCP, QUIC, IPFIX, LOGSTREAM
port	<Integer>	Read-write	Port number of the service. Range 1 - 65535 * in CLI is represented as 65535 in NITRO API
cleartextport	<Integer>	Read-write	Port to which clear text data must be sent after the appliance decrypts incoming SSL traffic. Applicable to transparent SSL services. Minimum value = 1
cachetype	<String>	Read-write	Cache type supported by the cache server. Possible values = TRANSPARENT,

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
			REVERSE, FORWARD
maxclient	<Double>	Read-write	Maximum number of simultaneous open connections to the service. Minimum value = 0 Maximum value = 4294967294
healthmonitor	<String>	Read-write	Monitor the health of this service. Available settings function as follows: YES - Send probes to check the health of the service. NO - Do not send probes to check the health of the service. With the NO option, the appliance shows the service as UP at all times. Default value: YES Possible values = YES, NO
maxreq	<Double>	Read-write	Maximum number of requests that can be sent on a persistent connection to the service.

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
			Note: Connection requests beyond this value are rejected. Minimum value = 0 Maximum value = 65535
cacheable	<String>	Read-write	Use the transparent cache redirection virtual server to forward requests to the cache server. Note: Do not specify this parameter if you set the Cache Type parameter. Default value: NO Possible values = YES, NO
cip	<String>	Read-write	Before forwarding a request to the service, insert an HTTP header with the client's IPv4 or IPv6 address as its value. Used if the server needs the client's IP address for security, accounting, or other purposes, and setting the Use Source IP parameter is not a viable option.

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
			Possible values = ENABLED, DISABLED
cipheader	<String>	Read-write	Name for the HTTP header whose value must be set to the IP address of the client. Used with the Client IP parameter. If you set the Client IP parameter, and you do not specify a name for the header, the appliance uses the header name specified for the global Client IP Header parameter (the cipHeader parameter in the set ns param CLI command or the Client IP Header parameter in the Configure HTTP Parameters dialog box at System > Settings > Change HTTP parameters). If the global Client IP Header parameter is not specified, the appliance inserts a header with the name "client-ip.". Minimum length = 1
usip	<String>	Read-write	Use the client's IP address as the source IP address

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
			when initiating a connection to the server. When creating a service, if you do not set this parameter, the service inherits the global Use Source IP setting (available in the enable ns mode and disable ns mode CLI commands, or in the System > Settings > Configure modes > Configure Modes dialog box). However, you can override this setting after you create the service. Possible values = YES, NO
pathmonitor	<String>	Read-write	Path monitoring for clustering. Possible values = YES, NO
pathmonitorindv	<String>	Read-write	Individual Path monitoring decisions. Possible values = YES, NO
useproxyport	<String>	Read-write	Use the proxy port as the source port when initiating connections with the server. With the NO setting, the client-side connection port

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
			is used as the source port for the server-side connection. Note: This parameter is available only when the Use Source IP (USIP) parameter is set to YES. Possible values = YES, NO
sc	<String>	Read-write	State of SureConnect for the service. Default value: OFF Possible values = ON, OFF
sp	<String>	Read-write	Enable surge protection for the service. Possible values = ON, OFF
rtspsessionidremap	<String>	Read-write	Enable RTSP session ID mapping for the service. Default value: OFF Possible values = ON, OFF
clttimeout	<Double>	Read-write	Time, in seconds, after which to terminate an idle client connection.

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
			Minimum value = 0 Maximum value = 31536000
svrtimeout	<Double>	Read-write	Time, in seconds, after which to terminate an idle server connection. Minimum value = 0 Maximum value = 31536000
customserverid	<String>	Read-write	Unique identifier for the service. Used when the persistency type for the virtual server is set to Custom Server ID. Default value: "None"
serverid	<Double>	Read-write	The identifier for the service. This is used when the persistency type is set to Custom Server ID.
cka	<String>	Read-write	Enable client keep-alive for the service. Possible values = YES, NO
tcpb	<String>	Read-write	Enable TCP buffering for the service. Possible values = YES, NO

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
cmp	<String>	Read-write	Enable compression for the service. Possible values = YES, NO
maxbandwidth	<Double>	Read-write	Maximum bandwidth, in Kbps, allocated to the service. Minimum value = 0 Maximum value = 4294967287
accessdown	<String>	Read-write	Use Layer 2 mode to bridge the packets sent to this service if it is marked as DOWN. If the service is DOWN, and this parameter is disabled, the packets are dropped. Default value: NO Possible values = YES, NO
monthreshold	<Double>	Read-write	Minimum sum of weights of the monitors that are bound to this service. Used to determine whether to mark a service as UP or DOWN. Minimum value = 0 Maximum value = 65535

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
state	<String>	Read-write	Initial state of the service. Default value: ENABLED Possible values = ENABLED, DISABLED
downstateflush	<String>	Read-write	Flush all active transactions associated with a service whose state transitions from UP to DOWN. Do not enable this option for applications that must complete their transactions. Default value: ENABLED Possible values = ENABLED, DISABLED
tcpprofilename	<String>	Read-write	Name of the TCP profile that contains TCP configuration settings for the service. Minimum length = 1 Maximum length = 127
httpprofilename	<String>	Read-write	Name of the HTTP profile that contains HTTP configuration

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
			settings for the service. Minimum length = 1 Maximum length = 127
contentinspectionprofilename	<String>	Read-write	Name of the ContentInspection profile that contains IPS/IDS communication related setting for the service. Minimum length = 1 Maximum length = 127
hashid	<Double>	Read-write	A numerical identifier that can be used by hash based load balancing methods. Must be unique for each service. Minimum value = 1
comment	<String>	Read-write	Any information about the service.
appflowlog	<String>	Read-write	Enable logging of AppFlow information. Default value: ENABLED Possible values = ENABLED, DISABLED

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
netprofile	<String>	Read-write	Network profile to use for the service. Minimum length = 1 Maximum length = 127
td	<Double>	Read-write	Integer value that uniquely identifies the traffic domain in which you want to configure the entity. If you do not specify an ID, the entity becomes part of the default traffic domain, which has an ID of 0. Minimum value = 0 Maximum value = 4094
processlocal	<String>	Read-write	By turning on this option packets destined to a service in a cluster will not under go any steering. Turn this option for single packet request response mode or when the upstream device is performing a proper RSS for connection based distribution. Default value: DISABLED

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
			Possible values = ENABLED, DISABLED
dnsprofilename	<String>	Read-write	Name of the DNS profile to be associated with the service. DNS profile properties will applied to the transactions processed by a service. This parameter is valid only for ADNS and ADNS-TCP services. Minimum length = 1 Maximum length = 127
monconnectionclose	<String>	Read-write	Close monitoring connections by sending the service a connection termination message with the specified bit set. Default value: NONE Possible values = RESET, FIN
ipaddress	<String>	Read-write	The new IP address of the service.
weight	<Double>	Read-write	Weight to assign to the monitor-service binding. When a monitor is UP, the

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
			weight assigned to its binding with the service determines how much the monitor contributes toward keeping the health of the service above the value configured for the Monitor Threshold parameter. Minimum value = 1 Maximum value = 100
monitor_name_svc	<String>	Read-write	Name of the monitor bound to the specified service. Minimum length = 1
riseapbrstatsmsgcode	<Integer>	Read-write	The code indicating the rise apbr status.
delay	<Double>	Read-write	Time, in seconds, allocated to the Citrix ADC for a graceful shutdown of the service. During this period, new requests are sent to the service only for clients who already have persistent sessions on the appliance. Requests from new clients are load balanced among other available services. After the delay time

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
			expires, no requests are sent to the service, and the service is marked as unavailable (OUT OF SERVICE).
graceful	<String>	Read-write	Shut down gracefully, not accepting any new connections, and disabling the service when all of its connections are closed. Default value: NO Possible values = YES, NO
all	<Boolean>	Read-write	Display both user-configured and dynamically learned services.
Internal	<Boolean>	Read-write	Display only dynamically learned services.
newname	<String>	Read-write	New name for the service. Must begin with an ASCII alphabetic or underscore (_) character, and must contain only ASCII alphanumeric, underscore, hash (#), period (.), space, colon (:), at (@), equals (=), and

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
			hyphen (-) characters. Minimum length = 1
numofconnections	<Integer>	Read-only	This will tell the number of client side connections are still open.
policyname	<String>	Read-only	The name of the policyname for which this service is bound.
serviceconftype	<Boolean>	Read-only	The configuration type of the service.
serviceconftype2	<String>	Read-only	The configuration type of the service (Internal/Dynamic/ Configured). Possible values = Configured, Dynamic, Internal
value	<String>	Read-only	SSL status. Possible values = Certkey not bound, SSL feature disabled
gslb	<String>	Read-only	The GSLB option for the corresponding virtual server. Possible values = REMOTE, LOCAL
dup_state	<String>	Read-only	Added this field for getting state value from table.

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
			Possible values = ENABLED, DISABLED
publicip	<String>	Read-only	public ip.
publicport	<Integer>	Read-only	public port. Minimum value = 1 Range 1 - 65535 * in CLI is represented as 65535 in NITRO API
svrstate	<String>	Read-only	The state of the service. Possible values = UP, DOWN, UNKNOWN, BUSY, OUT OF SERVICE, GOING OUT OF SERVICE, DOWN WHEN GOING OUT OF SERVICE, NS_EMPTY_STR, Unknown, DISABLED
monitor_state	<String>	Read-only	The running state of the monitor on this service. Possible values = UP, DOWN, UNKNOWN, BUSY, OUT OF SERVICE, GOING OUT OF

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
			SERVICE, DOWN WHEN GOING OUT OF SERVICE, NS_EMPTY_STR, Unknown, DISABLED
monstatcode	<Integer>	Read-only	The code indicating the monitor response.
lastresponse	<String>	Read-only	The string form of monstatcode.
responsetime	<Double>	Read-only	Response time of this monitor.
riseapbrstatsmsgcode2	<Integer>	Read-only	The code indicating other rise stats.
monstatparam1	<Integer>	Read-only	First parameter for use with message code.
monstatparam2	<Integer>	Read-only	Second parameter for use with message code.
monstatparam3	<Integer>	Read-only	Third parameter for use with message code.
statechangetimesec	<String>	Read-only	Time when last state change happened. Seconds part.
statechangetimemsec	<Double>	Read-only	Time at which last state change happened. Milliseconds part.

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
tickssincelaststatechange	<Double>	Read-only	Time in 10 millisecond ticks since the last state change.
stateupdatereason	<Double>	Read-only	Checks state update reason on the secondary node.
clmonowner	<Double>	Read-only	Tells the mon owner of the service. Minimum value = 0 Maximum value = 32
clmonview	<Double>	Read-only	Tells the view id of the monitoring owner.
serviceipstr	<String>	Read-only	This field has been intorduced to show the dbs services ip.
oracleserverversion	<String>	Read-only	Oracle server version. Default value: 10G Possible values = 10G, 11G
nodefaultbindings	<String>	Read-only	to determine if the configuration will have default ssl CIPHER and ECC curve bindings. Default value: NO Possible values = YES, NO

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
__count	<Double>	Read-only	count parameter

Operations

(click to see Properties)

ADD| DELETE| UPDATE| UNSET| ENABLE| DISABLE| RENAME| GET (ALL)| GET| COUNT

Some options that you can use for each operations:

- **Getting warnings in response:**NITRO allows you to get warnings in an operation by specifying the "warning" query parameter as "yes". For example, to get warnings while connecting to the Citrix ADC appliance, the URL is as follows:

  http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/login?warning=yes

  If any, the warnings are displayed in the response payload with the HTTP code "209 X-NITRO-WARNING".
- **Authenticated access for individual NITRO operations:**NITRO allows you to logon to the Citrix ADC appliance to perform individual operations. You can use this option instead of creating a NITRO session (using the login object) and then using that session to perform all operations,

  To do this, you must specify the username and password in the request header of the NITRO request as follows:

  X-NITRO-USER:<username>

  X-NITRO-PASS:<password>

  Note: In such cases, make sure that the request header DOES not include the following:

  Cookie:NITRO_AUTH_TOKEN=<tokenvalue>

`*Note:`*

Mandatory parameters are marked in redand placeholder content is marked in <green>.

add

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/service

HTTP Method:POST

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Content-Type:application/json

Request Payload:

```
{"service":{

<b>"name":<String_value>,

</b>"ip":<String_value>,

"servername":<String_value>,

<b>"servicetype":<String_value>,

</b><b>"port":<Integer_value>,

</b>"cleartextport":<Integer_value>,

"cachetype":<String_value>,

"maxclient":<Double_value>,

"healthmonitor":<String_value>,

"maxreq":<Double_value>,

"cacheable":<String_value>,

"cip":<String_value>,

"cipheader":<String_value>,

"usip":<String_value>,

"pathmonitor":<String_value>,

"pathmonitorindv":<String_value>,

"useproxyport":<String_value>,
```

```
"sc":<String_value>,

"sp":<String_value>,

"rtspsessionidremap":<String_value>,

"clttimeout":<Double_value>,

"svrtimeout":<Double_value>,

"customserverid":<String_value>,

"serverid":<Double_value>,

"cka":<String_value>,

"tcpb":<String_value>,

"cmp":<String_value>,

"maxbandwidth":<Double_value>,

"accessdown":<String_value>,

"monthreshold":<Double_value>,

"state":<String_value>,

"downstateflush":<String_value>,

"tcpprofilename":<String_value>,

"httpprofilename":<String_value>,

"contentinspectionprofilename":<String_value>,

"hashid":<Double_value>,

"comment":<String_value>,

"appflowlog":<String_value>,

"netprofile":<String_value>,
```

```
"td":<Double_value>,

"processlocal":<String_value>,

"dnsprofilename":<String_value>,

"monconnectionclose":<String_value>

}}
```

Response:

HTTP Status Code on Success: 201 Created HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error

delete

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/service/name_value<String>

HTTP Method:DELETE

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue>

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error

update

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/service

HTTP Method:PUT

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Content-Type:application/json

Request Payload:

```
{"service":{

<b>"name":<String_value>,

</b>"ipaddress":<String_value>,

"maxclient":<Double_value>,

"maxreq":<Double_value>,

"cacheable":<String_value>,

"cip":<String_value>,

"cipheader":<String_value>,

"usip":<String_value>,

"pathmonitor":<String_value>,

"pathmonitorindv":<String_value>,

"useproxyport":<String_value>,

"sc":<String_value>,

"sp":<String_value>,

"rtspsessionidremap":<String_value>,

"healthmonitor":<String_value>,

"clttimeout":<Double_value>,

"svrtimeout":<Double_value>,

"customserverid":<String_value>,

"serverid":<Double_value>,

"cka":<String_value>,
```

```
"tcpb":<String_value>,

"cmp":<String_value>,

"maxbandwidth":<Double_value>,

"accessdown":<String_value>,

"monthreshold":<Double_value>,

"weight":<Double_value>,

"monitor_name_svc":<String_value>,

"downstateflush":<String_value>,

"tcpprofilename":<String_value>,

"httpprofilename":<String_value>,

"contentinspectionprofilename":<String_value>,

"hashid":<Double_value>,

"comment":<String_value>,

"appflowlog":<String_value>,

"netprofile":<String_value>,

"processlocal":<String_value>,

"dnsprofilename":<String_value>,

"monconnectionclose":<String_value>

}}
```

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error

unset

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/service?**action=unset**

HTTP Method:POST

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Content-Type:application/json

Request Payload:

```
{"service":{

<b>"name":<String_value>,

</b>"maxclient":true,

"maxreq":true,

"cacheable":true,

"cip":true,

"usip":true,

"pathmonitor":true,

"pathmonitorindv":true,

"useproxyport":true,

"sc":true,

"sp":true,

"rtspsessionidremap":true,
```

```
"customserverid":true,

"serverid":true,

"cka":true,

"tcpb":true,

"cmp":true,

"maxbandwidth":true,

"accessdown":true,

"monthreshold":true,

"clttimeout":true,

"riseapbrstatsmsgcode":true,

"svrtimeout":true,

"tcpprofilename":true,

"httpprofilename":true,

"contentinspectionprofilename":true,

"hashid":true,

"appflowlog":true,

"netprofile":true,

"processlocal":true,

"dnsprofilename":true,

"monconnectionclose":true,

"cipheader":true,

"healthmonitor":true,
```

```
"downstateflush":true,

"comment":true

}}
```

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error

enable

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/service?**action=enable**

HTTP Method:POST

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Content-Type:application/json

Request Payload:

```
{"service":{

<b>"name":<String_value>

</b>}}
```

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error

disable

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/service?**action=disable**

HTTP Method:POST

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Content-Type:application/json

Request Payload:

```
{"service":{

<b>"name":<String_value>,

</b>"delay":<Double_value>,

"graceful":<String_value>

}}
```

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error

rename

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/service?**action=rename**

HTTP Method:POST

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Content-Type:application/json

Request Payload:

```
{"service":{

<b>"name":<String_value>,
```

```
</b><b>"newname":<String_value>

</b>}}
```

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error

get (all)

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/service

Query-parameters:

args

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/service?**args=name:<String_value>,all:<Boolean_value>,Internal:<Boolean_value>**

Use this query-parameter to get service resources based on additional properties.

attrs

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/service?**attrs=property-name1,property-name2**

Use this query parameter to specify the resource details that you want to retrieve.

filter

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/service?**filter=property-name1:property-val1,property-name2:property-val2**

Use this query-parameter to get the filtered set of service resources configured on Citrix ADC. Filtering can be done on any of the properties of the resource.

view

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/service?**view=summary**

Note: By default, the retrieved results are displayed in detail view (?view=detail).

pagination

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/service?**pagesize=#no;pageno=#no**

Use this query-parameter to get the service resources in chunks.

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "service": [ {

name:<String_value>,all:<Boolean_value>,Internal:<Boolean_value>"numofconn
ections":<Integer_value>,

"servername":<String_value>,

"policyname":<String_value>,

"servicetype":<String_value>,

"serviceconftpye":<Boolean_value>,

"serviceconftype":<Boolean_value>,

"serviceconftype2":<String_value>,

"port":<Integer_value>,

"value":<String_value>,
```

```
"cleartextport":<Integer_value>,

"gslb":<String_value>,

"cachetype":<String_value>,

"maxclient":<Double_value>,

"maxreq":<Double_value>,

"cacheable":<String_value>,

"cip":<String_value>,

"cipheader":<String_value>,

"usip":<String_value>,

"pathmonitor":<String_value>,

"pathmonitorindv":<String_value>,

"useproxyport":<String_value>,

"sc":<String_value>,

"dup_state":<String_value>,

"sp":<String_value>,

"rtspsessionidremap":<String_value>,

"clttimeout":<Double_value>,

"svrtimeout":<Double_value>,

"publicip":<String_value>,

"publicport":<Integer_value>,

"customserverid":<String_value>,

"serverid":<Double_value>,
```

```
"cka":<String_value>,

"tcpb":<String_value>,

"processlocal":<String_value>,

"cmp":<String_value>,

"maxbandwidth":<Double_value>,

"accessdown":<String_value>,

"svrstate":<String_value>,

"delay":<Double_value>,

"ipaddress":<String_value>,

"monthreshold":<Double_value>,

"monitor_state":<String_value>,

"monstatcode":<Integer_value>,

"lastresponse":<String_value>,

"responsetime":<Double_value>,

"riseapbrstatsmsgcode":<Integer_value>,

"riseapbrstatsmsgcode2":<Integer_value>,

"monstatparam1":<Integer_value>,

"monstatparam2":<Integer_value>,

"monstatparam3":<Integer_value>,

"downstateflush":<String_value>,

"statechangetimesec":<String_value>,

"statechangetimemsec":<Double_value>,
```

```
"timesincelaststatechange":<Double_value>,

"tickssincelaststatechange":<Double_value>,

"stateupdatereason":<Double_value>,

"clmonowner":<Double_value>,

"clmonview":<Double_value>,

"tcpprofilename":<String_value>,

"httpprofilename":<String_value>,

"contentinspectionprofilename":<String_value>,

"hashid":<Double_value>,

"graceful":<String_value>,

"comment":<String_value>,

"healthmonitor":<String_value>,

"appflowlog":<String_value>,

"netprofile":<String_value>,

"serviceipstr":<String_value>,

"weight":<Double_value>,

"td":<Double_value>,

"oracleserverversion":<String_value>,

"dnsprofilename":<String_value>,

"monconnectionclose":<String_value>,

"nodefaultbindings":<String_value>

}]}
```

get

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/service/name_value<String>

Query-parameters:

attrs

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/service/name_value<String>?**attrs=property-name1,property-name2**

Use this query parameter to specify the resource details that you want to retrieve.

view

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/service/name_value<String>?**view=summary**

Note: By default, the retrieved results are displayed in detail view (?view=detail).

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "service": [ {

name:<String_value>,all:<Boolean_value>,Internal:<Boolean_value>"numofconn
ections":<Integer_value>,

"servername":<String_value>,
```

```
"policyname":<String_value>,

"servicetype":<String_value>,

"serviceconftpye":<Boolean_value>,

"serviceconftype":<Boolean_value>,

"serviceconftype2":<String_value>,

"port":<Integer_value>,

"value":<String_value>,

"cleartextport":<Integer_value>,

"gslb":<String_value>,

"cachetype":<String_value>,

"maxclient":<Double_value>,

"maxreq":<Double_value>,

"cacheable":<String_value>,

"cip":<String_value>,

"cipheader":<String_value>,

"usip":<String_value>,

"pathmonitor":<String_value>,

"pathmonitorindv":<String_value>,

"useproxyport":<String_value>,

"sc":<String_value>,

"dup_state":<String_value>,

"sp":<String_value>,
```

```
"rtspsessionidremap":<String_value>,

"clttimeout":<Double_value>,

"svrtimeout":<Double_value>,

"publicip":<String_value>,

"publicport":<Integer_value>,

"customserverid":<String_value>,

"serverid":<Double_value>,

"cka":<String_value>,

"tcpb":<String_value>,

"processlocal":<String_value>,

"cmp":<String_value>,

"maxbandwidth":<Double_value>,

"accessdown":<String_value>,

"svrstate":<String_value>,

"delay":<Double_value>,

"ipaddress":<String_value>,

"monthreshold":<Double_value>,

"monitor_state":<String_value>,

"monstatcode":<Integer_value>,

"lastresponse":<String_value>,

"responsetime":<Double_value>,

"riseapbrstatsmsgcode":<Integer_value>,
```

```
"riseapbrstatsmsgcode2":<Integer_value>,

"monstatparam1":<Integer_value>,

"monstatparam2":<Integer_value>,

"monstatparam3":<Integer_value>,

"downstateflush":<String_value>,

"statechangetimesec":<String_value>,

"statechangetimemsec":<Double_value>,

"timesincelaststatechange":<Double_value>,

"tickssincelaststatechange":<Double_value>,

"stateupdatereason":<Double_value>,

"clmonowner":<Double_value>,

"clmonview":<Double_value>,

"tcpprofilename":<String_value>,

"httpprofilename":<String_value>,

"contentinspectionprofilename":<String_value>,

"hashid":<Double_value>,

"graceful":<String_value>,

"comment":<String_value>,

"healthmonitor":<String_value>,

"appflowlog":<String_value>,

"netprofile":<String_value>,

"serviceipstr":<String_value>,
```

```
"weight":<Double_value>,

"td":<Double_value>,

"oracleserverversion":<String_value>,

"dnsprofilename":<String_value>,

"monconnectionclose":<String_value>,

"nodefaultbindings":<String_value>

}]}
```

count

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/service?**count=yes**

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "service": [ { "__count": "#no"} ] }
```

servicegroup

Configuration for service group resource.

Properties

(click to see Operations)

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
servicegroupname	<String>	Read-write	Name of the service group. Must begin with an ASCII alphabetic or underscore (_) character, and must contain only ASCII alphanumeric, underscore, hash (#), period (.), space, colon (:), at (@), equals (=), and hyphen (-) characters. Can be changed after the name is created. Minimum length = 1
servicetype	<String>	Read-write	Protocol used to exchange data with the service. Possible values = HTTP, FTP, TCP, UDP, SSL, SSL_BRIDGE, SSL_TCP, DTLS, NNTP, RPCSVR, DNS, ADNS, SNMP, RTSP, DHCPRA, ANY, SIP_UDP, SIP_TCP, SIP_SSL, DNS_TCP, ADNS_TCP, MYSQL, MSSQL, ORACLE, RADIUS,

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
			RADIUSListener, RDP, DIAMETER, SSL_DIAMETER, TFTP, SMPP, PPTP, GRE, SYSLOGTCP, SYSLOGUDP, FIX, SSL_FIX, USER_TCP, USER_SSL_TCP, QUIC, IPFIX, LOGSTREAM
cachetype	<String>	Read-write	Cache type supported by the cache server. Possible values = TRANSPARENT, REVERSE, FORWARD
td	<Double>	Read-write	Integer value that uniquely identifies the traffic domain in which you want to configure the entity. If you do not specify an ID, the entity becomes part of the default traffic domain, which has an ID of 0. Minimum value = 0 Maximum value = 4094
maxclient	<Double>	Read-write	Maximum number of simultaneous open

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
			connections for the service group. Minimum value = 0 Maximum value = 4294967294
maxreq	<Double>	Read-write	Maximum number of requests that can be sent on a persistent connection to the service group. Note: Connection requests beyond this value are rejected. Minimum value = 0 Maximum value = 65535
cacheable	<String>	Read-write	Use the transparent cache redirection virtual server to forward the request to the cache server. Note: Do not set this parameter if you set the Cache Type. Default value: NO Possible values = YES, NO
cip	<String>	Read-write	Insert the Client IP header in requests

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
			forwarded to the service. Possible values = ENABLED, DISABLED
cipheader	<String>	Read-write	Name of the HTTP header whose value must be set to the IP address of the client. Used with the Client IP parameter. If client IP insertion is enabled, and the client IP header is not specified, the value of Client IP Header parameter or the value set by the set ns config command is used as client's IP header name. Minimum length = 1
usip	<String>	Read-write	Use client's IP address as the source IP address when initiating connection to the server. With the NO setting, which is the default, a mapped IP (MIP) address or subnet IP (SNIP) address is used as the source IP address to initiate server side connections.

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
			Possible values = YES, NO
pathmonitor	<String>	Read-write	Path monitoring for clustering. Possible values = YES, NO
pathmonitorindv	<String>	Read-write	Individual Path monitoring decisions. Possible values = YES, NO
useproxyport	<String>	Read-write	Use the proxy port as the source port when initiating connections with the server. With the NO setting, the client-side connection port is used as the source port for the server-side connection. Note: This parameter is available only when the Use Source IP (USIP) parameter is set to YES. Possible values = YES, NO
healthmonitor	<String>	Read-write	Monitor the health of this service. Available settings function as follows:

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
			YES - Send probes to check the health of the service. NO - Do not send probes to check the health of the service. With the NO option, the appliance shows the service as UP at all times. Default value: YES Possible values = YES, NO
sc	<String>	Read-write	State of the SureConnect feature for the service group. Default value: OFF Possible values = ON, OFF
sp	<String>	Read-write	Enable surge protection for the service group. Default value: OFF Possible values = ON, OFF
rtspsessionidremap	<String>	Read-write	Enable RTSP session ID mapping for the service group. Default value: OFF

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
			Possible values = ON, OFF
clttimeout	<Double>	Read-write	Time, in seconds, after which to terminate an idle client connection. Minimum value = 0 Maximum value = 31536000
svrtimeout	<Double>	Read-write	Time, in seconds, after which to terminate an idle server connection. Minimum value = 0 Maximum value = 31536000
cka	<String>	Read-write	Enable client keep-alive for the service group. Possible values = YES, NO
tcpb	<String>	Read-write	Enable TCP buffering for the service group. Possible values = YES, NO
cmp	<String>	Read-write	Enable compression for the specified service. Possible values = YES, NO

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
maxbandwidth	<Double>	Read-write	Maximum bandwidth, in Kbps, allocated for all the services in the service group. Minimum value = 0 Maximum value = 4294967287
monthreshold	<Double>	Read-write	Minimum sum of weights of the monitors that are bound to this service. Used to determine whether to mark a service as UP or DOWN. Minimum value = 0 Maximum value = 65535
state	<String>	Read-write	Initial state of the service group. Default value: ENABLED Possible values = ENABLED, DISABLED
downstateflush	<String>	Read-write	Flush all active transactions associated with all the services in the service group whose state transitions from UP to DOWN. Do not enable this option for applications that

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
			must complete their transactions. Default value: ENABLED Possible values = ENABLED, DISABLED
tcpprofilename	<String>	Read-write	Name of the TCP profile that contains TCP configuration settings for the service group. Minimum length = 1 Maximum length = 127
httpprofilename	<String>	Read-write	Name of the HTTP profile that contains HTTP configuration settings for the service group. Minimum length = 1 Maximum length = 127
comment	<String>	Read-write	Any information about the service group.
appflowlog	<String>	Read-write	Enable logging of AppFlow information for the specified service group. Default value: ENABLED

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
			Possible values = ENABLED, DISABLED
netprofile	<String>	Read-write	Network profile for the service group. Minimum length = 1 Maximum length = 127
autoscale	<String>	Read-write	Auto scale option for a servicegroup. Default value: DISABLED Possible values = DISABLED, DNS, POLICY, CLOUD
memberport	<Integer>	Read-write	member port.
autodisablegraceful	<String>	Read-write	Indicates graceful shutdown of the service. System will wait for all outstanding connections to this service to be closed before disabling the service. Default value: NO Possible values = YES, NO
autodisabledelay	<Double>	Read-write	The time allowed (in seconds) for a graceful shutdown. During this period, new connections or

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
			requests will continue to be sent to this service for clients who already have a persistent session on the system. Connections or requests from fresh or new clients who do not yet have a persistence sessions on the system will not be sent to the service. Instead, they will be load balanced among other available services. After the delay time expires, no new requests or connections will be sent to the service.
monconnectionclose	<String>	Read-write	Close monitoring connections by sending the service a connection termination message with the specified bit set. Default value: NONE Possible values = RESET, FIN
servername	<String>	Read-write	Name of the server to which to bind the service group.

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
			Minimum length = 1
port	<Integer>	Read-write	Server port number. Range 1 - 65535 * in CLI is represented as 65535 in NITRO API
weight	<Double>	Read-write	Weight to assign to the servers in the service group. Specifies the capacity of the servers relative to the other servers in the load balancing configuration. The higher the weight, the higher the percentage of requests sent to the service. Minimum value = 1 Maximum value = 100
customserverid	<String>	Read-write	The identifier for this IP:Port pair. Used when the persistency type is set to Custom Server ID. Default value: "None"
serverid	<Double>	Read-write	The identifier for the service. This is used when the persistency

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
			type is set to Custom Server ID.
hashid	<Double>	Read-write	The hash identifier for the service. This must be unique for each service. This parameter is used by hash based load balancing methods. Minimum value = 1
nameserver	<String>	Read-write	Specify the nameserver to which the query for bound domain needs to be sent. If not specified, use the global nameserver.
dbsttl	<Double>	Read-write	Specify the TTL for DNS record for domain based service.The default value of ttl is 0 which indicates to use the TTL received in DNS response for monitors. Default value: 0
monitor_name_svc	<String>	Read-write	Name of the monitor bound to the service group. Used to assign a weight to the monitor. Minimum length = 1
dup_weight	<Double>	Read-write	weight of the monitor that is bound to servicegroup.

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
			Minimum value = 1
riseapbrstatsmsgcode	<Integer>	Read-write	The code indicating the rise apbr status.
delay	<Double>	Read-write	Time, in seconds, allocated for a shutdown of the services in the service group. During this period, new requests are sent to the service only for clients who already have persistent sessions on the appliance. Requests from new clients are load balanced among other available services. After the delay time expires, no requests are sent to the service, and the service is marked as unavailable (OUT OF SERVICE).
graceful	<String>	Read-write	Wait for all existing connections to the service to terminate before shutting down the service. Default value: NO Possible values = YES, NO
includemembers	<Boolean>	Read-write	Display the members of the listed service

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
			groups in addition to their settings. Can be specified when no service group name is provided in the command. In that case, the details displayed for each service group are identical to the details displayed when a service group name is provided, except that bound monitors are not displayed.
newname	<String>	Read-write	New name for the service group. Minimum length = 1
numofconnections	<Integer>	Read-only	This will tell the number of client side connections are still open.
serviceconftype	<Boolean>	Read-only	The configuration type of the service group.
value	<String>	Read-only	SSL Status. Possible values = Certkey not bound, SSL feature disabled
svrstate	<String>	Read-only	The state of the service. Possible values = UP, DOWN,

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
			UNKNOWN, BUSY, OUT OF SERVICE, GOING OUT OF SERVICE, DOWN WHEN GOING OUT OF SERVICE, NS_EMPTY_STR, Unknown, DISABLED
ip	<String>	Read-only	IP Address.
monstatcode	<Integer>	Read-only	The code indicating the monitor response.
monstatparam1	<Integer>	Read-only	First parameter for use with message code.
monstatparam2	<Integer>	Read-only	Second parameter for use with message code.
monstatparam3	<Integer>	Read-only	Third parameter for use with message code.
statechangetimemsec	<Double>	Read-only	Time when last state change occurred. Milliseconds part.
stateupdatereason	<Double>	Read-only	Checks state update reason on the secondary node.
clmonowner	<Double>	Read-only	Tells the mon owner of the service.

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
clmonview	<Double>	Read-only	Tells the view id of the monitoring owner.
groupcount	<Double>	Read-only	Servicegroup Count.
riseapbrstatsmsgcode2	<Integer>	Read-only	The code indicating other rise stats.
serviceipstr	<String>	Read-only	This field has been intorduced to show the dbs services ip.
servicegroupeffectivestate	<String>	Read-only	Indicates the effective servicegroup state based on the state of the bound service items.If all services are UP the effective state is UP, if all are DOWN its DOWN,if all are OFS its OFS.If atleast one serviceis UP and rest are either DOWN or OFS, the effective state is PARTIAL-UP.If atleast one bound service is DOWN and rest are OFS the effective state is PARTIAL DOWN. Possible values = UP, DOWN, OUT OF SERVICE, PARTIAL-UP, PARTIAL-DOWN

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
nodefaultbindings	<String>	Read-only	to determine if the configuration is from stylebooks. Default value: NO Possible values = YES, NO
svcitmactsvcs	<Double>	Read-only	This gives the total active service items for an FQDN for SRV type server binding.
svcitmboundsvcs	<Double>	Read-only	This gives the total bound items for an FQDN for SRV type server binding.
__count	<Double>	Read-only	count parameter

Operations

(click to see Properties)

ADD| DELETE| UPDATE| UNSET| ENABLE| DISABLE| RENAME| GET (ALL)| GET| COUNT

Some options that you can use for each operations:

- **Getting warnings in response:**NITRO allows you to get warnings in an operation by specifying the "warning" query parameter as "yes". For example, to get warnings while connecting to the Citrix ADC appliance, the URL is as follows:

  http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/login?warning=yes

  If any, the warnings are displayed in the response payload with the HTTP code "209 X-NITRO-WARNING".

- **Authenticated access for individual NITRO operations:**NITRO allows you to logon to the Citrix ADC appliance to perform individual operations. You can use this option instead of creating a NITRO session (using the login object) and then using that session to perform all operations,

To do this, you must specify the username and password in the request header of the NITRO request as follows:

X-NITRO-USER:<username>

X-NITRO-PASS:<password>

Note: In such cases, make sure that the request header DOES not include the following:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue>

`*Note:`*

Mandatory parameters are marked in redand placeholder content is marked in <green>.

add

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/servicegroup

HTTP Method:POST

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Content-Type:application/json

Request Payload:

```
{"servicegroup":{

<b>"servicegroupname":<String_value>,

</b><b>"servicetype":<String_value>,

</b>"cachetype":<String_value>,

"td":<Double_value>,

"maxclient":<Double_value>,

"maxreq":<Double_value>,

"cacheable":<String_value>,
```

```
"cip":<String_value>,

"cipheader":<String_value>,

"usip":<String_value>,

"pathmonitor":<String_value>,

"pathmonitorindv":<String_value>,

"useproxyport":<String_value>,

"healthmonitor":<String_value>,

"sc":<String_value>,

"sp":<String_value>,

"rtspsessionidremap":<String_value>,

"clttimeout":<Double_value>,

"svrtimeout":<Double_value>,

"cka":<String_value>,

"tcpb":<String_value>,

"cmp":<String_value>,

"maxbandwidth":<Double_value>,

"monthreshold":<Double_value>,

"state":<String_value>,

"downstateflush":<String_value>,

"tcpprofilename":<String_value>,

"httpprofilename":<String_value>,

"comment":<String_value>,
```

```
"appflowlog":<String_value>,

"netprofile":<String_value>,

"autoscale":<String_value>,

"memberport":<Integer_value>,

"autodisablegraceful":<String_value>,

"autodisabledelay":<Double_value>,

"monconnectionclose":<String_value>

}}
```

Response:

HTTP Status Code on Success: 201 Created HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error

delete

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/servicegroup/servicegroupname_value<String>

HTTP Method:DELETE

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue>

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error

update

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/servicegroup

HTTP Method:PUT

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Content-Type:application/json

Request Payload:

```
{"servicegroup":{

<b>"servicegroupname":<String_value>,

</b>"servername":<String_value>,

"port":<Integer_value>,

"weight":<Double_value>,

"customserverid":<String_value>,

"serverid":<Double_value>,

"hashid":<Double_value>,

"nameserver":<String_value>,

"dbsttl":<Double_value>,

"monitor_name_svc":<String_value>,

"dup_weight":<Double_value>,

"maxclient":<Double_value>,

"maxreq":<Double_value>,

"healthmonitor":<String_value>,

"cacheable":<String_value>,

"cip":<String_value>,
```

```
"cipheader":<String_value>,

"usip":<String_value>,

"pathmonitor":<String_value>,

"pathmonitorindv":<String_value>,

"useproxyport":<String_value>,

"sc":<String_value>,

"sp":<String_value>,

"rtspsessionidremap":<String_value>,

"clttimeout":<Double_value>,

"svrtimeout":<Double_value>,

"cka":<String_value>,

"tcpb":<String_value>,

"cmp":<String_value>,

"maxbandwidth":<Double_value>,

"monthreshold":<Double_value>,

"downstateflush":<String_value>,

"tcpprofilename":<String_value>,

"httpprofilename":<String_value>,

"comment":<String_value>,

"appflowlog":<String_value>,

"netprofile":<String_value>,

"autodisablegraceful":<String_value>,
```

```
"autodisabledelay":<Double_value>,

"monconnectionclose":<String_value>

}}
```

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error

unset

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/servicegroup?**action=unset**

HTTP Method:POST

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Content-Type:application/json

Request Payload:

```
{"servicegroup":{

<b>"servicegroupname":<String_value>,

</b>"servername":true,

"port":true,

"weight":true,

"customserverid":true,

"serverid":true,

"hashid":true,
```

```
"riseapbrstatsmsgcode":true,

"nameserver":true,

"dbsttl":true,

"maxclient":true,

"maxreq":true,

"cacheable":true,

"cip":true,

"usip":true,

"useproxyport":true,

"sc":true,

"sp":true,

"rtspsessionidremap":true,

"clttimeout":true,

"svrtimeout":true,

"cka":true,

"tcpb":true,

"cmp":true,

"maxbandwidth":true,

"monthreshold":true,

"tcpprofilename":true,

"httpprofilename":true,

"appflowlog":true,
```

```
"netprofile":true,

"autodisablegraceful":true,

"autodisabledelay":true,

"monitor_name_svc":true,

"dup_weight":true,

"healthmonitor":true,

"cipheader":true,

"pathmonitor":true,

"pathmonitorindv":true,

"downstateflush":true,

"comment":true,

"monconnectionclose":true

}}
```

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error

enable

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/servicegroup?**action=enable**

HTTP Method:POST

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Content-Type:application/json

Request Payload:

```
{"servicegroup":{

<b>"servicegroupname":<String_value>,

</b>"servername":<String_value>,

"port":<Integer_value>

}}
```

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error

disable

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/servicegroup?**action=disable**

HTTP Method:POST

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Content-Type:application/json

Request Payload:

```
{"servicegroup":{

<b>"servicegroupname":<String_value>,

</b>"servername":<String_value>,

"port":<Integer_value>,

"delay":<Double_value>,
```

```
"graceful":<String_value>

}}
```

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error

rename

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/servicegroup?**action=rename**

HTTP Method:POST

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Content-Type:application/json

Request Payload:

```
{"servicegroup":{

<b>"servicegroupname":<String_value>,

</b><b>"newname":<String_value>

</b>}}
```

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error

get (all)

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/servicegroup

Query-parameters:

args

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/servicegroup?**args=servicegroupname:<String_value>,includemembers:<Boolean_value>**

Use this query-parameter to get servicegroup resources based on additional properties.

attrs

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/servicegroup?**attrs=property-name1,property-name2**

Use this query parameter to specify the resource details that you want to retrieve.

filter

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/servicegroup?**filter=property-name1:property-val1,property-name2:property-val2**

Use this query-parameter to get the filtered set of servicegroup resources configured on Citrix ADC. Filtering can be done on any of the properties of the resource.

view

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/servicegroup?**view=summary**

Note: By default, the retrieved results are displayed in detail view (?view=detail).

pagination

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/servicegroup?**pagesize=#no;pageno=#no**

Use this query-parameter to get the servicegroup resources in chunks.

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "servicegroup": [ {

servicegroupname:<String_value>,includemembers:<Boolean_value>"numofconnec
tions":<Integer_value>,

"servicetype":<String_value>,

"port":<Integer_value>,

"td":<Double_value>,

"serviceconftpye":<Boolean_value>,

"serviceconftype":<Boolean_value>,

"value":<String_value>,

"cachetype":<String_value>,

"maxclient":<Double_value>,

"maxreq":<Double_value>,

"cacheable":<String_value>,

"cip":<String_value>,

"cipheader":<String_value>,

"usip":<String_value>,

"pathmonitor":<String_value>,
```

```
"pathmonitorindv":<String_value>,

"useproxyport":<String_value>,

"sc":<String_value>,

"sp":<String_value>,

"rtspsessionidremap":<String_value>,

"clttimeout":<Double_value>,

"svrtimeout":<Double_value>,

"cka":<String_value>,

"tcpb":<String_value>,

"cmp":<String_value>,

"maxbandwidth":<Double_value>,

"state":<String_value>,

"svrstate":<String_value>,

"delay":<Double_value>,

"ip":<String_value>,

"servername":<String_value>,

"monthreshold":<Double_value>,

"weight":<Double_value>,

"customserverid":<String_value>,

"serverid":<Double_value>,

"monstatcode":<Integer_value>,

"monstatparam1":<Integer_value>,
```

```
"monstatparam2":<Integer_value>,

"monstatparam3":<Integer_value>,

"downstateflush":<String_value>,

"statechangetimemsec":<Double_value>,

"timesincelaststatechange":<Double_value>,

"stateupdatereason":<Double_value>,

"clmonowner":<Double_value>,

"clmonview":<Double_value>,

"groupcount":<Double_value>,

"comment":<String_value>,

"tcpprofilename":<String_value>,

"httpprofilename":<String_value>,

"hashid":<Double_value>,

"riseapbrstatsmsgcode":<Integer_value>,

"riseapbrstatsmsgcode2":<Integer_value>,

"graceful":<String_value>,

"healthmonitor":<String_value>,

"appflowlog":<String_value>,

"netprofile":<String_value>,

"autoscale":<String_value>,

"memberport":<Integer_value>,

"autodisablegraceful":<String_value>,
```

```
"autodisabledelay":<Double_value>,

"serviceipstr":<String_value>,

"servicegroupeffectivestate":<String_value>,

"monconnectionclose":<String_value>,

"nodefaultbindings":<String_value>,

"nameserver":<String_value>,

"dbsttl":<Double_value>,

"svcitmactsvcs":<Double_value>,

"svcitmboundsvcs":<Double_value>

}]}
```

get

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/servicegroup/servicegroupname_value<String>

Query-parameters:

attrs

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/servicegroup/servicegroupname_value<String>?**attrs=property-name1,property-name2**

Use this query parameter to specify the resource details that you want to retrieve.

view

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/servicegroup/servicegroupname_value<String>?**view=summary**

Note: By default, the retrieved results are displayed in detail view (?view=detail).

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "servicegroup": [ {

servicegroupname:<String_value>,includemembers:<Boolean_value>"numofconnec
tions":<Integer_value>,

"servicetype":<String_value>,

"port":<Integer_value>,

"td":<Double_value>,

"serviceconftpye":<Boolean_value>,

"serviceconftype":<Boolean_value>,

"value":<String_value>,

"cachetype":<String_value>,

"maxclient":<Double_value>,

"maxreq":<Double_value>,

"cacheable":<String_value>,

"cip":<String_value>,

"cipheader":<String_value>,
```

```
"usip":<String_value>,

"pathmonitor":<String_value>,

"pathmonitorindv":<String_value>,

"useproxyport":<String_value>,

"sc":<String_value>,

"sp":<String_value>,

"rtspsessionidremap":<String_value>,

"clttimeout":<Double_value>,

"svrtimeout":<Double_value>,

"cka":<String_value>,

"tcpb":<String_value>,

"cmp":<String_value>,

"maxbandwidth":<Double_value>,

"state":<String_value>,

"svrstate":<String_value>,

"delay":<Double_value>,

"ip":<String_value>,

"servername":<String_value>,

"monthreshold":<Double_value>,

"weight":<Double_value>,

"customserverid":<String_value>,

"serverid":<Double_value>,
```

```
"monstatcode":<Integer_value>,

"monstatparam1":<Integer_value>,

"monstatparam2":<Integer_value>,

"monstatparam3":<Integer_value>,

"downstateflush":<String_value>,

"statechangetimemsec":<Double_value>,

"timesincelaststatechange":<Double_value>,

"stateupdatereason":<Double_value>,

"clmonowner":<Double_value>,

"clmonview":<Double_value>,

"groupcount":<Double_value>,

"comment":<String_value>,

"tcpprofilename":<String_value>,

"httpprofilename":<String_value>,

"hashid":<Double_value>,

"riseapbrstatsmsgcode":<Integer_value>,

"riseapbrstatsmsgcode2":<Integer_value>,

"graceful":<String_value>,

"healthmonitor":<String_value>,

"appflowlog":<String_value>,

"netprofile":<String_value>,

"autoscale":<String_value>,
```

```
"memberport":<Integer_value>,

"autodisablegraceful":<String_value>,

"autodisabledelay":<Double_value>,

"serviceipstr":<String_value>,

"servicegroupeffectivestate":<String_value>,

"monconnectionclose":<String_value>,

"nodefaultbindings":<String_value>,

"nameserver":<String_value>,

"dbsttl":<Double_value>,

"svcitmactsvcs":<Double_value>,

"svcitmboundsvcs":<Double_value>

}]}
```

count

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/servicegroup?**count=yes**

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "servicegroup": [ { "__count": "#no"} ] }
```

servicegroupbindings

Configuration for servicegroupbind resource.

Properties

(click to see Operations)

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
servicegroupname	<String>	Read-write	The name of the service. Minimum length = 1
ipaddress	<String>	Read-only	The IP address of the vserver.
port	<Integer>	Read-only	The port of the vserver. Range 1 - 65535 * in CLI is represented as 65535 in NITRO API
state	<String>	Read-only	The state of the service group. Possible values = ENABLED, DISABLED
svrstate	<String>	Read-only	The state of the vserver. Possible values = UP, DOWN, UNKNOWN, BUSY,

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
			OUT OF SERVICE, GOING OUT OF SERVICE, DOWN WHEN GOING OUT OF SERVICE, NS_EMPTY_STR, Unknown, DISABLED
vservername	<String>	Read-only	The name of the vserver.
__count	<Double>	Read-only	count parameter

Operations

(click to see Properties)

GET (ALL)| GET| COUNT

Some options that you can use for each operations:

- **Getting warnings in response:**NITRO allows you to get warnings in an operation by specifying the "warning" query parameter as "yes". For example, to get warnings while connecting to the Citrix ADC appliance, the URL is as follows:

  http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/login?warning=yes

  If any, the warnings are displayed in the response payload with the HTTP code "209 X-NITRO-WARNING".

- **Authenticated access for individual NITRO operations:**NITRO allows you to logon to the Citrix ADC appliance to perform individual operations. You can use this option instead of creating a NITRO session (using the login object) and then using that session to perform all operations,

  To do this, you must specify the username and password in the request header of the NITRO request as follows:

  X-NITRO-USER:<username>

X-NITRO-PASS:<password>

Note: In such cases, make sure that the request header DOES not include the following:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue>

`*Note:`*

Mandatory parameters are marked in redand placeholder content is marked in <green>.

get (all)

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/servicegroupbindings

Query-parameters:

args

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/servicegroupbindings?**args=servicegroupname:<String_value>,**

Use this query-parameter to get servicegroupbindings resources based on additional properties.

attrs

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/servicegroupbindings?**attrs=property-name1,property-name2**

Use this query parameter to specify the resource details that you want to retrieve.

filter

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/servicegroupbindings?**filter=property-name1:property-val1,property-name2:property-val2**

Use this query-parameter to get the filtered set of servicegroupbindings resources configured on Citrix ADC. Filtering can be done on any of the properties of the resource.

view

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/servicegroupbindings?**view=summary**

Note: By default, the retrieved results are displayed in detail view (?view=detail).

pagination

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/servicegroupbindings?**pagesize=#no;pageno=#no**

Use this query-parameter to get the servicegroupbindings resources in chunks.

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "servicegroupbindings": [ {

<b>servicegroupname:<String_value>,</b>"ipaddress":<String_value>,

"port":<Integer_value>,

"state":<String_value>,

"svrstate":<String_value>,

"vservername":<String_value>

}]}
```

get

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/servicegroupbindings/servicegroupname_value<String>

Query-parameters:

attrs

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/servicegroupbindings/servicegroupname_value<String>?**attrs=property-name1,property-name2**

Use this query parameter to specify the resource details that you want to retrieve.

view

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/servicegroupbindings/servicegroupname_value<String>?**view=summary**

Note: By default, the retrieved results are displayed in detail view (?view=detail).

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "servicegroupbindings": [ {

<b>servicegroupname:<String_value>,</b>"ipaddress":<String_value>,

"port":<Integer_value>,

"state":<String_value>,

"svrstate":<String_value>,

"vservername":<String_value>
```

```
}]}
```

count

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/servicegroupbindings?**args=servicegroupname:<String_value>,;count=yes**

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "servicegroupbindings": [ { "__count": "#no"} ] }
```

servicegroup_binding

Binding object which returns the resources bound to servicegroup.

Properties

(click to see Operations)

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
servicegroupname	<String>	Read-write	Name of the service group. Minimum length = 1

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
servicegroup_servicegroupentitymonbindings_binding	<servicegroup_servicegroupentitymonbindings_binding[]>	Read-only	servicegroupentitymonbindings that can be bound to servicegroup.
servicegroup_lbmonitor_binding	<servicegroup_lbmonitor_binding[]>	Read-only	lbmonitor that can be bound to servicegroup.
servicegroup_servicegroupmember_binding	<servicegroup_servicegroupmember_binding[]>	Read-only	servicegroupmember that can be bound to servicegroup.

Operations

(click to see Properties)

GET| GET (ALL)

Some options that you can use for each operations:

- **Getting warnings in response:**NITRO allows you to get warnings in an operation by specifying the "warning" query parameter as "yes". For example, to get warnings while connecting to the Citrix ADC appliance, the URL is as follows:

  http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/login?warning=yes

  If any, the warnings are displayed in the response payload with the HTTP code "209 X-NITRO-WARNING".

- **Authenticated access for individual NITRO operations:**NITRO allows you to logon to the Citrix ADC appliance to perform individual operations. You can use this option instead of creating a NITRO session (using the login object) and then using that session to perform all operations,

  To do this, you must specify the username and password in the request header of the NITRO request as follows:

  X-NITRO-USER:<username>

  X-NITRO-PASS:<password>

Note: In such cases, make sure that the request header DOES not include the following:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue>

`*Note:`*

Mandatory parameters are marked in redand placeholder content is marked in <green>.

get

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/servicegroup_binding/servicegroupname_value<String>

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "servicegroup_binding": [ {

"servicegroupname":<String_value>,

"servicegroup_lbmonitor_binding":<servicegroup_lbmonitor_binding[]_value>,

"servicegroup_servicegroupmember_binding":<servicegroup_servicegroupmember
_binding[]_value>,

"servicegroup_servicegroupentitymonbindings_binding":<servicegroup_service
groupentitymonbindings_binding[]_value>

}]}
```

get (all)

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/servicegroup_binding

Query-parameters:

bulkbindings

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/servicegroup_binding?**bulkbindings=yes**

NITRO allows you to fetch bindings in bulk.

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "servicegroup_binding": [ {

"servicegroupname":<String_value>,

"servicegroup_lbmonitor_binding":<servicegroup_lbmonitor_binding[]_value>,

"servicegroup_servicegroupmember_binding":<servicegroup_servicegroupmember
_binding[]_value>,

"servicegroup_servicegroupentitymonbindings_binding":<servicegroup_service
groupentitymonbindings_binding[]_value>

}]}
```

servicegroup_lbmonitor_binding

Binding object showing the lbmonitor that can be bound to servicegroup.

Properties

(click to see Operations)

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
servicegroupname	<String>	Read-write	Name of the service group. Minimum length = 1
port	<Integer>	Read-write	Port number of the service. Each service must have a unique port number. Range 1 - 65535 * in CLI is represented as 65535 in NITRO API
nameserver	<String>	Read-write	Specify the nameserver to which the query for bound domain needs to be sent. If not specified, use the global nameserver.
state	<String>	Read-write	Initial state of the service after binding. Default value: ENABLED

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
			Possible values = ENABLED, DISABLED
hashid	<Double>	Read-write	Unique numerical identifier used by hash based load balancing methods to identify a service. Minimum value = 1
serverid	<Double>	Read-write	The identifier for the service. This is used when the persistency type is set to Custom Server ID.
customserverid	<String>	Read-write	Unique service identifier. Used when the persistency type for the virtual server is set to Custom Server ID. Default value: "None"
weight	<Double>	Read-write	Weight to assign to the servers in the service group. Specifies the capacity of the servers relative to the other servers in the load balancing configuration. The higher the weight, the higher the percentage of requests sent to the service.

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
			Minimum value = 1 Maximum value = 100
monitor_name	<String>	Read-write	Monitor name.
dbsttl	<Double>	Read-write	Specify the TTL for DNS record for domain based service.The default value of ttl is 0 which indicates to use the TTL received in DNS response for monitors. Default value: 0
passive	<Boolean>	Read-write	Indicates if load monitor is passive. A passive load monitor does not remove service from LB decision when threshold is breached.
monstate	<String>	Read-write	Monitor state. Possible values = ENABLED, DISABLED
monweight	<Double>	Read-only	weight of the monitor that is bound to servicegroup.
__count	<Double>	Read-only	count parameter

Operations

(click to see Properties)

ADD:| DELETE:| GET| GET (ALL)| COUNT

Some options that you can use for each operations:

- **Getting warnings in response:**NITRO allows you to get warnings in an operation by specifying the "warning" query parameter as "yes". For example, to get warnings while connecting to the Citrix ADC appliance, the URL is as follows:

  http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/login?warning=yes

  If any, the warnings are displayed in the response payload with the HTTP code "209 X-NITRO-WARNING".

- **Authenticated access for individual NITRO operations:**NITRO allows you to logon to the Citrix ADC appliance to perform individual operations. You can use this option instead of creating a NITRO session (using the login object) and then using that session to perform all operations,

  To do this, you must specify the username and password in the request header of the NITRO request as follows:

  X-NITRO-USER:<username>

  X-NITRO-PASS:<password>

  Note: In such cases, make sure that the request header DOES not include the following:

  Cookie:NITRO_AUTH_TOKEN=<tokenvalue>

`*Note:`*

Mandatory parameters are marked in redand placeholder content is marked in <green>.

add

URL:http://<Citrix-ADC-IP-address(NSIP)/nitro/v1/config/servicegroup_lbmonitor_binding

HTTP Method:PUT

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Content-Type:application/json

Request Payload:

```
{
"servicegroup_lbmonitor_binding":{
<b>"servicegroupname":<String_value>,
</b>"port":<Integer_value>,
"monitor_name":<String_value>,
"monstate":<String_value>,
"passive":<Boolean_value>,
"weight":<Double_value>,
"customserverid":<String_value>,
"serverid":<Double_value>,
"state":<String_value>,
"hashid":<Double_value>,
"nameserver":<String_value>,
"dbsttl":<Double_value>
}}
```

Response:

HTTP Status Code on Success: 201 Created HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error

delete

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/servicegroup_lbmonitor_binding/servicegroupname_value<String>

Query-parameters:

args

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/servicegroup_lbmonitor_binding/servicegroupname_value<String>?**args=port:<Integer_value>,monitor_name:<String_value>**

HTTP Method:DELETE

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue>

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error

get

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/servicegroup_lbmonitor_binding/servicegroupname_value<String>

Query-parameters:

filter

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/servicegroup_lbmonitor_binding/servicegroupname_value<String>?**filter=property-name1:property-value1,property-name2:property-value2**

Use this query-parameter to get the filtered set of servicegroup_lbmonitor_binding resources configured on Citrix ADC. Filtering can be done on any of the properties of the resource.

pagination

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/servicegroup_lbmonitor_binding/servicegroupname_value<String>?**pagesize=#no;pageno=#no**

Use this query-parameter to get the servicegroup_lbmonitor_binding resources in chunks.

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "servicegroup_lbmonitor_binding": [ {

"servicegroupname":<String_value>,

"port":<Integer_value>,

"nameserver":<String_value>,

"state":<String_value>,

"hashid":<Double_value>,

"serverid":<Double_value>,

"customserverid":<String_value>,

"weight":<Double_value>,

"monitor_name":<String_value>,

"dbsttl":<Double_value>,

"passive":<Boolean_value>,

"monstate":<String_value>,

"monweight":<Double_value>

}]}
```

get (all)

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/servicegroup_lbmonitor_binding

Query-parameters:

bulkbindings

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/servicegroup_lbmonitor_binding?**bulkbindings=yes**

NITRO allows you to fetch bindings in bulk.

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "servicegroup_lbmonitor_binding": [ {

"servicegroupname":<String_value>,

"port":<Integer_value>,

"nameserver":<String_value>,

"state":<String_value>,

"hashid":<Double_value>,
```

```
"serverid":<Double_value>,

"customserverid":<String_value>,

"weight":<Double_value>,

"monitor_name":<String_value>,

"dbsttl":<Double_value>,

"passive":<Boolean_value>,

"monstate":<String_value>,

"monweight":<Double_value>

}]}
```

count

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/servicegroup_lbmonitor_binding/servicegroupname_value<String>?**count=yes**

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{"servicegroup_lbmonitor_binding": [ { "__count": "#no"} ] }
```

servicegroup_servicegroupentitymonbindings_binding

Binding object showing the servicegroupentitymonbindings that can be bound to servicegroup.

Properties

(click to see Operations)

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
servicegroupname	<String>	Read-write	Name of the service group. Minimum length = 1
servicegroupentname2	<String>	Read-write	.
port	<Integer>	Read-write	Port number of the service. Each service must have a unique port number. Range 1 - 65535 * in CLI is represented as 65535 in NITRO API
nameserver	<String>	Read-write	Specify the nameserver to which the query for bound domain needs to be sent. If not specified, use the global nameserver.
state	<String>	Read-write	Initial state of the service after binding. Default value: ENABLED

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
			Possible values = ENABLED, DISABLED
hashid	<Double>	Read-write	Unique numerical identifier used by hash based load balancing methods to identify a service. Minimum value = 1
serverid	<Double>	Read-write	The identifier for the service. This is used when the persistency type is set to Custom Server ID.
customserverid	<String>	Read-write	Unique service identifier. Used when the persistency type for the virtual server is set to Custom Server ID. Default value: "None"
weight	<Double>	Read-write	. Default value: 1 Minimum value = 1 Maximum value = 100
monitor_name	<String>	Read-write	Monitor name.
dbsttl	<Double>	Read-write	Specify the TTL for DNS record for domain based service.The default value of ttl is 0 which

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
			indicates to use the TTL received in DNS response for monitors. Default value: 0
passive	<Boolean>	Read-write	Indicates if load monitor is passive. A passive load monitor does not remove service from LB decision when threshold is breached.
lastresponse	<String>	Read-only	The string form of monstatcode.
monitor_state	<String>	Read-only	The running state of the monitor on this service. Possible values = UP, DOWN, UNKNOWN, BUSY, OUT OF SERVICE, GOING OUT OF SERVICE, DOWN WHEN GOING OUT OF SERVICE, NS_EMPTY_STR, Unknown, DISABLED
monitortotalprobes	<Double>	Read-only	Total number of probes sent to monitor this service.
monitortotalfailedprobes	<Double>	Read-only	Total number of failed probes.

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
monitorcurrentfailedprobes	<Double>	Read-only	Total number of currently failed probes.
__count	<Double>	Read-only	count parameter

Operations

(click to see Properties)

GET| GET (ALL)| COUNT

Some options that you can use for each operations:

- **Getting warnings in response:**NITRO allows you to get warnings in an operation by specifying the "warning" query parameter as "yes". For example, to get warnings while connecting to the Citrix ADC appliance, the URL is as follows:

  http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/login?warning=yes

  If any, the warnings are displayed in the response payload with the HTTP code "209 X-NITRO-WARNING".

- **Authenticated access for individual NITRO operations:**NITRO allows you to logon to the Citrix ADC appliance to perform individual operations. You can use this option instead of creating a NITRO session (using the login object) and then using that session to perform all operations,

  To do this, you must specify the username and password in the request header of the NITRO request as follows:

  X-NITRO-USER:<username>

  X-NITRO-PASS:<password>

  Note: In such cases, make sure that the request header DOES not include the following:

  Cookie:NITRO_AUTH_TOKEN=<tokenvalue>

`*Note:`*

Mandatory parameters are marked in redand placeholder content is marked in <green>.

get

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/servicegroup_servicegroupentitymonbindings_binding/servicegroupname_value<String>

Query-parameters:

filter

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/servicegroup_servicegroupentitymonbindings_binding/servicegroupname_value<String>?**filter=property-name1:property-value1,property-name2:property-value2**

Use this query-parameter to get the filtered set of servicegroup_servicegroupentitymonbindings_binding resources configured on Citrix ADC. Filtering can be done on any of the properties of the resource.

pagination

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/servicegroup_servicegroupentitymonbindings_binding/servicegroupname_value<String>?**pagesize=#no;pageno=#no**

Use this query-parameter to get the servicegroup_servicegroupentitymonbindings_binding resources in chunks.

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "servicegroup_servicegroupentitymonbindings_binding": [ {
"servicegroupname":<String_value>,
"servicegroupentname2":<String_value>,
"port":<Integer_value>,
"nameserver":<String_value>,
"state":<String_value>,
"hashid":<Double_value>,
"serverid":<Double_value>,
"customserverid":<String_value>,
"weight":<Double_value>,
"monitor_name":<String_value>,
"dbsttl":<Double_value>,
"passive":<Boolean_value>,
"lastresponse":<String_value>,
"monitor_state":<String_value>,
"monitortotalprobes":<Double_value>,
"monitortotalfailedprobes":<Double_value>,
"monitorcurrentfailedprobes":<Double_value>
}]}
```

get (all)

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/servicegroup_servicegroupentitymonbindings_binding

Query-parameters:

bulkbindings

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/servicegroup_servicegroupentitymonbindings_binding?**bulkbindings=yes**

NITRO allows you to fetch bindings in bulk.

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "servicegroup_servicegroupentitymonbindings_binding": [ {

"servicegroupname":<String_value>,

"servicegroupentname2":<String_value>,

"port":<Integer_value>,

"nameserver":<String_value>,

"state":<String_value>,

"hashid":<Double_value>,
```

```
"serverid":<Double_value>,

"customserverid":<String_value>,

"weight":<Double_value>,

"monitor_name":<String_value>,

"dbsttl":<Double_value>,

"passive":<Boolean_value>,

"lastresponse":<String_value>,

"monitor_state":<String_value>,

"monitortotalprobes":<Double_value>,

"monitortotalfailedprobes":<Double_value>,

"monitorcurrentfailedprobes":<Double_value>

}]}
```

count

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/servicegroup_servicegroupentitymonbindings_binding/servicegroupname_value<String>?**count=yes**

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{"servicegroup_servicegroupentitymonbindings_binding": [ { "__count":
"#no"} ] }
```

servicegroup_servicegroupmember_binding

Binding object showing the servicegroupmember that can be bound to servicegroup.

Properties

(click to see Operations)

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
servicegroupname	<String>	Read-write	Name of the service group. Minimum length = 1
ip	<String>	Read-write	IP Address.
port	<Integer>	Read-write	Server port number. Range 1 - 65535 * in CLI is represented as 65535 in NITRO API
nameserver	<String>	Read-write	Specify the nameserver to which the query for bound domain needs to be sent. If not specified, use the global nameserver.

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
state	<String>	Read-write	Initial state of the service group. Default value: ENABLED Possible values = ENABLED, DISABLED
hashid	<Double>	Read-write	The hash identifier for the service. This must be unique for each service. This parameter is used by hash based load balancing methods. Minimum value = 1
serverid	<Double>	Read-write	The identifier for the service. This is used when the persistency type is set to Custom Server ID.
servername	<String>	Read-write	Name of the server to which to bind the service group. Minimum length = 1
customserverid	<String>	Read-write	The identifier for this IP:Port pair. Used when the persistency type is set to Custom Server ID. Default value: "None"
weight	<Double>	Read-write	Weight to assign to the servers in the service group.

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
			Specifies the capacity of the servers relative to the other servers in the load balancing configuration. The higher the weight, the higher the percentage of requests sent to the service. Minimum value = 1 Maximum value = 100
dbsttl	<Double>	Read-write	Specify the TTL for DNS record for domain based service.The default value of ttl is 0 which indicates to use the TTL received in DNS response for monitors. Default value: 0
svcitmpriority	<Double>	Read-only	This gives the priority of the FQDN service items for SRV server binding.
delay	<Double>	Read-only	Time, in seconds, allocated for a shutdown of the services in the service group. During this period, new requests are sent to the service only for clients who

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
			already have persistent sessions on the appliance. Requests from new clients are load balanced among other available services. After the delay time expires, no requests are sent to the service, and the service is marked as unavailable (OUT OF SERVICE).
statechangetimesec	<String>	Read-only	Time when last state change occurred. Seconds part.
svrstate	<String>	Read-only	The state of the service. Possible values = UP, DOWN, UNKNOWN, BUSY, OUT OF SERVICE, GOING OUT OF SERVICE, DOWN WHEN GOING OUT OF SERVICE, NS_EMPTY_STR, Unknown, DISABLED
tickssincelaststatechange	<Double>	Read-only	Time in 10 millisecond ticks since the last state change.

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
graceful	<String>	Read-only	Wait for all existing connections to the service to terminate before shutting down the service. Default value: NO Possible values = YES, NO
__count	<Double>	Read-only	count parameter

Operations

(click to see Properties)

ADD:| DELETE:| GET| GET (ALL)| COUNT

Some options that you can use for each operations:

- **Getting warnings in response:**NITRO allows you to get warnings in an operation by specifying the "warning" query parameter as "yes". For example, to get warnings while connecting to the Citrix ADC appliance, the URL is as follows:

  http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/login?warning=yes

  If any, the warnings are displayed in the response payload with the HTTP code "209 X-NITRO-WARNING".

- **Authenticated access for individual NITRO operations:**NITRO allows you to logon to the Citrix ADC appliance to perform individual operations. You can use this option instead of creating a NITRO session (using the login object) and then using that session to perform all operations,

  To do this, you must specify the username and password in the request header of the NITRO request as follows:

  X-NITRO-USER:<username>

  X-NITRO-PASS:<password>

  Note: In such cases, make sure that the request header DOES not include the following:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue>

`*Note:`*

Mandatory parameters are marked in redand placeholder content is marked in <green>.

add

URL:http://<Citrix-ADC-IP-address(NSIP)/nitro/v1/config/servicegroup_servicegroupmember_binding

HTTP Method:PUT

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Content-Type:application/json

Request Payload:

```
{

"servicegroup_servicegroupmember_binding":{

<b>"servicegroupname":<String_value>,

</b>"ip":<String_value>,

"servername":<String_value>,

"port":<Integer_value>,

"weight":<Double_value>,

"customserverid":<String_value>,

"serverid":<Double_value>,

"state":<String_value>,

"hashid":<Double_value>,

"nameserver":<String_value>,
```

```
"dbsttl":<Double_value>

}}
```

Response:

HTTP Status Code on Success: 201 Created HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error

delete

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/servicegroup_servicegroupmember_binding/servicegroupname_value<String>

Query-parameters:

args

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/servicegroup_servicegroupmember_binding/servicegroupname_value<String>?
args=ip:<String_value>,servername:<String_value>,port:<Integer_value>

HTTP Method:DELETE

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue>

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error

get

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/servicegroup_servicegroupmember_binding/servicegroupname_value<String>

Query-parameters:

filter

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/servicegroup_servicegroupmember_binding/servicegroupname_value<String>?**filter=property-name1:property-value1,property-name2:property-value2**

Use this query-parameter to get the filtered set of servicegroup_servicegroupmember_binding resources configured on Citrix ADC. Filtering can be done on any of the properties of the resource.

pagination

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/servicegroup_servicegroupmember_binding/servicegroupname_value<String>?**pagesize=#no;pageno=#no**

Use this query-parameter to get the servicegroup_servicegroupmember_binding resources in chunks.

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "servicegroup_servicegroupmember_binding": [ {

"servicegroupname":<String_value>,

"ip":<String_value>,

"port":<Integer_value>,

"nameserver":<String_value>,

"state":<String_value>,
```

```
"hashid":<Double_value>,

"serverid":<Double_value>,

"servername":<String_value>,

"customserverid":<String_value>,

"weight":<Double_value>,

"dbsttl":<Double_value>,

"svcitmpriority":<Double_value>,

"delay":<Double_value>,

"statechangetimesec":<String_value>,

"svrstate":<String_value>,

"tickssincelaststatechange":<Double_value>,

"graceful":<String_value>

}]}
```

get (all)

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/servicegroup_servicegroupmember_binding

Query-parameters:

bulkbindings

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/servicegroup_servicegroupmember_binding?**bulkbindings=yes**

NITRO allows you to fetch bindings in bulk.

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "servicegroup_servicegroupmember_binding": [ {

"servicegroupname":<String_value>,

"ip":<String_value>,

"port":<Integer_value>,

"nameserver":<String_value>,

"state":<String_value>,

"hashid":<Double_value>,

"serverid":<Double_value>,

"servername":<String_value>,

"customserverid":<String_value>,

"weight":<Double_value>,

"dbsttl":<Double_value>,

"svcitmpriority":<Double_value>,

"delay":<Double_value>,
```

```
"statechangetimesec":<String_value>,

"svrstate":<String_value>,

"tickssincelaststatechange":<Double_value>,

"graceful":<String_value>

}]}
```

count

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/servicegroup_servicegroupmember_binding/servicegroupname_value<String>?**count=yes**

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{"servicegroup_servicegroupmember_binding": [ { "__count": "#no"} ] }
```

service_binding

Binding object which returns the resources bound to service.

Properties

(click to see Operations)

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
name	<String>	Read-write	Name of the service for which to display configuration details. Minimum length = 1
service_lbmonitor_binding	<service_lbmonitor_binding[]>	Read-only	lbmonitor that can be bound to service.
service_dospolicy_binding	<service_dospolicy_binding[]>	Read-only	dospolicy that can be bound to service.
service_scpolicy_binding	<service_scpolicy_binding[]>	Read-only	scpolicy that can be bound to service.

Operations

(click to see Properties)

GET| GET (ALL)

Some options that you can use for each operations:

- **Getting warnings in response:**NITRO allows you to get warnings in an operation by specifying the "warning" query parameter as "yes". For example, to get warnings while connecting to the Citrix ADC appliance, the URL is as follows:

  http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/login?warning=yes

  If any, the warnings are displayed in the response payload with the HTTP code "209 X-NITRO-WARNING".

- **Authenticated access for individual NITRO operations:**NITRO allows you to logon to the Citrix ADC appliance to perform individual operations. You can use this option instead of creating a NITRO session (using the login object) and then using that session to perform all operations,

  To do this, you must specify the username and password in the request header of the NITRO request as follows:

X-NITRO-USER:<username>

X-NITRO-PASS:<password>

Note: In such cases, make sure that the request header DOES not include the following:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue>

`*Note:`*

Mandatory parameters are marked in redand placeholder content is marked in <green>.

get

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/service_binding/name_value<String>

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "service_binding": [ {

"name":<String_value>,

"service_scpolicy_binding":<service_scpolicy_binding[]_value>,

"service_lbmonitor_binding":<service_lbmonitor_binding[]_value>,

"service_dospolicy_binding":<service_dospolicy_binding[]_value>

}]}
```

get (all)

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/service_binding

Query-parameters:

bulkbindings

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/service_binding?**bulkbindings=yes**

NITRO allows you to fetch bindings in bulk.

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "service_binding": [ {

"name":<String_value>,

"service_scpolicy_binding":<service_scpolicy_binding[]_value>,

"service_lbmonitor_binding":<service_lbmonitor_binding[]_value>,

"service_dospolicy_binding":<service_dospolicy_binding[]_value>

}]}
```

service_dospolicy_binding

Binding object showing the dospolicy that can be bound to service.

Properties

(click to see Operations)

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
policyname	<String>	Read-write	The name of the policyname for which this service is bound.
name	<String>	Read-write	Name of the service to which to bind a policy or monitor. Minimum length = 1
__count	<Double>	Read-write	count parameter

Operations

(click to see Properties)

ADD:| DELETE:| GET| GET (ALL)| COUNT

Some options that you can use for each operations:

- **Getting warnings in response:**NITRO allows you to get warnings in an operation by specifying the "warning" query parameter as "yes". For example, to get warnings while connecting to the Citrix ADC appliance, the URL is as follows:

  http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/login?warning=yes

  If any, the warnings are displayed in the response payload with the HTTP code "209 X-NITRO-WARNING".

- **Authenticated access for individual NITRO operations:**NITRO allows you to logon to the Citrix ADC appliance to perform individual operations. You can use this option instead of creating a NITRO session (using the login object) and then using that session to perform all operations,

  To do this, you must specify the username and password in the request header of the NITRO request as follows:

X-NITRO-USER:<username>

X-NITRO-PASS:<password>

Note: In such cases, make sure that the request header DOES not include the following:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue>

`*Note:`*

Mandatory parameters are marked in redand placeholder content is marked in <green>.

add

URL:http://<Citrix-ADC-IP-address(NSIP)/nitro/v1/config/service_dospolicy_binding

HTTP Method:PUT

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Content-Type:application/json

Request Payload:

```
{

"service_dospolicy_binding":{

<b>"name":<String_value>,

</b>"policyname":<String_value>

}}
```

Response:

HTTP Status Code on Success: 201 Created HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error

delete

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/service_dospolicy_binding/name_value<String>

Query-parameters:

args

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/service_dospolicy_binding/name_value<String>?**args=policyname:<String_value>**

HTTP Method:DELETE

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue>

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error

get

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/service_dospolicy_binding/name_value<String>

Query-parameters:

filter

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/service_dospolicy_binding/name_value<String>?**filter=property-name1:property-value1,property-name2:property-value2**

Use this query-parameter to get the filtered set of service_dospolicy_binding resources configured on Citrix ADC. Filtering can be done on any of the properties of the resource.

pagination

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/service_dospolicy_binding/name_value<String>?**pagesize=#no;pageno=#no**

Use this query-parameter to get the service_dospolicy_binding resources in chunks.

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "service_dospolicy_binding": [ {

"policyname":<String_value>,

"name":<String_value>

}]}
```

get (all)

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/service_dospolicy_binding

Query-parameters:

bulkbindings

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/service_dospolicy_binding?**bulkbindings=yes**

NITRO allows you to fetch bindings in bulk.

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "service_dospolicy_binding": [ {

"policyname":<String_value>,

"name":<String_value>

}]}
```

count

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/service_dospolicy_binding/name_value<String>?**count=yes**

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{"service_dospolicy_binding": [ { "__count": "#no"} ] }
```

service_lbmonitor_binding

Binding object showing the lbmonitor that can be bound to service.

Properties

(click to see Operations)

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
weight	<Double>	Read-write	Weight to assign to the monitor-service binding. When a monitor is UP, the weight assigned to its binding with the service determines how much the monitor contributes toward keeping the health of the service above the value configured for the Monitor Threshold parameter. Minimum value = 1 Maximum value = 100
name	<String>	Read-write	Name of the service to which to bind a policy or monitor. Minimum length = 1
passive	<Boolean>	Read-write	Indicates if load monitor is passive. A

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
			passive load monitor does not remove service from LB decision when threshold is breached.
monitor_name	<String>	Read-write	The monitor Names.
monstate	<String>	Read-write	The configured state (enable/disable) of the monitor on this server. Possible values = ENABLED, DISABLED
monstatcode	<Integer>	Read-only	The code indicating the monitor response.
dup_weight	<Double>	Read-only	The weight of the monitor.
responsetime	<Double>	Read-only	Response time of this monitor.
totalfailedprobes	<Double>	Read-only	The total number of failed probs.
monstatparam2	<Integer>	Read-only	Second parameter for use with message code.
lastresponse	<String>	Read-only	The string form of monstatcode.
failedprobes	<Double>	Read-only	Number of the current failed monitoring probes.

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
monstatparam3	<Integer>	Read-only	Third parameter for use with message code.
totalprobes	<Double>	Read-only	The total number of probs sent.
monitor_state	<String>	Read-only	The running state of the monitor on this service. Possible values = UP, DOWN, UNKNOWN, BUSY, OUT OF SERVICE, GOING OUT OF SERVICE, DOWN WHEN GOING OUT OF SERVICE, NS_EMPTY_STR, Unknown, DISABLED
dup_state	<String>	Read-only	Added this field for getting state value from table. Possible values = ENABLED, DISABLED
monitortotalprobes	<Double>	Read-only	Total number of probes sent to monitor this service.
monstatparam1	<Integer>	Read-only	First parameter for use with message code.
monitortotalfailedprobes	<Double>	Read-only	Total number of failed probes.

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
monitorcurrentfailedprobes	<Double>	Read-only	Total number of currently failed probes.
__count	<Double>	Read-only	count parameter

Operations

(click to see Properties)

ADD:| DELETE:| GET| GET (ALL)| COUNT

Some options that you can use for each operations:

- **Getting warnings in response:**NITRO allows you to get warnings in an operation by specifying the "warning" query parameter as "yes". For example, to get warnings while connecting to the Citrix ADC appliance, the URL is as follows:

  http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/login?warning=yes

  If any, the warnings are displayed in the response payload with the HTTP code "209 X-NITRO-WARNING".

- **Authenticated access for individual NITRO operations:**NITRO allows you to logon to the Citrix ADC appliance to perform individual operations. You can use this option instead of creating a NITRO session (using the login object) and then using that session to perform all operations,

  To do this, you must specify the username and password in the request header of the NITRO request as follows:

  X-NITRO-USER:<username>

  X-NITRO-PASS:<password>

  Note: In such cases, make sure that the request header DOES not include the following:

  Cookie:NITRO_AUTH_TOKEN=<tokenvalue>

`*Note:` *

Mandatory parameters are marked in redand placeholder content is marked in <green>.

add

URL:http://<Citrix-ADC-IP-address(NSIP)/nitro/v1/config/service_lbmonitor_binding

HTTP Method:PUT

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Content-Type:application/json

Request Payload:

```
{

"service_lbmonitor_binding":{

<b>"name":<String_value>,

</b>"monitor_name":<String_value>,

"monstate":<String_value>,

"weight":<Double_value>,

"passive":<Boolean_value>

}}
```

Response:

HTTP Status Code on Success: 201 Created HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error

delete

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/service_lbmonitor_binding/name_value<String>

Query-parameters:

args

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/service_lbmonitor_binding/name_value<String>?**args=monitor_name:<String_value>**

HTTP Method:DELETE

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue>

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error

get

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/service_lbmonitor_binding/name_value<String>

Query-parameters:

filter

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/service_lbmonitor_binding/name_value<String>?**filter=property-name1:property-value1,property-name2:property-value2**

Use this query-parameter to get the filtered set of service_lbmonitor_binding resources configured on Citrix ADC. Filtering can be done on any of the properties of the resource.

pagination

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/service_lbmonitor_binding/name_value<String>?**pagesize=#no;pageno=#no**

Use this query-parameter to get the service_lbmonitor_binding resources in chunks.

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "service_lbmonitor_binding": [ {

"weight":<Double_value>,

"name":<String_value>,

"passive":<Boolean_value>,

"monitor_name":<String_value>,

"monstate":<String_value>,

"monstatcode":<Integer_value>,

"dup_weight":<Double_value>,

"responsetime":<Double_value>,

"totalfailedprobes":<Double_value>,

"monstatparam2":<Integer_value>,

"lastresponse":<String_value>,

"failedprobes":<Double_value>,

"monstatparam3":<Integer_value>,

"totalprobes":<Double_value>,

"monitor_state":<String_value>,

"dup_state":<String_value>,
```

```
"monitortotalprobes":<Double_value>,

"monstatparam1":<Integer_value>,

"monitortotalfailedprobes":<Double_value>,

"monitorcurrentfailedprobes":<Double_value>

}]}
```

get (all)

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/service_lbmonitor_binding

Query-parameters:

bulkbindings

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/service_lbmonitor_binding?**bulkbindings=yes**

NITRO allows you to fetch bindings in bulk.

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "service_lbmonitor_binding": [ {
```

```
"weight":<Double_value>,

"name":<String_value>,

"passive":<Boolean_value>,

"monitor_name":<String_value>,

"monstate":<String_value>,

"monstatcode":<Integer_value>,

"dup_weight":<Double_value>,

"responsetime":<Double_value>,

"totalfailedprobes":<Double_value>,

"monstatparam2":<Integer_value>,

"lastresponse":<String_value>,

"failedprobes":<Double_value>,

"monstatparam3":<Integer_value>,

"totalprobes":<Double_value>,

"monitor_state":<String_value>,

"dup_state":<String_value>,

"monitortotalprobes":<Double_value>,

"monstatparam1":<Integer_value>,

"monitortotalfailedprobes":<Double_value>,

"monitorcurrentfailedprobes":<Double_value>

}]}
```

count

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/service_lbmonitor_binding/name_value<String>?**count=yes**

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{"service_lbmonitor_binding": [ { "__count": "#no"} ] }
```

service_scpolicy_binding

Binding object showing the scpolicy that can be bound to service.

Properties

(click to see Operations)

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
policyname	<String>	Read-write	The name of the policyname for which this service is bound.

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
name	<String>	Read-write	Name of the service to which to bind a policy or monitor. Minimum length = 1
__count	<Double>	Read-write	count parameter

Operations

(click to see Properties)

ADD:| DELETE:| GET| GET (ALL)| COUNT

Some options that you can use for each operations:

- **Getting warnings in response:**NITRO allows you to get warnings in an operation by specifying the "warning" query parameter as "yes". For example, to get warnings while connecting to the Citrix ADC appliance, the URL is as follows:

  http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/login?warning=yes

  If any, the warnings are displayed in the response payload with the HTTP code "209 X-NITRO-WARNING".

- **Authenticated access for individual NITRO operations:**NITRO allows you to logon to the Citrix ADC appliance to perform individual operations. You can use this option instead of creating a NITRO session (using the login object) and then using that session to perform all operations,

  To do this, you must specify the username and password in the request header of the NITRO request as follows:

  X-NITRO-USER:<username>

  X-NITRO-PASS:<password>

  Note: In such cases, make sure that the request header DOES not include the following:

  Cookie:NITRO_AUTH_TOKEN=<tokenvalue>

`*Note:`*

Mandatory parameters are marked in redand placeholder content is marked in <green>.

add

URL:http://<Citrix-ADC-IP-address(NSIP)/nitro/v1/config/service_scpolicy_binding

HTTP Method:PUT

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Content-Type:application/json

Request Payload:

```
{

"service_scpolicy_binding":{

<b>"name":<String_value>,

</b>"policyname":<String_value>

}}
```

Response:

HTTP Status Code on Success: 201 Created HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error

delete

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/service_scpolicy_binding/name_value<String>

Query-parameters:

args

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/service_scpolicy_binding/name_value<String>?**args=policyname:<String_value>**

HTTP Method:DELETE

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue>

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error

get

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/service_scpolicy_binding/name_value<String>

Query-parameters:

filter

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/service_scpolicy_binding/name_value<String>?**filter=property-name1:property-value1,property-name2:property-value2**

Use this query-parameter to get the filtered set of service_scpolicy_binding resources configured on Citrix ADC. Filtering can be done on any of the properties of the resource.

pagination

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/service_scpolicy_binding/name_value<String>?**pagesize=#no;pageno=#no**

Use this query-parameter to get the service_scpolicy_binding resources in chunks.

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "service_scpolicy_binding": [ {

"policyname":<String_value>,

"name":<String_value>

}]}
```

get (all)

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/service_scpolicy_binding

Query-parameters:

bulkbindings

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/service_scpolicy_binding?**bulkbindings=yes**

NITRO allows you to fetch bindings in bulk.

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "service_scpolicy_binding": [ {
```

```
"policyname":<String_value>,

"name":<String_value>

}]}
```

count

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/service_scpolicy_binding/name_value<String>?**count=yes**

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{"service_scpolicy_binding": [ { "__count": "#no"} ] }
```

svcbindings

Configuration for service bindings resource.

Properties

(click to see Operations)

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
servicename	<String>	Read-write	The name of the service. Minimum length = 1
ipaddress	<String>	Read-only	The IP address of the vserver.
port	<Integer>	Read-only	The port of the vserver. Range 1 - 65535 * in CLI is represented as 65535 in NITRO API
svrstate	<String>	Read-only	The state of the vserver. Possible values = UP, DOWN, UNKNOWN, BUSY, OUT OF SERVICE, GOING OUT OF SERVICE, DOWN WHEN GOING OUT OF SERVICE, NS_EMPTY_STR, Unknown, DISABLED
vservername	<String>	Read-only	The name of the vserver.
__count	<Double>	Read-only	count parameter

Operations

(click to see Properties)

GET (ALL)| GET| COUNT

Some options that you can use for each operations:

- **Getting warnings in response:**NITRO allows you to get warnings in an operation by specifying the "warning" query parameter as "yes". For example, to get warnings while connecting to the Citrix ADC appliance, the URL is as follows:

  http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/login?warning=yes

  If any, the warnings are displayed in the response payload with the HTTP code "209 X-NITRO-WARNING".

- **Authenticated access for individual NITRO operations:**NITRO allows you to logon to the Citrix ADC appliance to perform individual operations. You can use this option instead of creating a NITRO session (using the login object) and then using that session to perform all operations,

  To do this, you must specify the username and password in the request header of the NITRO request as follows:

  X-NITRO-USER:<username>

  X-NITRO-PASS:<password>

  Note: In such cases, make sure that the request header DOES not include the following:

  Cookie:NITRO_AUTH_TOKEN=<tokenvalue>

`*Note:`*

Mandatory parameters are marked in redand placeholder content is marked in <green>.

get (all)

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/svcbindings

Query-parameters:

args

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/svcbindings?
args=servicename:<String_value>,

Use this query-parameter to get svcbindings resources based on additional properties.

attrs

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/svcbindings?**attrs=property-name1,property-name2**

Use this query parameter to specify the resource details that you want to retrieve.

filter

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/svcbindings?**filter=property-name1:property-val1,property-name2:property-val2**

Use this query-parameter to get the filtered set of svcbindings resources configured on Citrix ADC. Filtering can be done on any of the properties of the resource.

view

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/svcbindings?**view=summary**

Note: By default, the retrieved results are displayed in detail view (?view=detail).

pagination

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/svcbindings?**pagesize=#no;pageno=#no**

Use this query-parameter to get the svcbindings resources in chunks.

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "svcbindings": [ {

<b>servicename:<String_value>,</b>"ipaddress":<String_value>,

"port":<Integer_value>,

"svrstate":<String_value>,

"vservername":<String_value>

}]}
```

get

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/svcbindings/servicename_value<String>

Query-parameters:

attrs

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/svcbindings/servicename_value<String>?**attrs=property-name1,property-name2**

Use this query parameter to specify the resource details that you want to retrieve.

view

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/svcbindings/servicename_value<String>?**view=summary**

Note: By default, the retrieved results are displayed in detail view (?view=detail).

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "svcbindings": [ {

<b>servicename:<String_value>,</b>"ipaddress":<String_value>,

"port":<Integer_value>,

"svrstate":<String_value>,

"vservername":<String_value>

}]}
```

count

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/svcbindings?**args=servicename:<String_value>,;count=yes**

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "svcbindings": [ { "__count": "#no"} ] }
```

vserver

Configuration for virtual server resource.

Properties

(click to see Operations)

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
name	<String>	Read-write	The name of the virtual server to be removed. Minimum length = 1
backupvserver	<String>	Read-write	The name of the backup virtual server for this virtual server. Minimum length = 1
redirecturl	<String>	Read-write	The URL where traffic is redirected if the virtual server in the system becomes unavailable. Minimum length = 1
cacheable	<String>	Read-write	Use this option to specify whether a virtual server (used for load balancing or content switching) routes requests to the cache redirection virtual server before sending it to the configured servers.

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
			Possible values = YES, NO
clttimeout	<Double>	Read-write	The timeout value in seconds for idle client connection. Minimum value = 0 Maximum value = 31536000
somethod	<String>	Read-write	The spillover factor. The system will use this value to determine if it should send traffic to the backupvserver when the main virtual server reaches the spillover threshold. Possible values = CONNECTION, DYNAMICCONNECTION, BANDWIDTH, HEALTH, NONE
sopersistence	<String>	Read-write	The state of the spillover persistence. Default value: DISABLED Possible values = ENABLED, DISABLED
sopersistencetimeout	<Double>	Read-write	The spillover persistence entry timeout.

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
			Default value: 2 Minimum value = 2 Maximum value = 1440
sothreshold	<Double>	Read-write	The spillver threshold value. Minimum value = 1 Maximum value = 4294967294
pushvserver	<String>	Read-write	The lb vserver of type PUSH/ SSL_PUSH to which server pushes the updates received on the client facing non-push lb vserver. Minimum length = 1

Operations

(click to see Properties)

DELETE| UPDATE| ENABLE| DISABLE

Some options that you can use for each operations:

- **Getting warnings in response:**NITRO allows you to get warnings in an operation by specifying the "warning" query parameter as "yes". For example, to get warnings while connecting to the Citrix ADC appliance, the URL is as follows:

  http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/login?warning=yes

  If any, the warnings are displayed in the response payload with the HTTP code "209 X-NITRO-WARNING".

- **Authenticated access for individual NITRO operations:**NITRO allows you to logon to the Citrix ADC appliance to perform individual operations. You can use this option instead of creating a NITRO session (using the login object) and then using that session to perform all operations,

  To do this, you must specify the username and password in the request header of the NITRO request as follows:

  X-NITRO-USER:<username>

  X-NITRO-PASS:<password>

  Note: In such cases, make sure that the request header DOES not include the following:

  Cookie:NITRO_AUTH_TOKEN=<tokenvalue>

`*Note:`*

Mandatory parameters are marked in redand placeholder content is marked in <green>.

delete

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/vserver/name_value<String>

HTTP Method:DELETE

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue>

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error

update

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/vserver

HTTP Method:PUT

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Content-Type:application/json

Request Payload:

```
{"vserver":{

<b>"name":<String_value>,

</b>"backupvserver":<String_value>,

"redirecturl":<String_value>,

"cacheable":<String_value>,

"clttimeout":<Double_value>,

"somethod":<String_value>,

"sopersistence":<String_value>,

"sopersistencetimeout":<Double_value>,

"sothreshold":<Double_value>,

"pushvserver":<String_value>

}}
```

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error

enable

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/vserver?**action=enable**

HTTP Method:POST

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Content-Type:application/json

Request Payload:

```
{"vserver":{

<b>"name":<String_value>

</b>}}
```

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error

disable

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/vserver?**action=disable**

HTTP Method:POST

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Content-Type:application/json

Request Payload:

```
{"vserver":{

<b>"name":<String_value>

</b>}}
```

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error

BFD

Bfd

Bidirectional Forwarding Detection

NAME	DESCRIPTION
bfdsession	Configuration for BFD configuration

bfdsession

Configuration for BFD configuration resource.

Properties

(click to see Operations)

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
localip	<String>	Read-write	IPV4 or IPV6 Address of Local Node.
remoteip	<String>	Read-write	IPV4 or IPV6 Address of Remote Node.
state	<String>	Read-only	Current state of the BFD session. Possible values = ADMIN DOWN, DOWN, INIT, UP
localport	<Double>	Read-only	Source Port used by Local node to send Control packets for the BFD session.
remoteport	<Double>	Read-only	Source Port used by Remote node to send Control packets for the BFD session.

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
minimumtransmitinterval	<Double>	Read-only	Minimum trasmit interval, in milliseconds, the local node would like to use when transmitting BFD Control packets.
negotiatedminimumtransmitinterval	<Double>	Read-only	Negotiated Minimum Transmit Interval. This is the interval at which the local node will be sending out BFD control packets.
minimumreceiveinterval	<Double>	Read-only	Minimum receive interval, in milliseconds, between received BFD Control packets that the local node is capable of supporting.
negotiatedminimumreceiveinterval	<Double>	Read-only	Negotiated Minimum Receive Interval. This is the interval at which the local node will be expecting BFD control packets.
multiplier	<Double>	Read-only	Detection Multiplier. The negotiated transmit interval multiplied by Detection multiplier provides the Detection Time for the remote node for the BFD session.

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
remotemultiplier	<Double>	Read-only	Your Multiplier. The negotiated receive interval multiplied by Your Multiplier provides the Detection Time for the local node for the BFD session.
vlan	<Double>	Read-only	VLAN ID on which the BDS session is configured.
localdiagnotic	<Double>	Read-only	Diagnostic Code specifying the local system's reason for the last change in session state.
localdiscriminator	<Double>	Read-only	A unique discriminator value generated by the local node for the session.
remotediscriminator	<Double>	Read-only	A unique discriminator value as received from the remote node for the session.
passive	<Boolean>	Read-only	Flag indicating that the session is passive.
multihop	<Boolean>	Read-only	Flag indicating if the session is multihop.
admindown	<Boolean>	Read-only	Flag indicating if admin down is being sent.

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
originalownerpe	<Double>	Read-only	Original Owner PE of the BFD session.
currentownerpe	<Double>	Read-only	Current Owner PE of the BFD session.
ownernode	<Double>	Read-only	The owner node in a Cluster for this BFD session. Owner node can vary from 0 to 31. If ownernode is not specified then the session is treated as Striped session. Default value: 255
__count	<Double>	Read-only	count parameter

Operations

(click to see Properties)

GET (ALL)| COUNT

Some options that you can use for each operations:

- **Getting warnings in response:**NITRO allows you to get warnings in an operation by specifying the "warning" query parameter as "yes". For example, to get warnings while connecting to the Citrix ADC appliance, the URL is as follows:

  http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/login?warning=yes

  If any, the warnings are displayed in the response payload with the HTTP code "209 X-NITRO-WARNING".

- **Authenticated access for individual NITRO operations:**NITRO allows you to logon to the Citrix ADC appliance to perform individual operations. You can use this option instead of creating a NITRO session (using the login object) and then using that session to perform all operations,

  To do this, you must specify the username and password in the request header of the NITRO request as follows:

X-NITRO-USER:<username>

X-NITRO-PASS:<password>

Note: In such cases, make sure that the request header DOES not include the following:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue>

`*Note:`*

Mandatory parameters are marked in redand placeholder content is marked in <green>.

get (all)

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/bfdsession

Query-parameters:

args

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/bfdsession?**args=localip:<String_value>,remoteip:<String_value>**

Use this query-parameter to get bfdsession resources based on additional properties.

attrs

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/bfdsession?**attrs=property-name1,property-name2**

Use this query parameter to specify the resource details that you want to retrieve.

filter

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/bfdsession?**filter=property-name1:property-val1,property-name2:property-val2**

Use this query-parameter to get the filtered set of bfdsession resources configured on Citrix ADC. Filtering can be done on any of the properties of the resource.

view

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/bfdsession?**view=summary**

Note: By default, the retrieved results are displayed in detail view (?view=detail).

pagination

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/bfdsession?**pagesize=#no;pageno=#no**

Use this query-parameter to get the bfdsession resources in chunks.

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "bfdsession": [ {

localip:<String_value>,remoteip:<String_value>"state":<String_value>,

"localport":<Double_value>,

"remoteport":<Double_value>,

"minimumtransmitinterval":<Double_value>,

"negotiatedminimumtransmitinterval":<Double_value>,

"minimumreceiveinterval":<Double_value>,

"negotiatedminimumreceiveinterval":<Double_value>,

"multiplier":<Double_value>,

"remotemultiplier":<Double_value>,

"vlan":<Double_value>,
```

```
"localdiagnotic":<Double_value>,

"localdiscriminator":<Double_value>,

"remotediscriminator":<Double_value>,

"passive":<Boolean_value>,

"multihop":<Boolean_value>,

"admindown":<Boolean_value>,

"originalownerpe":<Double_value>,

"currentownerpe":<Double_value>,

"ownernode":<Double_value>

}]}
```

count

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/bfdsession?**count=yes**

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "bfdsession": [ { "__count": "#no"} ] }
```

CA

Ca

Content Accelerator Configuration. The Content Accelerator feature is used to cache static and dynamic web application data.

NAME	DESCRIPTION
caaction	Configuration for ca action
caglobal_binding	Binding object showing the resources that can be bound to caglobal
caglobal_capolicy_binding	Binding object showing the capolicy that can be bound to caglobal
capolicy	Configuration for contentaccelerator policy
capolicy_binding	Binding object showing the resources that can be bound to capolicy
capolicy_caglobal_binding	Binding object showing the caglobal that can be bound to capolicy
capolicy_csvserver_binding	Binding object showing the csvserver that can be bound to capolicy
capolicy_lbvserver_binding	Binding object showing the lbvserver that can be bound to capolicy

caaction

Configuration for ca action resource.

Properties

(click to see Operations)

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
name	<String>	Read-write	Name of the content accelerator action. Must begin with an ASCII alphabetic or underscore (_) character, and must contain only ASCII alphanumeric, underscore, hash (#), period (.), space, colon (:), at (@), equals (=), and hyphen (-) characters.
accumressize	<Double>	Read-write	Size of the data, in KB, that the server must respond with. The Citrix ADC uses this data to compute a hash which is then used to lookup within the T2100 appliance.
lbvserver	<String>	Read-write	Name of the load balancing virtual server that has the T2100 appliances as services.
comment	<String>	Read-write	Information about the content accelerator action.
type	<String>	Read-write	Specifies whether the Citrix ADC must lookup for the response on the

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
			T2100 appliance or serve the response directly from the server. Possible values = nolookup, lookup, noop
newname	<String>	Read-write	New name for the content accelerator action. Must begin with a letter, number, or the underscore character (_), and must contain only letters, numbers, and the hyphen (-), period (.) hash (#), space ( ), at (@), equals (=), colon (:), and underscore characters. Can be changed after the content accelerator policy is added. The following requirement applies only to the Citrix ADC CLI: If the name includes one or more spaces, enclose the name in double or single quotation marks (for example,

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
			"my content accelerator action" or 'my content accelerator action'). Minimum length = 1
isdefault	<Boolean>	Read-only	A value of true is returned if it is a default content accelerator action.
hits	<Double>	Read-only	The number of times the action has been taken.
__count	<Double>	Read-only	count parameter

Operations

(click to see Properties)

ADD| UPDATE| UNSET| DELETE| RENAME| GET (ALL)| GET| COUNT

Some options that you can use for each operations:

- **Getting warnings in response:**NITRO allows you to get warnings in an operation by specifying the "warning" query parameter as "yes". For example, to get warnings while connecting to the Citrix ADC appliance, the URL is as follows:

  http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/login?warning=yes

  If any, the warnings are displayed in the response payload with the HTTP code "209 X-NITRO-WARNING".

- **Authenticated access for individual NITRO operations:**NITRO allows you to logon to the Citrix ADC appliance to perform individual operations. You can use this option instead of creating a NITRO session (using the login object) and then using that session to perform all operations,

  To do this, you must specify the username and password in the request header of the NITRO request as follows:

X-NITRO-USER:<username>

X-NITRO-PASS:<password>

Note: In such cases, make sure that the request header DOES not include the following:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue>

`*Note:`*

Mandatory parameters are marked in redand placeholder content is marked in <green>.

add

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/caaction

HTTP Method:POST

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Content-Type:application/json

Request Payload:

```
{"caaction":{

<b>"name":<String_value>,

</b>"accumressize":<Double_value>,

"lbvserver":<String_value>,

"comment":<String_value>,

<b>"type":<String_value>

</b>}}
```

Response:

HTTP Status Code on Success: 201 Created HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error

update

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/caaction

HTTP Method:PUT

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Content-Type:application/json

Request Payload:

```
{"caaction":{

<b>"name":<String_value>,

</b>"accumressize":<Double_value>,

"type":<String_value>,

"lbvserver":<String_value>,

"comment":<String_value>

}}
```

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error

unset

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/caaction?**action=unset**

HTTP Method:POST

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Content-Type:application/json

Request Payload:

```
{"caaction":{

<b>"name":<String_value>,

</b>"accumressize":true,

"type":true,

"lbvserver":true,

"comment":true

}}
```

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error

delete

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/caaction/name_value<String>

HTTP Method:DELETE

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue>

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error

rename

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/caaction?**action=rename**

HTTP Method:POST

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Content-Type:application/json

Request Payload:

```
{"caaction":{

<b>"name":<String_value>,

</b><b>"newname":<String_value>

</b>}}
```

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error

get (all)

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/caaction

Query-parameters:

attrs

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/caaction?**attrs=property-name1,property-name2**

Use this query parameter to specify the resource details that you want to retrieve.

filter

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/caaction?**filter=property-name1:property-val1,property-name2:property-val2**

Use this query-parameter to get the filtered set of caaction resources configured on Citrix ADC. Filtering can be done on any of the properties of the resource.

view

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/caaction?**view=summary**

Note: By default, the retrieved results are displayed in detail view (?view=detail).

pagination

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/caaction?**pagesize=#no;pageno=#no**

Use this query-parameter to get the caaction resources in chunks.

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "caaction": [ {

"name":<String_value>,

"type":<String_value>,

"accumressize":<Double_value>,

"lbvserver":<String_value>,
```

```
"isdefault":<Boolean_value>,

"hits":<Double_value>,

"comment":<String_value>

}]}
```

get

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/caaction/name_value<String>

Query-parameters:

attrs

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/caaction/name_value<String>?**attrs=property-name1,property-name2**

Use this query parameter to specify the resource details that you want to retrieve.

view

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/caaction/name_value<String>?**view=summary**

Note: By default, the retrieved results are displayed in detail view (?view=detail).

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "caaction": [ {

"name":<String_value>,

"type":<String_value>,

"accumressize":<Double_value>,

"lbvserver":<String_value>,

"isdefault":<Boolean_value>,

"hits":<Double_value>,

"comment":<String_value>

}]}
```

count

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/caaction?**count=yes**

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "caaction": [ { "__count": "#no"} ] }
```

caglobal_binding

Binding object which returns the resources bound to caglobal.

Properties

(click to see Operations)

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
caglobal_capolicy_binding	<caglobal_capolicy_binding[]>	Read-only	capolicy that can be bound to caglobal.

Operations

(click to see Properties)

GET| GET (ALL)

Some options that you can use for each operations:

- **Getting warnings in response:**NITRO allows you to get warnings in an operation by specifying the "warning" query parameter as "yes". For example, to get warnings while connecting to the Citrix ADC appliance, the URL is as follows:

  http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/login?warning=yes

  If any, the warnings are displayed in the response payload with the HTTP code "209 X-NITRO-WARNING".
- **Authenticated access for individual NITRO operations:**NITRO allows you to logon to the Citrix ADC appliance to perform individual operations. You can use this option instead of creating a NITRO session (using the login object) and then using that session to perform all operations,

  To do this, you must specify the username and password in the request header of the NITRO request as follows:

X-NITRO-USER:<username>

X-NITRO-PASS:<password>

Note: In such cases, make sure that the request header DOES not include the following:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue>

`*Note:`*

Mandatory parameters are marked in redand placeholder content is marked in <green>.

get

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/caglobal_binding

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "caglobal_binding": [ {

"caglobal_capolicy_binding":<caglobal_capolicy_binding[]_value>

}]}
```

get (all)

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/caglobal_binding

Query-parameters:

bulkbindings

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/caglobal_binding?**bulkbindings=yes**

NITRO allows you to fetch bindings in bulk.

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "caglobal_binding": [ {

"caglobal_capolicy_binding":<caglobal_capolicy_binding[]_value>

}]}
```

caglobal_capolicy_binding

Binding object showing the capolicy that can be bound to caglobal.

Properties

(click to see Operations)

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
priority	<Double>	Read-write	Specifies the priority of the content accelerator policy.

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
globalbindtype	<String>	Read-write	. Default value: SYSTEM_GLOBAL Possible values = SYSTEM_GLOBAL, VPN_GLOBAL, RNAT_GLOBAL
policyname	<String>	Read-write	Name of the content accelerator policy.
gotopriorityexpression	<String>	Read-write	.
type	<String>	Read-write	. Possible values = REQ_OVERRIDE, REQ_DEFAULT, RES_OVERRIDE, RES_DEFAULT
flowtype	<Double>	Read-only	flowtype of the bound content accelerator policy.
numpol	<Double>	Read-only	Number of polices bound to label.
__count	<Double>	Read-only	count parameter

Operations

(click to see Properties)

ADD:| DELETE:| GET| GET (ALL)| COUNT

Some options that you can use for each operations:

- **Getting warnings in response:**NITRO allows you to get warnings in an operation by specifying the "warning" query parameter as "yes". For example, to get warnings while connecting to the Citrix ADC appliance, the URL is as follows:

  http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/login?warning=yes

  If any, the warnings are displayed in the response payload with the HTTP code "209 X-NITRO-WARNING".

- **Authenticated access for individual NITRO operations:**NITRO allows you to logon to the Citrix ADC appliance to perform individual operations. You can use this option instead of creating a NITRO session (using the login object) and then using that session to perform all operations,

  To do this, you must specify the username and password in the request header of the NITRO request as follows:

  X-NITRO-USER:<username>

  X-NITRO-PASS:<password>

  Note: In such cases, make sure that the request header DOES not include the following:

  Cookie:NITRO_AUTH_TOKEN=<tokenvalue>

`*Note:`*

Mandatory parameters are marked in redand placeholder content is marked in <green>.

add

URL:http://<Citrix-ADC-IP-address(NSIP)/nitro/v1/config/caglobal_capolicy_binding

HTTP Method:PUT

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Content-Type:application/json

Request Payload:

```
{
```

```
"caglobal_capolicy_binding":{

<b>"policyname":<String_value>,

</b><b>"priority":<Double_value>,

</b>"gotopriorityexpression":<String_value>,

"type":<String_value>

}}
```

Response:

HTTP Status Code on Success: 201 Created HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error

delete

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/caglobal_capolicy_binding

Query-parameters:

args

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/caglobal_capolicy_binding?**args=policyname:<String_value>,type:<String_value>,priority:<Double_value>**

HTTP Method:DELETE

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue>

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error

get

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/caglobal_capolicy_binding

Query-parameters:

filter

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/caglobal_capolicy_binding?**filter=property-name1:property-value1,property-name2:property-value2**

Use this query-parameter to get the filtered set of caglobal_capolicy_binding resources configured on Citrix ADC. Filtering can be done on any of the properties of the resource.

pagination

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/caglobal_capolicy_binding?**pagesize=#no;pageno=#no**

Use this query-parameter to get the caglobal_capolicy_binding resources in chunks.

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "caglobal_capolicy_binding": [ {

"priority":<Double_value>,

"globalbindtype":<String_value>,
```

```
"policyname":<String_value>,

"gotopriorityexpression":<String_value>,

"type":<String_value>,

"flowtype":<Double_value>,

"numpol":<Double_value>

}]}
```

get (all)

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/caglobal_capolicy_binding

Query-parameters:

bulkbindings

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/caglobal_capolicy_binding?**bulkbindings=yes**

NITRO allows you to fetch bindings in bulk.

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "caglobal_capolicy_binding": [ {

"priority":<Double_value>,

"globalbindtype":<String_value>,

"policyname":<String_value>,

"gotopriorityexpression":<String_value>,

"type":<String_value>,

"flowtype":<Double_value>,

"numpol":<Double_value>

}]}
```

count

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/caglobal_capolicy_binding?**count=yes**

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "caglobal_capolicy_binding": [ { "__count": "#no"} ] }
```

capolicy

Configuration for contentaccelerator policy resource.

Properties

(click to see Operations)

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
name	<String>	Read-write	Name for the content accelerator policy. Must begin with an ASCII alphabetic or underscore (_) character, and must contain only ASCII alphanumeric, underscore, hash (#), period (.), space, colon (:), at (@), equals (=), and hyphen (-) characters. Can be changed after the policy is created.
rule	<String>	Read-write	Expression that determines which requests or responses match the content accelerator policy. When specifying the rule in the CLI, the description must be

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
			enclosed within double quotes.
action	<String>	Read-write	Name of content accelerator action to be executed when the rule is evaluated to true.
undefaction	<String>	Read-write	.
comment	<String>	Read-write	Information about the content accelerator policy.
logaction	<String>	Read-write	Name of messagelog action to use when a request matches this policy.
newname	<String>	Read-write	New name for the content accelerator policy. Minimum length = 1
hits	<Double>	Read-only	Number of hits.
undefhits	<Double>	Read-only	Number of Undef hits.
isdefault	<Boolean>	Read-only	A value of true is returned if it is a default content accelerator policy.
__count	<Double>	Read-only	count parameter

Operations

(click to see Properties)

ADD| DELETE| UPDATE| UNSET| RENAME| GET (ALL)| GET| COUNT

Some options that you can use for each operations:

- **Getting warnings in response:**NITRO allows you to get warnings in an operation by specifying the "warning" query parameter as "yes". For example, to get warnings while connecting to the Citrix ADC appliance, the URL is as follows:

  http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/login?warning=yes

  If any, the warnings are displayed in the response payload with the HTTP code "209 X-NITRO-WARNING".

- **Authenticated access for individual NITRO operations:**NITRO allows you to logon to the Citrix ADC appliance to perform individual operations. You can use this option instead of creating a NITRO session (using the login object) and then using that session to perform all operations,

  To do this, you must specify the username and password in the request header of the NITRO request as follows:

  X-NITRO-USER:<username>

  X-NITRO-PASS:<password>

  Note: In such cases, make sure that the request header DOES not include the following:

  Cookie:NITRO_AUTH_TOKEN=<tokenvalue>

`*Note:`*

Mandatory parameters are marked in redand placeholder content is marked in <green>.

add

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/capolicy

HTTP Method:POST

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Content-Type:application/json

Request Payload:

```
{"capolicy":{

<b>"name":<String_value>,

</b><b>"rule":<String_value>,

</b><b>"action":<String_value>,

</b>"undefaction":<String_value>,

"comment":<String_value>,

"logaction":<String_value>

}}
```

Response:

HTTP Status Code on Success: 201 Created HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error

delete

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/capolicy/name_value<String>

HTTP Method:DELETE

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue>

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error

update

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/capolicy

HTTP Method:PUT

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Content-Type:application/json

Request Payload:

```
{"capolicy":{

<b>"name":<String_value>,

</b>"rule":<String_value>,

"action":<String_value>,

"comment":<String_value>,

"logaction":<String_value>,

"undefaction":<String_value>

}}
```

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error

unset

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/capolicy?**action=unset**

HTTP Method:POST

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Content-Type:application/json

Request Payload:

```
{"capolicy":{

<b>"name":<String_value>,

</b>"comment":true,

"logaction":true,

"undefaction":true

}}
```

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error

rename

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/capolicy?**action=rename**

HTTP Method:POST

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Content-Type:application/json

Request Payload:

```
{"capolicy":{

<b>"name":<String_value>,

</b><b>"newname":<String_value>

</b>}}
```

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error

get (all)

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/capolicy

Query-parameters:

attrs

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/capolicy?**attrs=property-name1,property-name2**

Use this query parameter to specify the resource details that you want to retrieve.

filter

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/capolicy?**filter=property-name1:property-val1,property-name2:property-val2**

Use this query-parameter to get the filtered set of capolicy resources configured on Citrix ADC. Filtering can be done on any of the properties of the resource.

view

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/capolicy?**view=summary**

Note: By default, the retrieved results are displayed in detail view (?view=detail).

pagination

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/capolicy?**pagesize=#no;pageno=#no**

Use this query-parameter to get the capolicy resources in chunks.

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "capolicy": [ {

"name":<String_value>,

"undefaction":<String_value>,

"action":<String_value>,

"rule":<String_value>,

"hits":<Double_value>,

"undefhits":<Double_value>,

"isdefault":<Boolean_value>,

"comment":<String_value>,

"logaction":<String_value>

}]}
```

get

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/capolicy/name_value<String>

Query-parameters:

attrs

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/capolicy/name_value<String>?**attrs=property-name1,property-name2**

Use this query parameter to specify the resource details that you want to retrieve.

view

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/capolicy/name_value<String>?**view=summary**

Note: By default, the retrieved results are displayed in detail view (?view=detail).

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "capolicy": [ {

"name":<String_value>,

"undefaction":<String_value>,

"action":<String_value>,

"rule":<String_value>,

"hits":<Double_value>,

"undefhits":<Double_value>,
```

```
"isdefault":<Boolean_value>,

"comment":<String_value>,

"logaction":<String_value>

}]}
```

count

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/capolicy?**count=yes**

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "capolicy": [ { "__count": "#no"} ] }
```

capolicy_binding

Binding object which returns the resources bound to capolicy.

Properties

(click to see Operations)

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
name	<String>	Read-write	Name of the content accelerator policy to be displayed.
capolicy_lbvserver_binding	<capolicy_lbvserver_binding[]>	Read-only	lbvserver that can be bound to capolicy.
capolicy_caglobal_binding	<capolicy_caglobal_binding[]>	Read-only	caglobal that can be bound to capolicy.
capolicy_csvserver_binding	<capolicy_csvserver_binding[]>	Read-only	csvserver that can be bound to capolicy.

Operations

(click to see Properties)

GET| GET (ALL)

Some options that you can use for each operations:

- **Getting warnings in response:**NITRO allows you to get warnings in an operation by specifying the "warning" query parameter as "yes". For example, to get warnings while connecting to the Citrix ADC appliance, the URL is as follows:

  http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/login?warning=yes

  If any, the warnings are displayed in the response payload with the HTTP code "209 X-NITRO-WARNING".

- **Authenticated access for individual NITRO operations:**NITRO allows you to logon to the Citrix ADC appliance to perform individual operations. You can use this option instead of creating a NITRO session (using the login object) and then using that session to perform all operations,

  To do this, you must specify the username and password in the request header of the NITRO request as follows:

  X-NITRO-USER:<username>

  X-NITRO-PASS:<password>

Note: In such cases, make sure that the request header DOES not include the following:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue>

`*Note:` *

Mandatory parameters are marked in redand placeholder content is marked in <green>.

get

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/capolicy_binding/name_value<String>

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "capolicy_binding": [ {

"name":<String_value>,

"capolicy_csvserver_binding":<capolicy_csvserver_binding[]_value>,

"capolicy_caglobal_binding":<capolicy_caglobal_binding[]_value>,

"capolicy_lbvserver_binding":<capolicy_lbvserver_binding[]_value>

}]}
```

get (all)

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/capolicy_binding

Query-parameters:

bulkbindings

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/capolicy_binding?**bulkbindings=yes**

NITRO allows you to fetch bindings in bulk.

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "capolicy_binding": [ {

"name":<String_value>,

"capolicy_csvserver_binding":<capolicy_csvserver_binding[]_value>,

"capolicy_caglobal_binding":<capolicy_caglobal_binding[]_value>,

"capolicy_lbvserver_binding":<capolicy_lbvserver_binding[]_value>

}]}
```

capolicy_caglobal_binding

Binding object showing the caglobal that can be bound to capolicy.

Properties

(click to see Operations)

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
name	<String>	Read-write	Name of the content accelerator policy to be displayed.
boundto	<String>	Read-write	Location where policy is bound.
priority	<Double>	Read-only	Specifies the priority of the policy.
gotopriorityexpression	<String>	Read-only	Expression specifying the priority of the next policy which will get evaluated if the current policy rule evaluates to TRUE.
activepolicy	<Integer>	Read-only	Indicates whether policy is bound or not.
__count	<Double>	Read-only	count parameter

Operations

(click to see Properties)

GET| GET (ALL)| COUNT

Some options that you can use for each operations:

- **Getting warnings in response:**NITRO allows you to get warnings in an operation by specifying the "warning" query parameter as "yes". For example, to get warnings while connecting to the Citrix ADC appliance, the URL is as follows:

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/login?warning=yes

If any, the warnings are displayed in the response payload with the HTTP code "209 X-NITRO-WARNING".

- **Authenticated access for individual NITRO operations:**NITRO allows you to logon to the Citrix ADC appliance to perform individual operations. You can use this option instead of creating a NITRO session (using the login object) and then using that session to perform all operations,

  To do this, you must specify the username and password in the request header of the NITRO request as follows:

  X-NITRO-USER:<username>

  X-NITRO-PASS:<password>

  Note: In such cases, make sure that the request header DOES not include the following:

  Cookie:NITRO_AUTH_TOKEN=<tokenvalue>

`*Note:`*

Mandatory parameters are marked in redand placeholder content is marked in <green>.

get

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/capolicy_caglobal_binding/name_value<String>

Query-parameters:

filter

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/capolicy_caglobal_binding/name_value<String>?**filter=property-name1:property-value1,property-name2:property-value2**

Use this query-parameter to get the filtered set of capolicy_caglobal_binding resources configured on Citrix ADC. Filtering can be done on any of the properties of the resource.

pagination

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/capolicy_caglobal_binding/name_value<String>?**pagesize=#no;pageno=#no**

Use this query-parameter to get the capolicy_caglobal_binding resources in chunks.

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "capolicy_caglobal_binding": [ {

"name":<String_value>,

"boundto":<String_value>,

"priority":<Double_value>,

"gotopriorityexpression":<String_value>,

"activepolicy":<Integer_value>

}]}
```

get (all)

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/capolicy_caglobal_binding

Query-parameters:

bulkbindings

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/capolicy_caglobal_binding?**bulkbindings=yes**

NITRO allows you to fetch bindings in bulk.

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "capolicy_caglobal_binding": [ {

"name":<String_value>,

"boundto":<String_value>,

"priority":<Double_value>,

"gotopriorityexpression":<String_value>,

"activepolicy":<Integer_value>

}]}
```

count

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/capolicy_caglobal_binding/name_value<String>?**count=yes**

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{"capolicy_caglobal_binding": [ { "__count": "#no"} ] }
```

capolicy_csvserver_binding

Binding object showing the csvserver that can be bound to capolicy.

Properties

(click to see Operations)

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
name	<String>	Read-write	Name of the content accelerator policy to be displayed.
boundto	<String>	Read-write	Location where policy is bound.
priority	<Double>	Read-only	Specifies the priority of the policy.
gotopriorityexpression	<String>	Read-only	Expression specifying the priority of the next policy which will get evaluated if the

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
			current policy rule evaluates to TRUE.
activepolicy	<Integer>	Read-only	Indicates whether policy is bound or not.
__count	<Double>	Read-only	count parameter

Operations

(click to see Properties)

GET| GET (ALL)| COUNT

Some options that you can use for each operations:

- **Getting warnings in response:**NITRO allows you to get warnings in an operation by specifying the "warning" query parameter as "yes". For example, to get warnings while connecting to the Citrix ADC appliance, the URL is as follows:

  http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/login?warning=yes

  If any, the warnings are displayed in the response payload with the HTTP code "209 X-NITRO-WARNING".

- **Authenticated access for individual NITRO operations:**NITRO allows you to logon to the Citrix ADC appliance to perform individual operations. You can use this option instead of creating a NITRO session (using the login object) and then using that session to perform all operations,

  To do this, you must specify the username and password in the request header of the NITRO request as follows:

  X-NITRO-USER:<username>

  X-NITRO-PASS:<password>

  Note: In such cases, make sure that the request header DOES not include the following:

  Cookie:NITRO_AUTH_TOKEN=<tokenvalue>

`*Note:` *

Mandatory parameters are marked in redand placeholder content is marked in <green>.

get

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/capolicy_csvserver_binding/name_value<String>

Query-parameters:

filter

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/capolicy_csvserver_binding/name_value<String>?**filter=property-name1:property-value1,property-name2:property-value2**

Use this query-parameter to get the filtered set of capolicy_csvserver_binding resources configured on Citrix ADC. Filtering can be done on any of the properties of the resource.

pagination

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/capolicy_csvserver_binding/name_value<String>?**pagesize=#no;pageno=#no**

Use this query-parameter to get the capolicy_csvserver_binding resources in chunks.

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "capolicy_csvserver_binding": [ {

"name":<String_value>,
```

```
"boundto":<String_value>,

"priority":<Double_value>,

"gotopriorityexpression":<String_value>,

"activepolicy":<Integer_value>

}]}
```

get (all)

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/capolicy_csvserver_binding

Query-parameters:

bulkbindings

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/capolicy_csvserver_binding?**bulkbindings=yes**

NITRO allows you to fetch bindings in bulk.

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "capolicy_csvserver_binding": [ {
```

```
"name":<String_value>,

"boundto":<String_value>,

"priority":<Double_value>,

"gotopriorityexpression":<String_value>,

"activepolicy":<Integer_value>

}]}
```

count

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/capolicy_csvserver_binding/name_value<String>?**count=yes**

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{"capolicy_csvserver_binding": [ { "__count": "#no"} ] }
```

capolicy_lbvserver_binding

Binding object showing the lbvserver that can be bound to capolicy.

Properties

(click to see Operations)

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
name	<String>	Read-write	Name of the content accelerator policy to be displayed.
boundto	<String>	Read-write	Location where policy is bound.
priority	<Double>	Read-only	Specifies the priority of the policy.
gotopriorityexpression	<String>	Read-only	Expression specifying the priority of the next policy which will get evaluated if the current policy rule evaluates to TRUE.
activepolicy	<Integer>	Read-only	Indicates whether policy is bound or not.
__count	<Double>	Read-only	count parameter

Operations

(click to see Properties)

GET| GET (ALL)| COUNT

Some options that you can use for each operations:

- **Getting warnings in response:**NITRO allows you to get warnings in an operation by specifying the "warning" query parameter as "yes". For example, to get warnings while connecting to the Citrix ADC appliance, the URL is as follows:

  http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/login?warning=yes

If any, the warnings are displayed in the response payload with the HTTP code "209 X-NITRO-WARNING".

- **Authenticated access for individual NITRO operations:**NITRO allows you to logon to the Citrix ADC appliance to perform individual operations. You can use this option instead of creating a NITRO session (using the login object) and then using that session to perform all operations,

  To do this, you must specify the username and password in the request header of the NITRO request as follows:

  X-NITRO-USER:<username>

  X-NITRO-PASS:<password>

  Note: In such cases, make sure that the request header DOES not include the following:

  Cookie:NITRO_AUTH_TOKEN=<tokenvalue>

`*Note:`*

Mandatory parameters are marked in redand placeholder content is marked in <green>.

get

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/capolicy_lbvserver_binding/name_value<String>

Query-parameters:

filter

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/capolicy_lbvserver_binding/name_value<String>?**filter=property-name1:property-value1,property-name2:property-value2**

Use this query-parameter to get the filtered set of capolicy_lbvserver_binding resources configured on Citrix ADC. Filtering can be done on any of the properties of the resource.

pagination

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/capolicy_lbvserver_binding/name_value<String>?**pagesize=#no;pageno=#no**

Use this query-parameter to get the capolicy_lbvserver_binding resources in chunks.

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "capolicy_lbvserver_binding": [ {

"name":<String_value>,

"boundto":<String_value>,

"priority":<Double_value>,

"gotopriorityexpression":<String_value>,

"activepolicy":<Integer_value>

}]}
```

get (all)

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/capolicy_lbvserver_binding

Query-parameters:

bulkbindings

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/capolicy_lbvserver_binding?**bulkbindings=yes**

NITRO allows you to fetch bindings in bulk.

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "capolicy_lbvserver_binding": [ {

"name":<String_value>,

"boundto":<String_value>,

"priority":<Double_value>,

"gotopriorityexpression":<String_value>,

"activepolicy":<Integer_value>

}]}
```

count

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/capolicy_lbvserver_binding/name_value<String>?**count=yes**

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{"capolicy_lbvserver_binding": [ { "__count": "#no"} ] }
```

Cache Redirection

Cache Redirection

Cache redirection configuration. The cache redirection feature can transparently redirect cacheable HTTP requests to a cache and send non-cacheable or dynamic HTTP requests to the origin server. A cache stores or caches frequently requested web content and serves such web content to a client on behalf of the origin servers, alleviating the load on the origin server farm.

NAME	DESCRIPTION
craction	Configuration for cache redirection action
crpolicy	Configuration for cache redirection policy
crpolicy_binding	Binding object showing the resources that can be bound to crpolicy
crpolicy_crvserver_binding	Binding object showing the crvserver that can be bound to crpolicy
crvserver	Configuration for CR virtual server
crvserver_appflowpolicy_binding	Binding object showing the appflowpolicy that can be bound to crvserver

NAME	DESCRIPTION
crvserver_appfwpolicy_binding	Binding object showing the appfwpolicy that can be bound to crvserver
crvserver_appqoepolicy_binding	Binding object showing the appqoepolicy that can be bound to crvserver
crvserver_binding	Binding object showing the resources that can be bound to crvserver
crvserver_cachepolicy_binding	Binding object showing the cachepolicy that can be bound to crvserver
crvserver_cmppolicy_binding	Binding object showing the cmppolicy that can be bound to crvserver
crvserver_crpolicy_binding	Binding object showing the crpolicy that can be bound to crvserver
crvserver_cspolicy_binding	Binding object showing the cspolicy that can be bound to crvserver
crvserver_feopolicy_binding	Binding object showing the feopolicy that can be bound to crvserver
crvserver_filterpolicy_binding	Binding object showing the filterpolicy that can be bound to crvserver
crvserver_icapolicy_binding	Binding object showing the icapolicy that can be bound to crvserver
crvserver_lbvserver_binding	Binding object showing the lbvserver that can be bound to crvserver
crvserver_policymap_binding	Binding object showing the policymap that can be bound to crvserver
crvserver_responderpolicy_binding	Binding object showing the responderpolicy that can be bound to crvserver
crvserver_rewritepolicy_binding	Binding object showing the rewritepolicy that can be bound to crvserver

NAME	DESCRIPTION
crvserver_spilloverpolicy_binding	Binding object showing the spilloverpolicy that can be bound to crvserver

craction

Configuration for cache redirection action resource.

Properties

(click to see Operations)

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
name	<String>	Read-write	Name of the action for which to display detailed information. Minimum length = 1
crtype	<String>	Read-only	Type to decide where to redirect the requests if the cache redirection policy is hit. The valid options are as follows: *CACHE - Directs all the requests to the cache if cache redirection policy is hit. *ORIGIN - Directs all requests to the origin server if the cache redirection policy is hit.

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
			Possible values = origin, cache
isdefault	<Boolean>	Read-only	A value of true is returned if it is a default cr action.
hits	<Double>	Read-only	The number of times the action has been taken.
referencecount	<String>	Read-only	The number of references to the action.
undefhits	<Double>	Read-only	The number of times the action resulted in UNDEF.
comment	<String>	Read-only	Comment. Any type of information about this responder action.
__count	<Double>	Read-only	count parameter

Operations

(click to see Properties)

GET (ALL)| GET| COUNT

Some options that you can use for each operations:

- **Getting warnings in response:**NITRO allows you to get warnings in an operation by specifying the "warning" query parameter as "yes". For example, to get warnings while connecting to the Citrix ADC appliance, the URL is as follows:

  http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/login?warning=yes

  If any, the warnings are displayed in the response payload with the HTTP code "209 X-NITRO-WARNING".

- **Authenticated access for individual NITRO operations:**NITRO allows you to logon to the Citrix ADC appliance to perform individual operations. You can use this option instead of creating a NITRO session (using the login object) and then using that session to perform all operations,

  To do this, you must specify the username and password in the request header of the NITRO request as follows:

  X-NITRO-USER:<username>

  X-NITRO-PASS:<password>

  Note: In such cases, make sure that the request header DOES not include the following:

  Cookie:NITRO_AUTH_TOKEN=<tokenvalue>

`*Note:`*

Mandatory parameters are marked in redand placeholder content is marked in <green>.

get (all)

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/craction

Query-parameters:

attrs

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/craction?**attrs=property-name1,property-name2**

Use this query parameter to specify the resource details that you want to retrieve.

filter

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/craction?**filter=property-name1:property-val1,property-name2:property-val2**

Use this query-parameter to get the filtered set of craction resources configured on Citrix ADC. Filtering can be done on any of the properties of the resource.

view

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/craction?**view=summary**

Note: By default, the retrieved results are displayed in detail view (?view=detail).

pagination

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/craction?**pagesize=#no;pageno=#no**

Use this query-parameter to get the craction resources in chunks.

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "craction": [ {

"name":<String_value>,

"crtype":<String_value>,

"isdefault":<Boolean_value>,

"hits":<Double_value>,

"referencecount":<String_value>,

"undefhits":<Double_value>,

"comment":<String_value>

}]}
```

get

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/craction/name_value<String>

Query-parameters:

attrs

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/craction/name_value<String>?**attrs=property-name1,property-name2**

Use this query parameter to specify the resource details that you want to retrieve.

view

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/craction/name_value<String>?**view=summary**

Note: By default, the retrieved results are displayed in detail view (?view=detail).

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "craction": [ {

"name":<String_value>,

"crtype":<String_value>,

"isdefault":<Boolean_value>,
```

```
"hits":<Double_value>,

"referencecount":<String_value>,

"undefhits":<Double_value>,

"comment":<String_value>

}]}
```

count

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/craction?**count=yes**

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "craction": [ { "__count": "#no"} ] }
```

crpolicy

Configuration for cache redirection policy resource.

Properties

(click to see Operations)

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
policyname	<String>	Read-write	Name for the cache redirection policy. Must begin with an ASCII alphanumeric or underscore (_) character, and must contain only ASCII alphanumeric, underscore, hash (#), period (.), space, colon (:), at sign (@), equal sign (=), and hyphen (-) characters. Cannot be changed after the policy is created. The following requirement applies only to the Citrix ADC CLI: If the name includes one or more spaces, enclose the name in double or single quotation marks (for example, "my policy" or 'my policy'). Minimum length = 1
rule	<String>	Read-write	Expression, or name of a named expression, against which traffic is evaluated.

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
			The following requirements apply only to the Citrix ADC CLI: * If the expression includes one or more spaces, enclose the entire expression in double quotation marks. * If the expression itself includes double quotation marks, escape the quotations by using the \ character. * Alternatively, you can use single quotation marks to enclose the rule, in which case you do not have to escape the double quotation marks.
action	<String>	Read-write	Name of the built-in cache redirection action: CACHE/ ORIGIN.
logaction	<String>	Read-write	The log action associated with the cache redirection policy.

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
newname	<String>	Read-write	The new name of the content switching policy. Minimum length = 1
domain	<String>	Read-only	Domain name.
vstype	<Double>	Read-only	Virtual server type.
hits	<Double>	Read-only	Total number of hits.
bindhits	<Double>	Read-only	Total number of hits.
priority	<Double>	Read-only	priority of bound policy.
activepolicy	<Boolean>	Read-only	Indicates whether policy is bound or not.
cspolicytype	<String>	Read-only	Indicates whether policy is PI or not. (used only during display). Possible values = Classic Policy, Advanced Policy
labelname	<String>	Read-only	Name of the label invoked.
labeltype	<String>	Read-only	The invocation type. Possible values = reqvserver, resvserver, policylabel
isdefault	<Boolean>	Read-only	A value of true is returned if it is a default cr policy.

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
__count	<Double>	Read-only	count parameter

Operations

(click to see Properties)

ADD| DELETE| UPDATE| UNSET| RENAME| GET (ALL)| GET| COUNT

Some options that you can use for each operations:

- **Getting warnings in response:**NITRO allows you to get warnings in an operation by specifying the "warning" query parameter as "yes". For example, to get warnings while connecting to the Citrix ADC appliance, the URL is as follows:

  http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/login?warning=yes

  If any, the warnings are displayed in the response payload with the HTTP code "209 X-NITRO-WARNING".

- **Authenticated access for individual NITRO operations:**NITRO allows you to logon to the Citrix ADC appliance to perform individual operations. You can use this option instead of creating a NITRO session (using the login object) and then using that session to perform all operations,

  To do this, you must specify the username and password in the request header of the NITRO request as follows:

  X-NITRO-USER:<username>

  X-NITRO-PASS:<password>

  Note: In such cases, make sure that the request header DOES not include the following:

  Cookie:NITRO_AUTH_TOKEN=<tokenvalue>

`*Note:`*

Mandatory parameters are marked in redand placeholder content is marked in <green>.

add

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/crpolicy

HTTP Method:POST

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Content-Type:application/json

Request Payload:

```
{"crpolicy":{

<b>"policyname":<String_value>,

</b><b>"rule":<String_value>,

</b>"action":<String_value>,

"logaction":<String_value>

}}
```

Response:

HTTP Status Code on Success: 201 Created HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error

delete

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/crpolicy/policyname_value<String>

HTTP Method:DELETE

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue>

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error

update

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/crpolicy

HTTP Method:PUT

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Content-Type:application/json

Request Payload:

```
{"crpolicy":{

<b>"policyname":<String_value>,

</b>"rule":<String_value>,

"action":<String_value>,

"logaction":<String_value>

}}
```

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error

unset

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/crpolicy?**action=unset**

HTTP Method:POST

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Content-Type:application/json

Request Payload:

```
{"crpolicy":{

<b>"policyname":<String_value>,

</b>"logaction":true

}}
```

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error

rename

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/crpolicy?**action=rename**

HTTP Method:POST

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Content-Type:application/json

Request Payload:

```
{"crpolicy":{

<b>"policyname":<String_value>,

</b><b>"newname":<String_value>

</b>}}
```

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error

get (all)

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/crpolicy

Query-parameters:

attrs

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/crpolicy?**attrs=property-name1,property-name2**

Use this query parameter to specify the resource details that you want to retrieve.

filter

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/crpolicy?**filter=property-name1:property-val1,property-name2:property-val2**

Use this query-parameter to get the filtered set of crpolicy resources configured on Citrix ADC. Filtering can be done on any of the properties of the resource.

view

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/crpolicy?**view=summary**

Note: By default, the retrieved results are displayed in detail view (?view=detail).

pagination

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/crpolicy?**pagesize=#no;pageno=#no**

Use this query-parameter to get the crpolicy resources in chunks.

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "crpolicy": [ {

"policyname":<String_value>,

"rule":<String_value>,

"domain":<String_value>,

"action":<String_value>,

"vstype":<Double_value>,

"hits":<Double_value>,

"bindhits":<Double_value>,

"priority":<Double_value>,

"activepolicy":<Boolean_value>,

"cspolicytype":<String_value>,

"logaction":<String_value>,

"labelname":<String_value>,

"labeltype":<String_value>,

"isdefault":<Boolean_value>

}]}
```

get

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/crpolicy/policyname_value<String>

Query-parameters:

attrs

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/crpolicy/policyname_value<String>?**attrs=property-name1,property-name2**

Use this query parameter to specify the resource details that you want to retrieve.

view

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/crpolicy/policyname_value<String>?**view=summary**

Note: By default, the retrieved results are displayed in detail view (?view=detail).

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "crpolicy": [ {

"policyname":<String_value>,

"rule":<String_value>,

"domain":<String_value>,
```

```
"action":<String_value>,

"vstype":<Double_value>,

"hits":<Double_value>,

"bindhits":<Double_value>,

"priority":<Double_value>,

"activepolicy":<Boolean_value>,

"cspolicytype":<String_value>,

"logaction":<String_value>,

"labelname":<String_value>,

"labeltype":<String_value>,

"isdefault":<Boolean_value>

}]}
```

count

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/crpolicy?**count=yes**

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "crpolicy": [ { "__count": "#no"} ] }
```

crpolicy_binding

Binding object which returns the resources bound to crpolicy.

Properties

(click to see Operations)

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
policyname	<String>	Read-write	Name of the cache redirection policy to display. If this parameter is omitted, details of all the policies are displayed. Minimum length = 1
crpolicy_crvserver_binding	<crpolicy_crvserver_binding[]>	Read-only	crvserver that can be bound to crpolicy.

Operations

(click to see Properties)

GET| GET (ALL)

Some options that you can use for each operations:

- **Getting warnings in response:**NITRO allows you to get warnings in an operation by specifying the "warning" query parameter as "yes". For example, to get warnings while connecting to the Citrix ADC appliance, the URL is as follows:

  http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/login?warning=yes

  If any, the warnings are displayed in the response payload with the HTTP code "209 X-NITRO-WARNING".

- **Authenticated access for individual NITRO operations:**NITRO allows you to logon to the Citrix ADC appliance to perform individual operations. You can use this option instead of creating a NITRO session (using the login object) and then using that session to perform all operations,

  To do this, you must specify the username and password in the request header of the NITRO request as follows:

  X-NITRO-USER:<username>

  X-NITRO-PASS:<password>

  Note: In such cases, make sure that the request header DOES not include the following:

  Cookie:NITRO_AUTH_TOKEN=<tokenvalue>

`*Note:`*

Mandatory parameters are marked in redand placeholder content is marked in <green>.

get

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/crpolicy_binding/policyname_value<String>

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "crpolicy_binding": [ {
```

```
"policyname":<String_value>,

"crpolicy_crvserver_binding":<crpolicy_crvserver_binding[]_value>

}]}
```

get (all)

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/crpolicy_binding

Query-parameters:

bulkbindings

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/crpolicy_binding?**bulkbindings=yes**

NITRO allows you to fetch bindings in bulk.

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "crpolicy_binding": [ {

"policyname":<String_value>,

"crpolicy_crvserver_binding":<crpolicy_crvserver_binding[]_value>

}]}
```

crpolicy_crvserver_binding

Binding object showing the crvserver that can be bound to crpolicy.

Properties

(click to see Operations)

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
policyname	<String>	Read-write	Name of the cache redirection policy to display. If this parameter is omitted, details of all the policies are displayed. Minimum length = 1
domain	<String>	Read-write	Domain name.
priority	<Double>	Read-only	priority of bound policy.
labelname	<String>	Read-only	Name of the label invoked.
pihits	<Double>	Read-only	Total number of hits.
hits	<Double>	Read-only	Total number of hits.
gotopriorityexpression	<String>	Read-only	Expression specifying the priority of the next policy which will get evaluated if the current policy rule evaluates to TRUE.
pipolicyhits	<Double>	Read-only	bind hits for PI CR Policy.

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
labeltype	<String>	Read-only	The invocation type. Possible values = reqvserver, resvserver, policylabel
__count	<Double>	Read-only	count parameter

Operations

(click to see Properties)

GET| GET (ALL)| COUNT

Some options that you can use for each operations:

- **Getting warnings in response:**NITRO allows you to get warnings in an operation by specifying the "warning" query parameter as "yes". For example, to get warnings while connecting to the Citrix ADC appliance, the URL is as follows:

  http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/login?warning=yes

  If any, the warnings are displayed in the response payload with the HTTP code "209 X-NITRO-WARNING".

- **Authenticated access for individual NITRO operations:**NITRO allows you to logon to the Citrix ADC appliance to perform individual operations. You can use this option instead of creating a NITRO session (using the login object) and then using that session to perform all operations,

  To do this, you must specify the username and password in the request header of the NITRO request as follows:

  X-NITRO-USER:<username>

  X-NITRO-PASS:<password>

  Note: In such cases, make sure that the request header DOES not include the following:

  Cookie:NITRO_AUTH_TOKEN=<tokenvalue>

`*Note:`*

Mandatory parameters are marked in redand placeholder content is marked in <green>.

get

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/crpolicy_crvserver_binding/policyname_value<String>

Query-parameters:

filter

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/crpolicy_crvserver_binding/policyname_value<String>?**filter=property-name1:property-value1,property-name2:property-value2**

Use this query-parameter to get the filtered set of crpolicy_crvserver_binding resources configured on Citrix ADC. Filtering can be done on any of the properties of the resource.

pagination

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/crpolicy_crvserver_binding/policyname_value<String>?**pagesize=#no;pageno=#no**

Use this query-parameter to get the crpolicy_crvserver_binding resources in chunks.

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "crpolicy_crvserver_binding": [ {
```

```
"policyname":<String_value>,

"domain":<String_value>,

"priority":<Double_value>,

"labelname":<String_value>,

"pihits":<Double_value>,

"hits":<Double_value>,

"gotopriorityexpression":<String_value>,

"pipolicyhits":<Double_value>,

"labeltype":<String_value>

}]}
```

get (all)

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/crpolicy_crvserver_binding

Query-parameters:

bulkbindings

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/crpolicy_crvserver_binding?**bulkbindings=yes**

NITRO allows you to fetch bindings in bulk.

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "crpolicy_crvserver_binding": [ {

"policyname":<String_value>,

"domain":<String_value>,

"priority":<Double_value>,

"labelname":<String_value>,

"pihits":<Double_value>,

"hits":<Double_value>,

"gotopriorityexpression":<String_value>,

"pipolicyhits":<Double_value>,

"labeltype":<String_value>

}]}
```

count

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/crpolicy_crvserver_binding/policyname_value<String>?**count=yes**

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{"crpolicy_crvserver_binding": [ { "__count": "#no"} ] }
```

crvserver

Configuration for CR virtual server resource.

Properties

(click to see Operations)

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
name	<String>	Read-write	Name for the cache redirection virtual server. Must begin with an ASCII alphanumeric or underscore (_) character, and must contain only ASCII alphanumeric, underscore, hash (#), period (.), space, colon (:), at sign (@), equal sign (=), and hyphen (-) characters. Can be changed after the cache redirection

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
			virtual server is created. The following requirement applies only to the Citrix ADC CLI: If the name includes one or more spaces, enclose the name in double or single quotation marks (for example, "my server" or 'my server'). Minimum length = 1
td	<Double>	Read-write	Integer value that uniquely identifies the traffic domain in which you want to configure the entity. If you do not specify an ID, the entity becomes part of the default traffic domain, which has an ID of 0. Minimum value = 0 Maximum value = 4094
servicetype	<String>	Read-write	Protocol (type of service) handled by the virtual server.

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
			Possible values = HTTP, SSL, NNTP, HDX
ipv46	<String>	Read-write	IPv4 or IPv6 address of the cache redirection virtual server. Usually a public IP address. Clients send connection requests to this IP address. Note: For a transparent cache redirection virtual server, use an asterisk (*) to specify a wildcard virtual server address.
port	<Integer>	Read-write	Port number of the virtual server. Default value: 80 Minimum value = 1 Maximum value = 65534
ipset	<String>	Read-write	The list of IPv4/IPv6 addresses bound to ipset would form a part of listening service on the current cr vserver. Minimum length = 1

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
range	<Double>	Read-write	Number of consecutive IP addresses, starting with the address specified by the IPAddress parameter, to include in a range of addresses assigned to this virtual server. Default value: 1 Minimum value = 1 Maximum value = 254
cachetype	<String>	Read-write	Mode of operation for the cache redirection virtual server. Available settings function as follows: * TRANSPARENT - Intercept all traffic flowing to the appliance and apply cache redirection policies to determine whether content should be served from the cache or from the origin server. * FORWARD - Resolve the hostname of the incoming request,

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
			by using a DNS server, and forward requests for non-cacheable content to the resolved origin servers. Cacheable requests are sent to the configured cache servers. * REVERSE - Configure reverse proxy caches for specific origin servers. Incoming traffic directed to the reverse proxy can either be served from a cache server or be sent to the origin server with or without modification to the URL. The default value for cache type is TRANSPARENT if service is HTTP or SSL whereas the default cache type is FORWARD if the service is HDX. Possible values = TRANSPARENT,

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
			REVERSE, FORWARD
redirect	<String>	Read-write	Type of cache server to which to redirect HTTP requests. Available settings function as follows: * CACHE - Direct all requests to the cache. * POLICY - Apply the cache redirection policy to determine whether the request should be directed to the cache or to the origin. * ORIGIN - Direct all requests to the origin server. Default value: POLICY Possible values = CACHE, POLICY, ORIGIN
onpolicymatch	<String>	Read-write	Redirect requests that match the policy to either the cache or the origin server, as specified.

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
			Note: For this option to work, you must set the cache redirection type to POLICY. Default value: ORIGIN Possible values = CACHE, ORIGIN
redirecturl	<String>	Read-write	URL of the server to which to redirect traffic if the cache redirection virtual server configured on the Citrix ADC becomes unavailable. Minimum length = 1 Maximum length = 128
clttimeout	<Double>	Read-write	Time-out value, in seconds, after which to terminate an idle client connection. Minimum value = 0 Maximum value = 31536000
precedence	<String>	Read-write	Type of policy (URL or RULE) that takes precedence on the cache redirection virtual server. Applies only to cache redirection virtual

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
			servers that have both URL and RULE based policies. If you specify URL, URL based policies are applied first, in the following order: 1. Domain and exact URL 2. Domain, prefix and suffix 3. Domain and suffix 4. Domain and prefix 5. Domain only 6. Exact URL 7. Prefix and suffix 8. Suffix only 9. Prefix only 10. Default If you specify RULE, the rule based policies are applied before URL based policies are applied. Default value: RULE

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
			Possible values = RULE, URL
arp	<String>	Read-write	Use ARP to determine the destination MAC address. Possible values = ON, OFF
ghost	<String>	Read-write	. Possible values = ON, OFF
map	<String>	Read-write	Obsolete. Possible values = ON, OFF
format	<String>	Read-write	. Possible values = ON, OFF
via	<String>	Read-write	Insert a via header in each HTTP request. In the case of a cache miss, the request is redirected from the cache server to the origin server. This header indicates whether the request is being sent from a cache server. Default value: ON Possible values = ON, OFF
cachevserver	<String>	Read-write	Name of the default cache virtual server

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
			to which to redirect requests (the default target of the cache redirection virtual server). Minimum length = 1
dnsvservername	<String>	Read-write	Name of the DNS virtual server that resolves domain names arriving at the forward proxy virtual server. Note: This parameter applies only to forward proxy virtual servers, not reverse or transparent. Minimum length = 1
destinationvserver	<String>	Read-write	Destination virtual server for a transparent or forward proxy cache redirection virtual server. Minimum length = 1
domain	<String>	Read-write	Default domain for reverse proxies. Domains are configured to direct an incoming request from a specified source domain to a specified target domain. There can be several

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
			configured pairs of source and target domains. You can select one pair to be the default. If the host header or URL of an incoming request does not include a source domain, this option sends the request to the specified target domain. Minimum length = 1
sopersistencetimeout	<Double>	Read-write	Time-out, in minutes, for spillover persistence. Minimum value = 2 Maximum value = 24
sothreshold	<Double>	Read-write	For CONNECTION (or) DYNAMICCONNECTION spillover, the number of connections above which the virtual server enters spillover mode. For BANDWIDTH spillover, the amount of incoming and outgoing traffic (in Kbps) before spillover. For HEALTH spillover, the percentage of active services (by

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
			weight) below which spillover occurs. Minimum value = 1
reuse	<String>	Read-write	Reuse TCP connections to the origin server across client connections. Do not set this parameter unless the Service Type parameter is set to HTTP. If you set this parameter to OFF, the possible settings of the Redirect parameter function as follows: * CACHE - TCP connections to the cache servers are not reused. * ORIGIN - TCP connections to the origin servers are not reused. * POLICY - TCP connections to the origin servers are not reused. If you set the Reuse parameter to ON, connections to origin servers and connections to

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
			cache servers are reused. Default value: ON Possible values = ON, OFF
state	<String>	Read-write	Initial state of the cache redirection virtual server. Default value: ENABLED Possible values = ENABLED, DISABLED
downstateflush	<String>	Read-write	Perform delayed cleanup of connections to this virtual server. Default value: ENABLED Possible values = ENABLED, DISABLED
backupvserver	<String>	Read-write	Name of the backup virtual server to which traffic is forwarded if the active server becomes unavailable. Minimum length = 1
disableprimaryondown	<String>	Read-write	Continue sending traffic to a backup virtual server even

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
			after the primary virtual server comes UP from the DOWN state. Default value: DISABLED Possible values = ENABLED, DISABLED
l2conn	<String>	Read-write	Use L2 parameters, such as MAC, VLAN, and channel to identify a connection. Possible values = ON, OFF
backendssl	<String>	Read-write	Decides whether the backend connection made by Citrix ADC to the origin server will be HTTP or SSL. Applicable only for SSL type CR Forward proxy vserver. Default value: DISABLED Possible values = ENABLED, DISABLED
listenpolicy	<String>	Read-write	String specifying the listen policy for the cache redirection virtual server. Can be either an in-line expression or the

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
			name of a named expression. Default value: "NONE"
listenpriority	<Double>	Read-write	Priority of the listen policy specified by the Listen Policy parameter. The lower the number, higher the priority. Default value: 101 Minimum value = 0 Maximum value = 100
tcpprofilename	<String>	Read-write	Name of the profile containing TCP configuration information for the cache redirection virtual server. Minimum length = 1 Maximum length = 127
httpprofilename	<String>	Read-write	Name of the profile containing HTTP configuration information for cache redirection virtual server. Minimum length = 1 Maximum length = 127

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
comment	<String>	Read-write	Comments associated with this virtual server. Maximum length = 256
srcipexpr	<String>	Read-write	Expression used to extract the source IP addresses from the requests originating from the cache. Can be either an in-line expression or the name of a named expression. Minimum length = 1 Maximum length = 1500
originusip	<String>	Read-write	Use the client's IP address as the source IP address in requests sent to the origin server. Note: You can enable this parameter to implement fully transparent CR deployment. Possible values = ON, OFF
useportrange	<String>	Read-write	Use a port number from the port range (set by using the set ns param command, or in the Create

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
			Virtual Server (Cache Redirection) dialog box) as the source port in the requests sent to the origin server. Default value: OFF Possible values = ON, OFF
appflowlog	<String>	Read-write	Enable logging of AppFlow information. Default value: ENABLED Possible values = ENABLED, DISABLED
netprofile	<String>	Read-write	Name of the network profile containing network configurations for the cache redirection virtual server. Minimum length = 1 Maximum length = 127
icmpvsrresponse	<String>	Read-write	Criterion for responding to PING requests sent to this virtual server. If ACTIVE, respond only if the virtual server is available. If PASSIVE, respond even if the virtual

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
			server is not available. Default value: PASSIVE Possible values = PASSIVE, ACTIVE
rhistate	<String>	Read-write	A host route is injected according to the setting on the virtual servers * If set to PASSIVE on all the virtual servers that share the IP address, the appliance always injects the hostroute. * If set to ACTIVE on all the virtual servers that share the IP address, the appliance injects even if one virtual server is UP. * If set to ACTIVE on some virtual servers and PASSIVE on the others, the appliance, injects even if one virtual server set to ACTIVE is UP.

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
			Default value: PASSIVE Possible values = PASSIVE, ACTIVE
useoriginipportforcache	<String>	Read-write	Use origin ip/port while forwarding request to the cache. Change the destination IP, destination port of the request came to CR vserver to Origin IP and Origin Port and forward it to Cache. Default value: NO Possible values = YES, NO
newname	<String>	Read-write	New name for the cache redirection virtual server. Must begin with an ASCII alphanumeric or underscore (_) character, and must contain only ASCII alphanumeric, underscore, hash (#), period (.), space, colon (:), at sign (@), equal sign (=), and hyphen (-) characters. If the name includes one or more spaces, enclose the name in double or single

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
			quotation marks (for example, "my name" or 'my name'). Minimum length = 1
ip	<String>	Read-only	.
value	<String>	Read-only	The ssl card status for the transparent ssl cr vserver. Possible values = Certkey not bound, SSL feature disabled
ngname	<String>	Read-only	Nodegroup devno to which this crvserver belongs to.
type	<String>	Read-only	Virtual server type. Possible values = CONTENT, ADDRESS
curstate	<String>	Read-only	The state of the cr vserver. Possible values = UP, DOWN, UNKNOWN, BUSY, OUT OF SERVICE, GOING OUT OF SERVICE, DOWN WHEN GOING OUT OF SERVICE, NS_EMPTY_STR, Unknown, DISABLED

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
status	<Integer>	Read-only	Status.
sc	<String>	Read-only	The state of SureConnect the specified virtual server. Possible values = ON, OFF
authentication	<String>	Read-only	Authentication. Possible values = ON, OFF
homepage	<String>	Read-only	Home page.
rule	<String>	Read-only	Rule.
policyname	<String>	Read-only	Policies bound to this vserver.
pipolicyhits	<Double>	Read-only	Number of hits.
servicename	<String>	Read-only	Service name.
weight	<Double>	Read-only	Weight for this service.
targetvserver	<String>	Read-only	The CSW target server names.
priority	<Double>	Read-only	The priority for the policy.
somethod	<String>	Read-only	The spillover factor. When the main virtual server reaches this spillover threshold, it will give further traffic to the backupvserver.

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
			Possible values = CONNECTION, DYNAMICCONNECTION, BANDWIDTH, HEALTH, NONE
sopersistence	<String>	Read-only	The state of spillover persistence. Possible values = ENABLED, DISABLED
lbvserver	<String>	Read-only	The Default target server name. Minimum length = 1
bindpoint	<String>	Read-only	The bindpoint to which the policy is bound. Possible values = REQUEST, RESPONSE, ICA_REQUEST
invoke	<Boolean>	Read-only	Invoke flag.
labeltype	<String>	Read-only	The invocation type. Possible values = reqvserver, resvserver, policylabel
labelname	<String>	Read-only	Name of the label invoked.
gotopriorityexpression	<String>	Read-only	Expression specifying the priority of the next policy

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
			which will get evaluated if the current policy rule evaluates to TRUE.
nodefaultbindings	<String>	Read-only	to determine if the configuration will have default ssl CIPHER and ECC curve bindings. Default value: NO Possible values = YES, NO
__count	<Double>	Read-only	count parameter

Operations

(click to see Properties)

ADD| DELETE| UPDATE| UNSET| ENABLE| DISABLE| RENAME| GET (ALL)| GET| COUNT

Some options that you can use for each operations:

- **Getting warnings in response:**NITRO allows you to get warnings in an operation by specifying the "warning" query parameter as "yes". For example, to get warnings while connecting to the Citrix ADC appliance, the URL is as follows:

  http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/login?warning=yes

  If any, the warnings are displayed in the response payload with the HTTP code "209 X-NITRO-WARNING".

- **Authenticated access for individual NITRO operations:**NITRO allows you to logon to the Citrix ADC appliance to perform individual operations. You can use this option instead of creating a NITRO session (using the login object) and then using that session to perform all operations,

  To do this, you must specify the username and password in the request header of the NITRO request as follows:

X-NITRO-USER:<username>

X-NITRO-PASS:<password>

Note: In such cases, make sure that the request header DOES not include the following:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue>

`*Note:`*

Mandatory parameters are marked in redand placeholder content is marked in <green>.

add

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/crvserver

HTTP Method:POST

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Content-Type:application/json

Request Payload:

```
{"crvserver":{

<b>"name":<String_value>,

</b>"td":<Double_value>,

<b>"servicetype":<String_value>,

</b>"ipv46":<String_value>,

"port":<Integer_value>,

"ipset":<String_value>,

"range":<Double_value>,

"cachetype":<String_value>,

"redirect":<String_value>,
```

```
"onpolicymatch":<String_value>,

"redirecturl":<String_value>,

"clttimeout":<Double_value>,

"precedence":<String_value>,

"arp":<String_value>,

"ghost":<String_value>,

"map":<String_value>,

"format":<String_value>,

"via":<String_value>,

"cachevserver":<String_value>,

"dnsvservername":<String_value>,

"destinationvserver":<String_value>,

"domain":<String_value>,

"sopersistencetimeout":<Double_value>,

"sothreshold":<Double_value>,

"reuse":<String_value>,

"state":<String_value>,

"downstateflush":<String_value>,

"backupvserver":<String_value>,

"disableprimaryondown":<String_value>,

"l2conn":<String_value>,

"backendssl":<String_value>,
```

```
"listenpolicy":<String_value>,

"listenpriority":<Double_value>,

"tcpprofilename":<String_value>,

"httpprofilename":<String_value>,

"comment":<String_value>,

"srcipexpr":<String_value>,

"originusip":<String_value>,

"useportrange":<String_value>,

"appflowlog":<String_value>,

"netprofile":<String_value>,

"icmpvsrresponse":<String_value>,

"rhistate":<String_value>,

"useoriginipportforcache":<String_value>

}}
```

Response:

HTTP Status Code on Success: 201 Created HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error

delete

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/crvserver/name_value<String>

HTTP Method:DELETE

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue>

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error

update

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/crvserver

HTTP Method:PUT

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Content-Type:application/json

Request Payload:

```
{"crvserver":{

<b>"name":<String_value>,

</b>"ipv46":<String_value>,

"ipset":<String_value>,

"redirect":<String_value>,

"onpolicymatch":<String_value>,

"precedence":<String_value>,

"arp":<String_value>,

"via":<String_value>,

"cachevserver":<String_value>,

"dnsvservername":<String_value>,

"destinationvserver":<String_value>,
```

```
"domain":<String_value>,

"reuse":<String_value>,

"backupvserver":<String_value>,

"disableprimaryondown":<String_value>,

"redirecturl":<String_value>,

"clttimeout":<Double_value>,

"downstateflush":<String_value>,

"l2conn":<String_value>,

"backendssl":<String_value>,

"listenpolicy":<String_value>,

"listenpriority":<Double_value>,

"tcpprofilename":<String_value>,

"httpprofilename":<String_value>,

"netprofile":<String_value>,

"comment":<String_value>,

"srcipexpr":<String_value>,

"originusip":<String_value>,

"useportrange":<String_value>,

"appflowlog":<String_value>,

"icmpvsrresponse":<String_value>,

"rhistate":<String_value>,

"useoriginipportforcache":<String_value>
```

```
}}
```

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error

unset

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/crvserver?**action=unset**

HTTP Method:POST

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Content-Type:application/json

Request Payload:

```
{"crvserver":{

<b>"name":<String_value>,

</b>"cachevserver":true,

"dnsvservername":true,

"destinationvserver":true,

"domain":true,

"backupvserver":true,

"clttimeout":true,

"redirecturl":true,

"l2conn":true,
```

```
"backendssl":true,

"originusip":true,

"useportrange":true,

"srcipexpr":true,

"tcpprofilename":true,

"httpprofilename":true,

"appflowlog":true,

"netprofile":true,

"icmpvsrresponse":true,

"ipset":true,

"redirect":true,

"onpolicymatch":true,

"precedence":true,

"arp":true,

"via":true,

"reuse":true,

"disableprimaryondown":true,

"downstateflush":true,

"listenpolicy":true,

"listenpriority":true,

"comment":true,

"rhistate":true,
```

```
"useoriginipportforcache":true

}}
```

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error

enable

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/crvserver?**action=enable**

HTTP Method:POST

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Content-Type:application/json

Request Payload:

```
{"crvserver":{

<b>"name":<String_value>

</b>}}
```

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error

disable

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/crvserver?**action=disable**

HTTP Method:POST

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Content-Type:application/json

Request Payload:

```
{"crvserver":{

<b>"name":<String_value>

</b>}}
```

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error

rename

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/crvserver?**action=rename**

HTTP Method:POST

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Content-Type:application/json

Request Payload:

```
{"crvserver":{

<b>"name":<String_value>,

</b><b>"newname":<String_value>

</b>}}
```

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error

get (all)

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/crvserver

Query-parameters:

attrs

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/crvserver?**attrs=property-name1,property-name2**

Use this query parameter to specify the resource details that you want to retrieve.

filter

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/crvserver?**filter=property-name1:property-val1,property-name2:property-val2**

Use this query-parameter to get the filtered set of crvserver resources configured on Citrix ADC. Filtering can be done on any of the properties of the resource.

view

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/crvserver?**view=summary**

Note: By default, the retrieved results are displayed in detail view (?view=detail).

pagination

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/crvserver?**pagesize=#no;pageno=#no**

Use this query-parameter to get the crvserver resources in chunks.

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "crvserver": [ {

"name":<String_value>,

"ip":<String_value>,

"td":<Double_value>,

"ipv46":<String_value>,

"value":<String_value>,

"port":<Integer_value>,

"range":<Double_value>,

"servicetype":<String_value>,

"ngname":<String_value>,

"type":<String_value>,

"curstate":<String_value>,

"status":<Integer_value>,

"sc":<String_value>,

"cachetype":<String_value>,

"redirect":<String_value>,

"onpolicymatch":<String_value>,
```

```
"precedence":<String_value>,

"redirecturl":<String_value>,

"authentication":<String_value>,

"homepage":<String_value>,

"dnsvservername":<String_value>,

"domain":<String_value>,

"rule":<String_value>,

"policyname":<String_value>,

"pipolicyhits":<Double_value>,

"servicename":<String_value>,

"weight":<Double_value>,

"cachevserver":<String_value>,

"targetvserver":<String_value>,

"backupvserver":<String_value>,

"priority":<Double_value>,

"clttimeout":<Double_value>,

"somethod":<String_value>,

"sopersistence":<String_value>,

"sopersistencetimeout":<Double_value>,

"sothreshold":<Double_value>,

"reuse":<String_value>,

"arp":<String_value>,
```

```
"destinationvserver":<String_value>,

"via":<String_value>,

"downstateflush":<String_value>,

"disableprimaryondown":<String_value>,

"l2conn":<String_value>,

"backendssl":<String_value>,

"comment":<String_value>,

"listenpolicy":<String_value>,

"listenpriority":<Double_value>,

"tcpprofilename":<String_value>,

"httpprofilename":<String_value>,

"srcipexpr":<String_value>,

"originusip":<String_value>,

"useportrange":<String_value>,

"appflowlog":<String_value>,

"netprofile":<String_value>,

"icmpvsrresponse":<String_value>,

"rhistate":<String_value>,

"lbvserver":<String_value>,

"bindpoint":<String_value>,

"invoke":<Boolean_value>,

"labeltype":<String_value>,
```

```
"labelname":<String_value>,

"gotopriorityexpression":<String_value>,

"ipset":<String_value>,

"nodefaultbindings":<String_value>,

"useoriginipportforcache":<String_value>

}]}
```

get

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/crvserver/name_value<String>

Query-parameters:

attrs

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/crvserver/name_value<String>?**attrs=property-name1,property-name2**

Use this query parameter to specify the resource details that you want to retrieve.

view

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/crvserver/name_value<String>?**view=summary**

Note: By default, the retrieved results are displayed in detail view (?view=detail).

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "crvserver": [ {

"name":<String_value>,

"ip":<String_value>,

"td":<Double_value>,

"ipv46":<String_value>,

"value":<String_value>,

"port":<Integer_value>,

"range":<Double_value>,

"servicetype":<String_value>,

"ngname":<String_value>,

"type":<String_value>,

"curstate":<String_value>,

"status":<Integer_value>,

"sc":<String_value>,

"cachetype":<String_value>,

"redirect":<String_value>,

"onpolicymatch":<String_value>,

"precedence":<String_value>,

"redirecturl":<String_value>,
```

```
"authentication":<String_value>,

"homepage":<String_value>,

"dnsvservername":<String_value>,

"domain":<String_value>,

"rule":<String_value>,

"policyname":<String_value>,

"pipolicyhits":<Double_value>,

"servicename":<String_value>,

"weight":<Double_value>,

"cachevserver":<String_value>,

"targetvserver":<String_value>,

"backupvserver":<String_value>,

"priority":<Double_value>,

"clttimeout":<Double_value>,

"somethod":<String_value>,

"sopersistence":<String_value>,

"sopersistencetimeout":<Double_value>,

"sothreshold":<Double_value>,

"reuse":<String_value>,

"arp":<String_value>,

"destinationvserver":<String_value>,

"via":<String_value>,
```

```
"downstateflush":<String_value>,

"disableprimaryondown":<String_value>,

"l2conn":<String_value>,

"backendssl":<String_value>,

"comment":<String_value>,

"listenpolicy":<String_value>,

"listenpriority":<Double_value>,

"tcpprofilename":<String_value>,

"httpprofilename":<String_value>,

"srcipexpr":<String_value>,

"originusip":<String_value>,

"useportrange":<String_value>,

"appflowlog":<String_value>,

"netprofile":<String_value>,

"icmpvsrresponse":<String_value>,

"rhistate":<String_value>,

"lbvserver":<String_value>,

"bindpoint":<String_value>,

"invoke":<Boolean_value>,

"labeltype":<String_value>,

"labelname":<String_value>,

"gotopriorityexpression":<String_value>,
```

```
"ipset":<String_value>,

"nodefaultbindings":<String_value>,

"useoriginipportforcache":<String_value>

}]}
```

count

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/crvserver?**count=yes**

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "crvserver": [ { "__count": "#no"} ] }
```

crvserver_appflowpolicy_binding

Binding object showing the appflowpolicy that can be bound to crvserver.

Properties

(click to see Operations)

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
priority	<Double>	Read-write	The priority for the policy.
bindpoint	<String>	Read-write	For a rewrite policy, the bind point to which to bind the policy. Note: This parameter applies only to rewrite policies, because content switching policies are evaluated only at request time. Possible values = REQUEST, RESPONSE, ICA_REQUEST
policyname	<String>	Read-write	Policies bound to this vserver.
labelname	<String>	Read-write	Name of the label invoked.
name	<String>	Read-write	Name of the cache redirection virtual server to which to bind the cache redirection policy. Minimum length = 1
targetvserver	<String>	Read-write	Name of the virtual server to which content is forwarded. Applicable only if the policy is a map policy and the cache redirection virtual

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
			server is of type REVERSE. Minimum length = 1
gotopriorityexpression	<String>	Read-write	Expression specifying the priority of the next policy which will get evaluated if the current policy rule evaluates to TRUE.
invoke	<Boolean>	Read-write	Invoke flag.
labeltype	<String>	Read-write	The invocation type. Possible values = reqvserver, resvserver, policylabel
__count	<Double>	Read-write	count parameter

Operations

(click to see Properties)

ADD:| DELETE:| GET| GET (ALL)| COUNT

Some options that you can use for each operations:

- **Getting warnings in response:**NITRO allows you to get warnings in an operation by specifying the "warning" query parameter as "yes". For example, to get warnings while connecting to the Citrix ADC appliance, the URL is as follows:

  http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/login?warning=yes

  If any, the warnings are displayed in the response payload with the HTTP code "209 X-NITRO-WARNING".
- **Authenticated access for individual NITRO operations:**NITRO allows you to logon to the Citrix ADC appliance to perform individual operations. You can use this option instead of

creating a NITRO session (using the login object) and then using that session to perform all operations,

To do this, you must specify the username and password in the request header of the NITRO request as follows:

X-NITRO-USER:<username>

X-NITRO-PASS:<password>

Note: In such cases, make sure that the request header DOES not include the following:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue>

`*Note:`*

Mandatory parameters are marked in redand placeholder content is marked in <green>.

add

URL:http://<Citrix-ADC-IP-address(NSIP)/nitro/v1/config/crvserver_appflowpolicy_binding

HTTP Method:PUT

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Content-Type:application/json

Request Payload:

```
{

"crvserver_appflowpolicy_binding":{

<b>"name":<String_value>,

</b>"policyname":<String_value>,

"targetvserver":<String_value>,

"priority":<Double_value>,

"gotopriorityexpression":<String_value>,
```

```
"bindpoint":<String_value>,

"invoke":<Boolean_value>,

"labeltype":<String_value>,

"labelname":<String_value>

}}
```

Response:

HTTP Status Code on Success: 201 Created HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error

delete

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/crvserver_appflowpolicy_binding/name_value<String>

Query-parameters:

args

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/crvserver_appflowpolicy_binding/name_value<String>?
args=policyname:<String_value>,bindpoint:<String_value>,priority:<Double_value>

HTTP Method:DELETE

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue>

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error

get

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/crvserver_appflowpolicy_binding/name_value<String>

Query-parameters:

filter

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/crvserver_appflowpolicy_binding/name_value<String>?**filter=property-name1:property-value1,property-name2:property-value2**

Use this query-parameter to get the filtered set of crvserver_appflowpolicy_binding resources configured on Citrix ADC. Filtering can be done on any of the properties of the resource.

pagination

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/crvserver_appflowpolicy_binding/name_value<String>?**pagesize=#no;pageno=#no**

Use this query-parameter to get the crvserver_appflowpolicy_binding resources in chunks.

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "crvserver_appflowpolicy_binding": [ {

"priority":<Double_value>,

"bindpoint":<String_value>,
```

```
"policyname":<String_value>,

"labelname":<String_value>,

"name":<String_value>,

"targetvserver":<String_value>,

"gotopriorityexpression":<String_value>,

"invoke":<Boolean_value>,

"labeltype":<String_value>

}]}
```

get (all)

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/crvserver_appflowpolicy_binding

Query-parameters:

bulkbindings

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/crvserver_appflowpolicy_binding?**bulkbindings=yes**

NITRO allows you to fetch bindings in bulk.

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "crvserver_appflowpolicy_binding": [ {

"priority":<Double_value>,

"bindpoint":<String_value>,

"policyname":<String_value>,

"labelname":<String_value>,

"name":<String_value>,

"targetvserver":<String_value>,

"gotopriorityexpression":<String_value>,

"invoke":<Boolean_value>,

"labeltype":<String_value>

}]}
```

count

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/crvserver_appflowpolicy_binding/name_value<String>?**count=yes**

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{"crvserver_appflowpolicy_binding": [ { "__count": "#no"} ] }
```

crvserver_appfwpolicy_binding

Binding object showing the appfwpolicy that can be bound to crvserver.

Properties

(click to see Operations)

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
priority	<Double>	Read-write	The priority for the policy.
bindpoint	<String>	Read-write	The bindpoint to which the policy is bound. Possible values = REQUEST, RESPONSE, ICA_REQUEST
policyname	<String>	Read-write	Policies bound to this vserver.
labelname	<String>	Read-write	Name of the label invoked.
name	<String>	Read-write	Name of the cache redirection virtual server to which to bind the cache redirection policy. Minimum length = 1

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
targetvserver	<String>	Read-write	Name of the virtual server to which content is forwarded. Applicable only if the policy is a map policy and the cache redirection virtual server is of type REVERSE. Minimum length = 1
gotopriorityexpression	<String>	Read-write	Expression specifying the priority of the next policy which will get evaluated if the current policy rule evaluates to TRUE.
invoke	<Boolean>	Read-write	Invoke flag.
labeltype	<String>	Read-write	The invocation type. Possible values = reqvserver, resvserver, policylabel
sc	<String>	Read-only	The state of SureConnect the specified virtual server. Possible values = ON, OFF
__count	<Double>	Read-only	count parameter

Operations

(click to see Properties)

ADD:| DELETE:| GET| GET (ALL)| COUNT

Some options that you can use for each operations:

- **Getting warnings in response:**NITRO allows you to get warnings in an operation by specifying the "warning" query parameter as "yes". For example, to get warnings while connecting to the Citrix ADC appliance, the URL is as follows:

  http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/login?warning=yes

  If any, the warnings are displayed in the response payload with the HTTP code "209 X-NITRO-WARNING".

- **Authenticated access for individual NITRO operations:**NITRO allows you to logon to the Citrix ADC appliance to perform individual operations. You can use this option instead of creating a NITRO session (using the login object) and then using that session to perform all operations,

  To do this, you must specify the username and password in the request header of the NITRO request as follows:

  X-NITRO-USER:<username>

  X-NITRO-PASS:<password>

  Note: In such cases, make sure that the request header DOES not include the following:

  Cookie:NITRO_AUTH_TOKEN=<tokenvalue>

`*Note:`*

Mandatory parameters are marked in redand placeholder content is marked in <green>.

add

URL:http://<Citrix-ADC-IP-address(NSIP)/nitro/v1/config/crvserver_appfwpolicy_binding

HTTP Method:PUT

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Content-Type:application/json

Request Payload:

```
{

"crvserver_appfwpolicy_binding":{

<b>"name":<String_value>,

</b>"policyname":<String_value>,

"targetvserver":<String_value>,

"priority":<Double_value>,

"gotopriorityexpression":<String_value>,

"bindpoint":<String_value>,

"invoke":<Boolean_value>,

"labeltype":<String_value>,

"labelname":<String_value>

}}
```

Response:

HTTP Status Code on Success: 201 Created HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error

delete

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/crvserver_appfwpolicy_binding/name_value<String>

Query-parameters:

args

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/crvserver_appfwpolicy_binding/name_value<String>?**args=policyname:<String_value>,bindpoint:<String_value>,priority:<Double_value>**

HTTP Method:DELETE

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue>

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error

get

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/crvserver_appfwpolicy_binding/name_value<String>

Query-parameters:

filter

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/crvserver_appfwpolicy_binding/name_value<String>?**filter=property-name1:property-value1,property-name2:property-value2**

Use this query-parameter to get the filtered set of crvserver_appfwpolicy_binding resources configured on Citrix ADC. Filtering can be done on any of the properties of the resource.

pagination

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/crvserver_appfwpolicy_binding/name_value<String>?**pagesize=#no;pageno=#no**

Use this query-parameter to get the crvserver_appfwpolicy_binding resources in chunks.

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "crvserver_appfwpolicy_binding": [ {

"priority":<Double_value>,

"bindpoint":<String_value>,

"policyname":<String_value>,

"labelname":<String_value>,

"name":<String_value>,

"targetvserver":<String_value>,

"gotopriorityexpression":<String_value>,

"invoke":<Boolean_value>,

"labeltype":<String_value>,

"sc":<String_value>

}]}
```

get (all)

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/crvserver_appfwpolicy_binding

Query-parameters:

bulkbindings

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/crvserver_appfwpolicy_binding?**bulkbindings=yes**

NITRO allows you to fetch bindings in bulk.

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "crvserver_appfwpolicy_binding": [ {

"priority":<Double_value>,

"bindpoint":<String_value>,

"policyname":<String_value>,

"labelname":<String_value>,

"name":<String_value>,

"targetvserver":<String_value>,

"gotopriorityexpression":<String_value>,

"invoke":<Boolean_value>,

"labeltype":<String_value>,

"sc":<String_value>
```

```
}]}
```

count

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/crvserver_appfwpolicy_binding/name_value<String>?**count=yes**

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{"crvserver_appfwpolicy_binding": [ { "__count": "#no"} ] }
```

crvserver_appqoepolicy_binding

Binding object showing the appqoepolicy that can be bound to crvserver.

Properties

(click to see Operations)

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
priority	<Double>	Read-write	The priority for the policy.

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
bindpoint	<String>	Read-write	The bindpoint to which the policy is bound. Possible values = REQUEST, RESPONSE, ICA_REQUEST
policyname	<String>	Read-write	Policies bound to this vserver.
labelname	<String>	Read-write	Name of the label invoked.
name	<String>	Read-write	Name of the cache redirection virtual server to which to bind the cache redirection policy. Minimum length = 1
targetvserver	<String>	Read-write	Name of the virtual server to which content is forwarded. Applicable only if the policy is a map policy and the cache redirection virtual server is of type REVERSE. Minimum length = 1
gotopriorityexpression	<String>	Read-write	Expression specifying the priority of the next policy which will get evaluated if the current policy rule evaluates to TRUE.

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
invoke	<Boolean>	Read-write	Invoke flag.
labeltype	<String>	Read-write	The invocation type. Possible values = reqvserver, resvserver, policylabel
__count	<Double>	Read-write	count parameter

Operations

(click to see Properties)

ADD:| DELETE:| GET| GET (ALL)| COUNT

Some options that you can use for each operations:

- **Getting warnings in response:**NITRO allows you to get warnings in an operation by specifying the "warning" query parameter as "yes". For example, to get warnings while connecting to the Citrix ADC appliance, the URL is as follows:

  http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/login?warning=yes

  If any, the warnings are displayed in the response payload with the HTTP code "209 X-NITRO-WARNING".

- **Authenticated access for individual NITRO operations:**NITRO allows you to logon to the Citrix ADC appliance to perform individual operations. You can use this option instead of creating a NITRO session (using the login object) and then using that session to perform all operations,

  To do this, you must specify the username and password in the request header of the NITRO request as follows:

  X-NITRO-USER:<username>

  X-NITRO-PASS:<password>

  Note: In such cases, make sure that the request header DOES not include the following:

  Cookie:NITRO_AUTH_TOKEN=<tokenvalue>

`*Note:` *

Mandatory parameters are marked in redand placeholder content is marked in <green>.

add

URL:http://<Citrix-ADC-IP-address(NSIP)/nitro/v1/config/crvserver_appqoepolicy_binding

HTTP Method:PUT

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Content-Type:application/json

Request Payload:

```
{

"crvserver_appqoepolicy_binding":{

<b>"name":<String_value>,

</b>"policyname":<String_value>,

"targetvserver":<String_value>,

"priority":<Double_value>,

"gotopriorityexpression":<String_value>,

"bindpoint":<String_value>,

"invoke":<Boolean_value>,

"labeltype":<String_value>,

"labelname":<String_value>

}}
```

Response:

HTTP Status Code on Success: 201 Created HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error

delete

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/crvserver_appqoepolicy_binding/name_value<String>

Query-parameters:

args

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/crvserver_appqoepolicy_binding/name_value<String>?**args=policyname:<String_value>,bindpoint:<String_value>,priority:<Double_value>**

HTTP Method:DELETE

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue>

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error

get

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/crvserver_appqoepolicy_binding/name_value<String>

Query-parameters:

filter

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/crvserver_appqoepolicy_binding/name_value<String>?**filter=property-name1:property-value1,property-name2:property-value2**

Use this query-parameter to get the filtered set of crvserver_appqoepolicy_binding resources configured on Citrix ADC. Filtering can be done on any of the properties of the resource.

pagination

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/crvserver_appqoepolicy_binding/name_value<String>?**pagesize=#no;pageno=#no**

Use this query-parameter to get the crvserver_appqoepolicy_binding resources in chunks.

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "crvserver_appqoepolicy_binding": [ {

"priority":<Double_value>,

"bindpoint":<String_value>,

"policyname":<String_value>,

"labelname":<String_value>,

"name":<String_value>,

"targetvserver":<String_value>,

"gotopriorityexpression":<String_value>,

"invoke":<Boolean_value>,

"labeltype":<String_value>

}]}
```

get (all)

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/crvserver_appqoepolicy_binding

Query-parameters:

bulkbindings

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/crvserver_appqoepolicy_binding?**bulkbindings=yes**

NITRO allows you to fetch bindings in bulk.

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "crvserver_appqoepolicy_binding": [ {

"priority":<Double_value>,

"bindpoint":<String_value>,

"policyname":<String_value>,

"labelname":<String_value>,

"name":<String_value>,
```

```
"targetvserver":<String_value>,

"gotopriorityexpression":<String_value>,

"invoke":<Boolean_value>,

"labeltype":<String_value>

}]}
```

count

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/crvserver_appqoepolicy_binding/name_value<String>?**count=yes**

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{"crvserver_appqoepolicy_binding": [ { "__count": "#no"} ] }
```

crvserver_binding

Binding object which returns the resources bound to crvserver.

Properties

(click to see Operations)

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
name	<String>	Read-write	Name of a cache redirection virtual server about which to display detailed information. Minimum length = 1
crvserver_spilloverpolicy_binding	<crvserver_spilloverpolicy_binding[]>	Read-only	spilloverpolicy that can be bound to crvserver.
crvserver_filterpolicy_binding	<crvserver_filterpolicy_binding[]>	Read-only	filterpolicy that can be bound to crvserver.
crvserver_icapolicy_binding	<crvserver_icapolicy_binding[]>	Read-only	icapolicy that can be bound to crvserver.
crvserver_cmppolicy_binding	<crvserver_cmppolicy_binding[]>	Read-only	cmppolicy that can be bound to crvserver.
crvserver_lbvserver_binding	<crvserver_lbvserver_binding[]>	Read-only	lbvserver that can be bound to crvserver.
crvserver_appflowpolicy_binding	<crvserver_appflowpolicy_binding[]>	Read-only	appflowpolicy that can be bound to crvserver.
crvserver_responderpolicy_binding	<crvserver_responderpolicy_binding[]>	Read-only	responderpolicy that can be bound to crvserver.
crvserver_policymap_binding	<crvserver_policymap_binding[]>	Read-only	policymap that can be bound to crvserver.

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
crvserver_feopolicy_binding	<crvserver_feopolicy_binding[]>	Read-only	feopolicy that can be bound to crvserver.
crvserver_cachepolicy_binding	<crvserver_cachepolicy_binding[]>	Read-only	cachepolicy that can be bound to crvserver.
crvserver_rewritepolicy_binding	<crvserver_rewritepolicy_binding[]>	Read-only	rewritepolicy that can be bound to crvserver.
crvserver_cspolicy_binding	<crvserver_cspolicy_binding[]>	Read-only	cspolicy that can be bound to crvserver.
crvserver_appqoepolicy_binding	<crvserver_appqoepolicy_binding[]>	Read-only	appqoepolicy that can be bound to crvserver.
crvserver_appfwpolicy_binding	<crvserver_appfwpolicy_binding[]>	Read-only	appfwpolicy that can be bound to crvserver.
crvserver_crpolicy_binding	<crvserver_crpolicy_binding[]>	Read-only	crpolicy that can be bound to crvserver.

Operations

(click to see Properties)

GET| GET (ALL)

Some options that you can use for each operations:

- **Getting warnings in response:**NITRO allows you to get warnings in an operation by specifying the "warning" query parameter as "yes". For example, to get warnings while connecting to the Citrix ADC appliance, the URL is as follows:

  http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/login?warning=yes

  If any, the warnings are displayed in the response payload with the HTTP code "209 X-NITRO-WARNING".

- **Authenticated access for individual NITRO operations:**NITRO allows you to logon to the Citrix ADC appliance to perform individual operations. You can use this option instead of creating a NITRO session (using the login object) and then using that session to perform all operations,

  To do this, you must specify the username and password in the request header of the NITRO request as follows:

  X-NITRO-USER:<username>

  X-NITRO-PASS:<password>

  Note: In such cases, make sure that the request header DOES not include the following:

  Cookie:NITRO_AUTH_TOKEN=<tokenvalue>

`*Note:`*

Mandatory parameters are marked in redand placeholder content is marked in <green>.

get

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/crvserver_binding/name_value<String>

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "crvserver_binding": [ {

"name":<String_value>,
```

```
"crvserver_spilloverpolicy_binding":<crvserver_spilloverpolicy_binding[]_value>,

"crvserver_policymap_binding":<crvserver_policymap_binding[]_value>,

"crvserver_icapolicy_binding":<crvserver_icapolicy_binding[]_value>,

"crvserver_cachepolicy_binding":<crvserver_cachepolicy_binding[]_value>,

"crvserver_lbvserver_binding":<crvserver_lbvserver_binding[]_value>,

"crvserver_appfwpolicy_binding":<crvserver_appfwpolicy_binding[]_value>,

"crvserver_responderpolicy_binding":<crvserver_responderpolicy_binding[]_value>,

"crvserver_filterpolicy_binding":<crvserver_filterpolicy_binding[]_value>,

"crvserver_cmppolicy_binding":<crvserver_cmppolicy_binding[]_value>,

"crvserver_appqoepolicy_binding":<crvserver_appqoepolicy_binding[]_value>,

"crvserver_feopolicy_binding":<crvserver_feopolicy_binding[]_value>,

"crvserver_cspolicy_binding":<crvserver_cspolicy_binding[]_value>,

"crvserver_crpolicy_binding":<crvserver_crpolicy_binding[]_value>,

"crvserver_rewritepolicy_binding":<crvserver_rewritepolicy_binding[]_value>,

"crvserver_appflowpolicy_binding":<crvserver_appflowpolicy_binding[]_value>

}]}
```

get (all)

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/crvserver_binding

Query-parameters:

bulkbindings

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/crvserver_binding?**bulkbindings=yes**

NITRO allows you to fetch bindings in bulk.

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "crvserver_binding": [ {

"name":<String_value>,

"crvserver_spilloverpolicy_binding":<crvserver_spilloverpolicy_binding[]_value>,

"crvserver_policymap_binding":<crvserver_policymap_binding[]_value>,

"crvserver_icapolicy_binding":<crvserver_icapolicy_binding[]_value>,

"crvserver_cachepolicy_binding":<crvserver_cachepolicy_binding[]_value>,

"crvserver_lbvserver_binding":<crvserver_lbvserver_binding[]_value>,

"crvserver_appfwpolicy_binding":<crvserver_appfwpolicy_binding[]_value>,

"crvserver_responderpolicy_binding":<crvserver_responderpolicy_binding[]_value>,
```

```
"crvserver_filterpolicy_binding":<crvserver_filterpolicy_binding[]_value>,

"crvserver_cmppolicy_binding":<crvserver_cmppolicy_binding[]_value>,

"crvserver_appqoepolicy_binding":<crvserver_appqoepolicy_binding[]_value>,

"crvserver_feopolicy_binding":<crvserver_feopolicy_binding[]_value>,

"crvserver_cspolicy_binding":<crvserver_cspolicy_binding[]_value>,

"crvserver_crpolicy_binding":<crvserver_crpolicy_binding[]_value>,

"crvserver_rewritepolicy_binding":<crvserver_rewritepolicy_binding[]_value
>,

"crvserver_appflowpolicy_binding":<crvserver_appflowpolicy_binding[]_value
>

}]}
```

crvserver_cachepolicy_binding

Binding object showing the cachepolicy that can be bound to crvserver.

Properties

(click to see Operations)

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
priority	<Double>	Read-write	The priority for the policy.
bindpoint	<String>	Read-write	The bindpoint to which the policy is bound. Possible values = REQUEST, RESPONSE, ICA_REQUEST

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
policyname	<String>	Read-write	Policies bound to this vserver.
labelname	<String>	Read-write	Name of the label invoked.
name	<String>	Read-write	Name of the cache redirection virtual server to which to bind the cache redirection policy. Minimum length = 1
targetvserver	<String>	Read-write	Name of the virtual server to which content is forwarded. Applicable only if the policy is a map policy and the cache redirection virtual server is of type REVERSE. Minimum length = 1
gotopriorityexpression	<String>	Read-write	Expression specifying the priority of the next policy which will get evaluated if the current policy rule evaluates to TRUE.
invoke	<Boolean>	Read-write	Invoke flag.
labeltype	<String>	Read-write	The invocation type. Possible values = reqvserver, resvserver, policylabel

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
__count	<Double>	Read-write	count parameter

Operations

(click to see Properties)

ADD:| DELETE:| GET| GET (ALL)| COUNT

Some options that you can use for each operations:

- **Getting warnings in response:**NITRO allows you to get warnings in an operation by specifying the "warning" query parameter as "yes". For example, to get warnings while connecting to the Citrix ADC appliance, the URL is as follows:

  http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/login?warning=yes

  If any, the warnings are displayed in the response payload with the HTTP code "209 X-NITRO-WARNING".
- **Authenticated access for individual NITRO operations:**NITRO allows you to logon to the Citrix ADC appliance to perform individual operations. You can use this option instead of creating a NITRO session (using the login object) and then using that session to perform all operations,

  To do this, you must specify the username and password in the request header of the NITRO request as follows:

  X-NITRO-USER:<username>

  X-NITRO-PASS:<password>

  Note: In such cases, make sure that the request header DOES not include the following:

  Cookie:NITRO_AUTH_TOKEN=<tokenvalue>

`*Note:`*

Mandatory parameters are marked in redand placeholder content is marked in <green>.

add

URL:http://<Citrix-ADC-IP-address(NSIP)/nitro/v1/config/crvserver_cachepolicy_binding

HTTP Method:PUT

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Content-Type:application/json

Request Payload:

```
{
"crvserver_cachepolicy_binding":{
<b>"name":<String_value>,
</b>"policyname":<String_value>,
"targetvserver":<String_value>,
"priority":<Double_value>,
"gotopriorityexpression":<String_value>,
"bindpoint":<String_value>,
"invoke":<Boolean_value>,
"labeltype":<String_value>,
"labelname":<String_value>
}}
```

Response:

HTTP Status Code on Success: 201 Created HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error

delete

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/crvserver_cachepolicy_binding/name_value<String>

Query-parameters:

args

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/crvserver_cachepolicy_binding/name_value<String>?**args=policyname:<String_value>,bindpoint:<String_value>,priority:<Double_value>**

HTTP Method:DELETE

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue>

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error

get

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/crvserver_cachepolicy_binding/name_value<String>

Query-parameters:

filter

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/crvserver_cachepolicy_binding/name_value<String>?**filter=property-name1:property-value1,property-name2:property-value2**

Use this query-parameter to get the filtered set of crvserver_cachepolicy_binding resources configured on Citrix ADC. Filtering can be done on any of the properties of the resource.

pagination

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/crvserver_cachepolicy_binding/name_value<String>?**pagesize=#no;pageno=#no**

Use this query-parameter to get the crvserver_cachepolicy_binding resources in chunks.

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "crvserver_cachepolicy_binding": [ {

"priority":<Double_value>,

"bindpoint":<String_value>,

"policyname":<String_value>,

"labelname":<String_value>,

"name":<String_value>,

"targetvserver":<String_value>,

"gotopriorityexpression":<String_value>,

"invoke":<Boolean_value>,

"labeltype":<String_value>

}]}
```

get (all)

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/crvserver_cachepolicy_binding

Query-parameters:

bulkbindings

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/crvserver_cachepolicy_binding?**bulkbindings=yes**

NITRO allows you to fetch bindings in bulk.

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "crvserver_cachepolicy_binding": [ {

"priority":<Double_value>,

"bindpoint":<String_value>,

"policyname":<String_value>,

"labelname":<String_value>,

"name":<String_value>,

"targetvserver":<String_value>,

"gotopriorityexpression":<String_value>,

"invoke":<Boolean_value>,

"labeltype":<String_value>

}]}
```

count

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/crvserver_cachepolicy_binding/name_value<String>?**count=yes**

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{"crvserver_cachepolicy_binding": [ { "__count": "#no"} ] }
```

crvserver_cmppolicy_binding

Binding object showing the cmppolicy that can be bound to crvserver.

Properties

(click to see Operations)

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
priority	<Double>	Read-write	The priority for the policy.
bindpoint	<String>	Read-write	The bindpoint to which the policy is bound.

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
			Possible values = REQUEST, RESPONSE, ICA_REQUEST
policyname	<String>	Read-write	Policies bound to this vserver.
labelname	<String>	Read-write	Name of the label invoked.
name	<String>	Read-write	Name of the cache redirection virtual server to which to bind the cache redirection policy. Minimum length = 1
targetvserver	<String>	Read-write	Name of the virtual server to which content is forwarded. Applicable only if the policy is a map policy and the cache redirection virtual server is of type REVERSE. Minimum length = 1
gotopriorityexpression	<String>	Read-write	Expression specifying the priority of the next policy which will get evaluated if the current policy rule evaluates to TRUE.
invoke	<Boolean>	Read-write	Invoke flag.
labeltype	<String>	Read-write	The invocation type.

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
			Possible values = reqvserver, resvserver, policylabel
sc	<String>	Read-only	The state of SureConnect the specified virtual server. Possible values = ON, OFF
inherited	<String>	Read-only	On State describes that policy bound is inherited from global binding. Possible values = ON, OFF
__count	<Double>	Read-only	count parameter

Operations

(click to see Properties)

ADD:| DELETE:| GET| GET (ALL)| COUNT

Some options that you can use for each operations:

- **Getting warnings in response:**NITRO allows you to get warnings in an operation by specifying the "warning" query parameter as "yes". For example, to get warnings while connecting to the Citrix ADC appliance, the URL is as follows:

  http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/login?warning=yes

  If any, the warnings are displayed in the response payload with the HTTP code "209 X-NITRO-WARNING".
- **Authenticated access for individual NITRO operations:**NITRO allows you to logon to the Citrix ADC appliance to perform individual operations. You can use this option instead of

creating a NITRO session (using the login object) and then using that session to perform all operations,

To do this, you must specify the username and password in the request header of the NITRO request as follows:

X-NITRO-USER:<username>

X-NITRO-PASS:<password>

Note: In such cases, make sure that the request header DOES not include the following:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue>

`*Note:`*

Mandatory parameters are marked in redand placeholder content is marked in <green>.

add

URL:http://<Citrix-ADC-IP-address(NSIP)/nitro/v1/config/crvserver_cmppolicy_binding

HTTP Method:PUT

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Content-Type:application/json

Request Payload:

```
{

"crvserver_cmppolicy_binding":{

<b>"name":<String_value>,

</b>"policyname":<String_value>,

"targetvserver":<String_value>,

"priority":<Double_value>,

"gotopriorityexpression":<String_value>,
```

```
"bindpoint":<String_value>,

"invoke":<Boolean_value>,

"labeltype":<String_value>,

"labelname":<String_value>

}}
```

Response:

HTTP Status Code on Success: 201 Created HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error

delete

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/crvserver_cmppolicy_binding/name_value<String>

Query-parameters:

args

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/crvserver_cmppolicy_binding/name_value<String>?**args=policyname:<String_value>,bindpoint:<String_value>,priority:<Double_value>**

HTTP Method:DELETE

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue>

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error

get

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/crvserver_cmppolicy_binding/name_value<String>

Query-parameters:

filter

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/crvserver_cmppolicy_binding/name_value<String>?**filter=property-name1:property-value1,property-name2:property-value2**

Use this query-parameter to get the filtered set of crvserver_cmppolicy_binding resources configured on Citrix ADC. Filtering can be done on any of the properties of the resource.

pagination

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/crvserver_cmppolicy_binding/name_value<String>?**pagesize=#no;pageno=#no**

Use this query-parameter to get the crvserver_cmppolicy_binding resources in chunks.

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "crvserver_cmppolicy_binding": [ {

"priority":<Double_value>,

"bindpoint":<String_value>,
```

```
"policyname":<String_value>,

"labelname":<String_value>,

"name":<String_value>,

"targetvserver":<String_value>,

"gotopriorityexpression":<String_value>,

"invoke":<Boolean_value>,

"labeltype":<String_value>,

"sc":<String_value>,

"inherited":<String_value>

}]}
```

get (all)

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/crvserver_cmppolicy_binding

Query-parameters:

bulkbindings

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/crvserver_cmppolicy_binding?**bulkbindings=yes**

NITRO allows you to fetch bindings in bulk.

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "crvserver_cmppolicy_binding": [ {

"priority":<Double_value>,

"bindpoint":<String_value>,

"policyname":<String_value>,

"labelname":<String_value>,

"name":<String_value>,

"targetvserver":<String_value>,

"gotopriorityexpression":<String_value>,

"invoke":<Boolean_value>,

"labeltype":<String_value>,

"sc":<String_value>,

"inherited":<String_value>

}]}
```

count

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/crvserver_cmppolicy_binding/name_value<String>?**count=yes**

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{"crvserver_cmppolicy_binding": [ { "__count": "#no"} ] }
```

crvserver_crpolicy_binding

Binding object showing the crpolicy that can be bound to crvserver.

Properties

(click to see Operations)

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
priority	<Double>	Read-write	The priority for the policy.
bindpoint	<String>	Read-write	The bindpoint to which the policy is bound. Possible values = REQUEST, RESPONSE, ICA_REQUEST
policyname	<String>	Read-write	Policies bound to this vserver.

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
labelname	<String>	Read-write	Name of the label invoked.
name	<String>	Read-write	Name of the cache redirection virtual server to which to bind the cache redirection policy. Minimum length = 1
targetvserver	<String>	Read-write	Name of the virtual server to which content is forwarded. Applicable only if the policy is a map policy and the cache redirection virtual server is of type REVERSE. Minimum length = 1
gotopriorityexpression	<String>	Read-write	Expression specifying the priority of the next policy which will get evaluated if the current policy rule evaluates to TRUE.
invoke	<Boolean>	Read-write	Invoke flag.
labeltype	<String>	Read-write	The invocation type. Possible values = reqvserver, resvserver, policylabel
hits	<Double>	Read-only	Number of hits.
pipolicyhits	<Double>	Read-only	Number of hits.

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
__count	<Double>	Read-only	count parameter

Operations

(click to see Properties)

ADD:| DELETE:| GET| GET (ALL)| COUNT

Some options that you can use for each operations:

- **Getting warnings in response:**NITRO allows you to get warnings in an operation by specifying the "warning" query parameter as "yes". For example, to get warnings while connecting to the Citrix ADC appliance, the URL is as follows:

  http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/login?warning=yes

  If any, the warnings are displayed in the response payload with the HTTP code "209 X-NITRO-WARNING".
- **Authenticated access for individual NITRO operations:**NITRO allows you to logon to the Citrix ADC appliance to perform individual operations. You can use this option instead of creating a NITRO session (using the login object) and then using that session to perform all operations,

  To do this, you must specify the username and password in the request header of the NITRO request as follows:

  X-NITRO-USER:<username>

  X-NITRO-PASS:<password>

  Note: In such cases, make sure that the request header DOES not include the following:

  Cookie:NITRO_AUTH_TOKEN=<tokenvalue>

`*Note:`*

Mandatory parameters are marked in redand placeholder content is marked in <green>.

add

URL:http://<Citrix-ADC-IP-address(NSIP)/nitro/v1/config/crvserver_crpolicy_binding

HTTP Method:PUT

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Content-Type:application/json

Request Payload:

```
{
"crvserver_crpolicy_binding":{
<b>"name":<String_value>,
</b>"policyname":<String_value>,
"targetvserver":<String_value>,
"priority":<Double_value>,
"gotopriorityexpression":<String_value>,
"bindpoint":<String_value>,
"invoke":<Boolean_value>,
"labeltype":<String_value>,
"labelname":<String_value>
}}
```

Response:

HTTP Status Code on Success: 201 Created HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error

delete

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/crvserver_crpolicy_binding/name_value<String>

Query-parameters:

args

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/crvserver_crpolicy_binding/name_value<String>?**args=policyname:<String_value>,bindpoint:<String_value>,priority:<Double_value>**

HTTP Method:DELETE

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue>

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error

get

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/crvserver_crpolicy_binding/name_value<String>

Query-parameters:

filter

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/crvserver_crpolicy_binding/name_value<String>?**filter=property-name1:property-value1,property-name2:property-value2**

Use this query-parameter to get the filtered set of crvserver_crpolicy_binding resources configured on Citrix ADC. Filtering can be done on any of the properties of the resource.

pagination

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/crvserver_crpolicy_binding/name_value<String>?**pagesize=#no;pageno=#no**

Use this query-parameter to get the crvserver_crpolicy_binding resources in chunks.

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "crvserver_crpolicy_binding": [ {

"priority":<Double_value>,

"bindpoint":<String_value>,

"policyname":<String_value>,

"labelname":<String_value>,

"name":<String_value>,

"targetvserver":<String_value>,

"gotopriorityexpression":<String_value>,

"invoke":<Boolean_value>,

"labeltype":<String_value>,

"hits":<Double_value>,

"pipolicyhits":<Double_value>

}]}
```

get (all)

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/crvserver_crpolicy_binding

Query-parameters:

bulkbindings

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/crvserver_crpolicy_binding?**bulkbindings=yes**

NITRO allows you to fetch bindings in bulk.

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "crvserver_crpolicy_binding": [ {

"priority":<Double_value>,

"bindpoint":<String_value>,

"policyname":<String_value>,

"labelname":<String_value>,

"name":<String_value>,

"targetvserver":<String_value>,

"gotopriorityexpression":<String_value>,

"invoke":<Boolean_value>,
```

```
"labeltype":<String_value>,

"hits":<Double_value>,

"pipolicyhits":<Double_value>

}]}
```

count

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/crvserver_crpolicy_binding/name_value<String>?**count=yes**

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{"crvserver_crpolicy_binding": [ { "__count": "#no"} ] }
```

crvserver_cspolicy_binding

Binding object showing the cspolicy that can be bound to crvserver.

Properties

(click to see Operations)

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
priority	<Double>	Read-write	The priority for the policy.
bindpoint	<String>	Read-write	The bindpoint to which the policy is bound. Possible values = REQUEST, RESPONSE, ICA_REQUEST
policyname	<String>	Read-write	Policies bound to this vserver.
labelname	<String>	Read-write	Name of the label invoked.
name	<String>	Read-write	Name of the cache redirection virtual server to which to bind the cache redirection policy. Minimum length = 1
targetvserver	<String>	Read-write	The CSW target server names.
gotopriorityexpression	<String>	Read-write	Expression specifying the priority of the next policy which will get evaluated if the current policy rule evaluates to TRUE.
invoke	<Boolean>	Read-write	Invoke flag.
labeltype	<String>	Read-write	The invocation type. Possible values = reqvserver,

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
			resvserver, policylabel
hits	<Double>	Read-only	Number of hits.
pipolicyhits	<Double>	Read-only	Number of hits.
__count	<Double>	Read-only	count parameter

Operations

(click to see Properties)

ADD:| DELETE:| GET| GET (ALL)| COUNT

Some options that you can use for each operations:

- **Getting warnings in response:**NITRO allows you to get warnings in an operation by specifying the "warning" query parameter as "yes". For example, to get warnings while connecting to the Citrix ADC appliance, the URL is as follows:

  http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/login?warning=yes

  If any, the warnings are displayed in the response payload with the HTTP code "209 X-NITRO-WARNING".

- **Authenticated access for individual NITRO operations:**NITRO allows you to logon to the Citrix ADC appliance to perform individual operations. You can use this option instead of creating a NITRO session (using the login object) and then using that session to perform all operations,

  To do this, you must specify the username and password in the request header of the NITRO request as follows:

  X-NITRO-USER:<username>

  X-NITRO-PASS:<password>

  Note: In such cases, make sure that the request header DOES not include the following:

  Cookie:NITRO_AUTH_TOKEN=<tokenvalue>

`*Note:` *

Mandatory parameters are marked in redand placeholder content is marked in <green>.

add

URL:http://<Citrix-ADC-IP-address(NSIP)/nitro/v1/config/crvserver_cspolicy_binding

HTTP Method:PUT

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Content-Type:application/json

Request Payload:

```
{
"crvserver_cspolicy_binding":{
<b>"name":<String_value>,
</b>"policyname":<String_value>,
"targetvserver":<String_value>,
"priority":<Double_value>,
"gotopriorityexpression":<String_value>,
"bindpoint":<String_value>,
"invoke":<Boolean_value>,
"labeltype":<String_value>,
"labelname":<String_value>
}}
```

Response:

HTTP Status Code on Success: 201 Created HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error

delete

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/crvserver_cspolicy_binding/name_value<String>

Query-parameters:

args

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/crvserver_cspolicy_binding/name_value<String>?**args=policyname:<String_value>,bindpoint:<String_value>,priority:<Double_value>**

HTTP Method:DELETE

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue>

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error

get

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/crvserver_cspolicy_binding/name_value<String>

Query-parameters:

filter

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/crvserver_cspolicy_binding/name_value<String>?**filter=property-name1:property-value1,property-name2:property-value2**

Use this query-parameter to get the filtered set of crvserver_cspolicy_binding resources configured on Citrix ADC. Filtering can be done on any of the properties of the resource.

pagination

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/crvserver_cspolicy_binding/name_value<String>?**pagesize=#no;pageno=#no**

Use this query-parameter to get the crvserver_cspolicy_binding resources in chunks.

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "crvserver_cspolicy_binding": [ {

"priority":<Double_value>,

"bindpoint":<String_value>,

"policyname":<String_value>,

"labelname":<String_value>,

"name":<String_value>,

"targetvserver":<String_value>,

"gotopriorityexpression":<String_value>,

"invoke":<Boolean_value>,

"labeltype":<String_value>,

"hits":<Double_value>,
```

```
"pipolicyhits":<Double_value>

}]}
```

get (all)

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/crvserver_cspolicy_binding

Query-parameters:

bulkbindings

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/crvserver_cspolicy_binding?**bulkbindings=yes**

NITRO allows you to fetch bindings in bulk.

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "crvserver_cspolicy_binding": [ {

"priority":<Double_value>,

"bindpoint":<String_value>,

"policyname":<String_value>,
```

```
"labelname":<String_value>,

"name":<String_value>,

"targetvserver":<String_value>,

"gotopriorityexpression":<String_value>,

"invoke":<Boolean_value>,

"labeltype":<String_value>,

"hits":<Double_value>,

"pipolicyhits":<Double_value>

}]}
```

count

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/crvserver_cspolicy_binding/name_value<String>?**count=yes**

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{"crvserver_cspolicy_binding": [ { "__count": "#no"} ] }
```

crvserver_feopolicy_binding

Binding object showing the feopolicy that can be bound to crvserver.

Properties

(click to see Operations)

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
priority	<Double>	Read-write	The priority for the policy.
bindpoint	<String>	Read-write	The bindpoint to which the policy is bound. Possible values = REQUEST, RESPONSE, ICA_REQUEST
policyname	<String>	Read-write	Policies bound to this vserver.
labelname	<String>	Read-write	Name of the label to be invoked.
name	<String>	Read-write	Name of the cache redirection virtual server to which to bind the cache redirection policy. Minimum length = 1
targetvserver	<String>	Read-write	Name of the virtual server to which content is forwarded. Applicable only if the policy is a map policy and the cache redirection virtual

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
			server is of type REVERSE. Minimum length = 1
gotopriorityexpression	<String>	Read-write	Expression specifying the priority of the next policy which will get evaluated if the current policy rule evaluates to TRUE.
invoke	<Boolean>	Read-write	Invoke a policy label if this policy's rule evaluates to TRUE (valid only for default-syntax policies such as application firewall, transform, integrated cache, rewrite, responder, and content switching).
labeltype	<String>	Read-write	Type of label to be invoked. Possible values = reqvserver, resvserver, policylabel
__count	<Double>	Read-write	count parameter

Operations

(click to see Properties)

ADD:| DELETE:| GET| GET (ALL)| COUNT

Some options that you can use for each operations:

- **Getting warnings in response:**NITRO allows you to get warnings in an operation by specifying the "warning" query parameter as "yes". For example, to get warnings while connecting to the Citrix ADC appliance, the URL is as follows:

  http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/login?warning=yes

  If any, the warnings are displayed in the response payload with the HTTP code "209 X-NITRO-WARNING".

- **Authenticated access for individual NITRO operations:**NITRO allows you to logon to the Citrix ADC appliance to perform individual operations. You can use this option instead of creating a NITRO session (using the login object) and then using that session to perform all operations,

  To do this, you must specify the username and password in the request header of the NITRO request as follows:

  X-NITRO-USER:<username>

  X-NITRO-PASS:<password>

  Note: In such cases, make sure that the request header DOES not include the following:

  Cookie:NITRO_AUTH_TOKEN=<tokenvalue>

`*Note:`*

Mandatory parameters are marked in redand placeholder content is marked in <green>.

add

URL:http://<Citrix-ADC-IP-address(NSIP)/nitro/v1/config/crvserver_feopolicy_binding

HTTP Method:PUT

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Content-Type:application/json

Request Payload:

```
{
```

```
"crvserver_feopolicy_binding":{

<b>"name":<String_value>,

</b>"policyname":<String_value>,

"targetvserver":<String_value>,

"priority":<Double_value>,

"gotopriorityexpression":<String_value>,

"bindpoint":<String_value>,

"invoke":<Boolean_value>,

"labeltype":<String_value>,

"labelname":<String_value>

}}
```

Response:

HTTP Status Code on Success: 201 Created HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error

delete

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/crvserver_feopolicy_binding/name_value<String>

Query-parameters:

args

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/crvserver_feopolicy_binding/name_value<String>?
args=policyname:<String_value>,bindpoint:<String_value>,priority:<Double_value>

HTTP Method:DELETE

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue>

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error

get

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/crvserver_feopolicy_binding/name_value<String>

Query-parameters:

filter

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/crvserver_feopolicy_binding/name_value<String>?**filter=property-name1:property-value1,property-name2:property-value2**

Use this query-parameter to get the filtered set of crvserver_feopolicy_binding resources configured on Citrix ADC. Filtering can be done on any of the properties of the resource.

pagination

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/crvserver_feopolicy_binding/name_value<String>?**pagesize=#no;pageno=#no**

Use this query-parameter to get the crvserver_feopolicy_binding resources in chunks.

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "crvserver_feopolicy_binding": [ {

"priority":<Double_value>,

"bindpoint":<String_value>,

"policyname":<String_value>,

"labelname":<String_value>,

"name":<String_value>,

"targetvserver":<String_value>,

"gotopriorityexpression":<String_value>,

"invoke":<Boolean_value>,

"labeltype":<String_value>

}]}
```

get (all)

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/crvserver_feopolicy_binding

Query-parameters:

bulkbindings

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/crvserver_feopolicy_binding?**bulkbindings=yes**

NITRO allows you to fetch bindings in bulk.

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "crvserver_feopolicy_binding": [ {

"priority":<Double_value>,

"bindpoint":<String_value>,

"policyname":<String_value>,

"labelname":<String_value>,

"name":<String_value>,

"targetvserver":<String_value>,

"gotopriorityexpression":<String_value>,

"invoke":<Boolean_value>,

"labeltype":<String_value>

}]}
```

count

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/crvserver_feopolicy_binding/name_value<String>?**count=yes**

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{"crvserver_feopolicy_binding": [ { "__count": "#no"} ] }
```

crvserver_filterpolicy_binding

Binding object showing the filterpolicy that can be bound to crvserver.

Properties

(click to see Operations)

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
priority	<Double>	Read-write	The priority for the policy.
bindpoint	<String>	Read-write	For a rewrite policy, the bind point to which to bind the policy. Note: This parameter applies only to rewrite policies, because content switching policies are evaluated only at request time.

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
			Possible values = REQUEST, RESPONSE, ICA_REQUEST
policyname	<String>	Read-write	Policies bound to this vserver.
labelname	<String>	Read-write	Name of the label to be invoked.
name	<String>	Read-write	Name of the cache redirection virtual server to which to bind the cache redirection policy. Minimum length = 1
targetvserver	<String>	Read-write	Name of the virtual server to which content is forwarded. Applicable only if the policy is a map policy and the cache redirection virtual server is of type REVERSE. Minimum length = 1
gotopriorityexpression	<String>	Read-write	Expression or other value specifying the next policy to be evaluated if the current policy evaluates to TRUE. Specify one of the following values: * NEXT - Evaluate the policy with the next higher priority number. * END - End

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
			policy evaluation. * USE_INVOCATION_RESULT - Applicable if this policy invokes another policy label. If the final goto in the invoked policy label has a value of END, the evaluation stops. If the final goto is anything other than END, the current policy label performs a NEXT. * An expression that evaluates to a number. If you specify an expression, the number to which it evaluates determines the next policy to evaluate, as follows: * If the expression evaluates to a higher numbered priority, the policy with that priority is evaluated next. * If the expression evaluates to the priority of the current policy, the policy with the next higher numbered priority is evaluated next. * If the expression evaluates to a priority number that is numerically higher than the

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
			highest numbered priority, policy evaluation ends. An UNDEF event is triggered if: * The expression is invalid. * The expression evaluates to a priority number that is numerically lower than the current policy's priority. * The expression evaluates to a priority number that is between the current policy's priority number (say, 30) and the highest priority number (say, 100), b ut does not match any configured priority number (for example, the expression evaluates to the number 85). This example assumes that the priority number incr ements by 10 for every successive policy, and therefore a priority number of 85 does not exist in the policy label.
invoke	<Boolean>	Read-write	Invoke a policy label if this policy's rule evaluates to TRUE (valid only for default-syntax

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
			policies such as application firewall, transform, integrated cache, rewrite, responder, and content switching).
labeltype	<String>	Read-write	Type of label to be invoked. Possible values = reqvserver, resvserver, policylabel
inherited	<String>	Read-only	On State describes that policy bound is inherited from global binding. Possible values = ON, OFF
__count	<Double>	Read-only	count parameter

Operations

(click to see Properties)

ADD:| DELETE:| GET| GET (ALL)| COUNT

Some options that you can use for each operations:

- **Getting warnings in response:**NITRO allows you to get warnings in an operation by specifying the "warning" query parameter as "yes". For example, to get warnings while connecting to the Citrix ADC appliance, the URL is as follows:

  http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/login?warning=yes

  If any, the warnings are displayed in the response payload with the HTTP code "209 X-NITRO-WARNING".

- **Authenticated access for individual NITRO operations:**NITRO allows you to logon to the Citrix ADC appliance to perform individual operations. You can use this option instead of creating a NITRO session (using the login object) and then using that session to perform all operations,

  To do this, you must specify the username and password in the request header of the NITRO request as follows:

  X-NITRO-USER:<username>

  X-NITRO-PASS:<password>

  Note: In such cases, make sure that the request header DOES not include the following:

  Cookie:NITRO_AUTH_TOKEN=<tokenvalue>

`*Note:`*

Mandatory parameters are marked in redand placeholder content is marked in <green>.

add

URL:http://<Citrix-ADC-IP-address(NSIP)/nitro/v1/config/crvserver_filterpolicy_binding

HTTP Method:PUT

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Content-Type:application/json

Request Payload:

```
{

"crvserver_filterpolicy_binding":{

<b>"name":<String_value>,

</b>"policyname":<String_value>,

"targetvserver":<String_value>,

"priority":<Double_value>,
```

```
"gotopriorityexpression":<String_value>,

"bindpoint":<String_value>,

"invoke":<Boolean_value>,

"labeltype":<String_value>,

"labelname":<String_value>

}}
```

Response:

HTTP Status Code on Success: 201 Created HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error

delete

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/crvserver_filterpolicy_binding/name_value<String>

Query-parameters:

args

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/crvserver_filterpolicy_binding/name_value<String>?
args=policyname:<String_value>,bindpoint:<String_value>,priority:<Double_value>

HTTP Method:DELETE

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue>

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error

get

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/crvserver_filterpolicy_binding/name_value<String>

Query-parameters:

filter

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/crvserver_filterpolicy_binding/name_value<String>?**filter=property-name1:property-value1,property-name2:property-value2**

Use this query-parameter to get the filtered set of crvserver_filterpolicy_binding resources configured on Citrix ADC. Filtering can be done on any of the properties of the resource.

pagination

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/crvserver_filterpolicy_binding/name_value<String>?**pagesize=#no;pageno=#no**

Use this query-parameter to get the crvserver_filterpolicy_binding resources in chunks.

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "crvserver_filterpolicy_binding": [ {

"priority":<Double_value>,

"bindpoint":<String_value>,
```

```
"policyname":<String_value>,

"labelname":<String_value>,

"name":<String_value>,

"targetvserver":<String_value>,

"gotopriorityexpression":<String_value>,

"invoke":<Boolean_value>,

"labeltype":<String_value>,

"inherited":<String_value>

}]}
```

get (all)

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/crvserver_filterpolicy_binding

Query-parameters:

bulkbindings

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/crvserver_filterpolicy_binding?**bulkbindings=yes**

NITRO allows you to fetch bindings in bulk.

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "crvserver_filterpolicy_binding": [ {

"priority":<Double_value>,

"bindpoint":<String_value>,

"policyname":<String_value>,

"labelname":<String_value>,

"name":<String_value>,

"targetvserver":<String_value>,

"gotopriorityexpression":<String_value>,

"invoke":<Boolean_value>,

"labeltype":<String_value>,

"inherited":<String_value>

}]}
```

count

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/crvserver_filterpolicy_binding/name_value<String>?**count=yes**

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{"crvserver_filterpolicy_binding": [ { "__count": "#no"} ] }
```

crvserver_icapolicy_binding

Binding object showing the icapolicy that can be bound to crvserver.

Properties

(click to see Operations)

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
priority	<Double>	Read-write	The priority for the policy.
bindpoint	<String>	Read-write	For a rewrite policy, the bind point to which to bind the policy. Note: This parameter applies only to rewrite policies, because content switching policies are evaluated only at request time. Possible values = REQUEST, RESPONSE, ICA_REQUEST

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
policyname	<String>	Read-write	Policies bound to this vserver.
labelname	<String>	Read-write	Name of the label to be invoked.
name	<String>	Read-write	Name of the cache redirection virtual server to which to bind the cache redirection policy. Minimum length = 1
targetvserver	<String>	Read-write	Name of the virtual server to which content is forwarded. Applicable only if the policy is a map policy and the cache redirection virtual server is of type REVERSE. Minimum length = 1
gotopriorityexpression	<String>	Read-write	Expression specifying the priority of the next policy which will get evaluated if the current policy rule evaluates to TRUE.
invoke	<Boolean>	Read-write	Invoke a policy label if this policy's rule evaluates to TRUE (valid only for default-syntax policies such as application firewall, transform, integrated cache, rewrite,

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
			responder, and content switching).
labeltype	<String>	Read-write	Type of label to be invoked. Possible values = reqvserver, resvserver, policylabel
__count	<Double>	Read-write	count parameter

Operations

(click to see Properties)

ADD:| DELETE:| GET| GET (ALL)| COUNT

Some options that you can use for each operations:

- **Getting warnings in response:**NITRO allows you to get warnings in an operation by specifying the "warning" query parameter as "yes". For example, to get warnings while connecting to the Citrix ADC appliance, the URL is as follows:

  http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/login?warning=yes

  If any, the warnings are displayed in the response payload with the HTTP code "209 X-NITRO-WARNING".

- **Authenticated access for individual NITRO operations:**NITRO allows you to logon to the Citrix ADC appliance to perform individual operations. You can use this option instead of creating a NITRO session (using the login object) and then using that session to perform all operations,

  To do this, you must specify the username and password in the request header of the NITRO request as follows:

  X-NITRO-USER:<username>

  X-NITRO-PASS:<password>

  Note: In such cases, make sure that the request header DOES not include the following:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue>

`*Note:` *

Mandatory parameters are marked in redand placeholder content is marked in <green>.

add

URL:http://<Citrix-ADC-IP-address(NSIP)/nitro/v1/config/crvserver_icapolicy_binding

HTTP Method:PUT

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Content-Type:application/json

Request Payload:

```
{

"crvserver_icapolicy_binding":{

<b>"name":<String_value>,

</b>"policyname":<String_value>,

"targetvserver":<String_value>,

"priority":<Double_value>,

"gotopriorityexpression":<String_value>,

"bindpoint":<String_value>,

"invoke":<Boolean_value>,

"labeltype":<String_value>,

"labelname":<String_value>

}}
```

Response:

HTTP Status Code on Success: 201 Created HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error

delete

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/crvserver_icapolicy_binding/name_value<String>

Query-parameters:

args

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/crvserver_icapolicy_binding/name_value<String>?**args=policyname:<String_value>,bindpoint:<String_value>,priority:<Double_value>**

HTTP Method:DELETE

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue>

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error

get

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/crvserver_icapolicy_binding/name_value<String>

Query-parameters:

filter

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/crvserver_icapolicy_binding/name_value<String>?**filter=property-name1:property-value1,property-name2:property-value2**

Use this query-parameter to get the filtered set of crvserver_icapolicy_binding resources configured on Citrix ADC. Filtering can be done on any of the properties of the resource.

pagination

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/crvserver_icapolicy_binding/name_value<String>?**pagesize=#no;pageno=#no**

Use this query-parameter to get the crvserver_icapolicy_binding resources in chunks.

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "crvserver_icapolicy_binding": [ {

"priority":<Double_value>,

"bindpoint":<String_value>,

"policyname":<String_value>,

"labelname":<String_value>,

"name":<String_value>,

"targetvserver":<String_value>,

"gotopriorityexpression":<String_value>,

"invoke":<Boolean_value>,

"labeltype":<String_value>
```

```
}]}
```

get (all)

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/crvserver_icapolicy_binding

Query-parameters:

bulkbindings

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/crvserver_icapolicy_binding?**bulkbindings=yes**

NITRO allows you to fetch bindings in bulk.

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "crvserver_icapolicy_binding": [ {

"priority":<Double_value>,

"bindpoint":<String_value>,

"policyname":<String_value>,

"labelname":<String_value>,
```

```
"name":<String_value>,

"targetvserver":<String_value>,

"gotopriorityexpression":<String_value>,

"invoke":<Boolean_value>,

"labeltype":<String_value>

}]}
```

count

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/crvserver_icapolicy_binding/name_value<String>?**count=yes**

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{"crvserver_icapolicy_binding": [ { "__count": "#no"} ] }
```

crvserver_lbvserver_binding

Binding object showing the lbvserver that can be bound to crvserver.

Properties

(click to see Operations)

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
name	<String>	Read-write	Name of the cache redirection virtual server to which to bind the cache redirection policy. Minimum length = 1
lbvserver	<String>	Read-write	The Default target server name. Minimum length = 1
hits	<Double>	Read-only	Number of hits.
__count	<Double>	Read-only	count parameter

Operations

(click to see Properties)

ADD:| DELETE:| GET| GET (ALL)| COUNT

Some options that you can use for each operations:

- **Getting warnings in response:**NITRO allows you to get warnings in an operation by specifying the "warning" query parameter as "yes". For example, to get warnings while connecting to the Citrix ADC appliance, the URL is as follows:

  http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/login?warning=yes

  If any, the warnings are displayed in the response payload with the HTTP code "209 X-NITRO-WARNING".
- **Authenticated access for individual NITRO operations:**NITRO allows you to logon to the Citrix ADC appliance to perform individual operations. You can use this option instead of creating a NITRO session (using the login object) and then using that session to perform all operations,

To do this, you must specify the username and password in the request header of the NITRO request as follows:

X-NITRO-USER:<username>

X-NITRO-PASS:<password>

Note: In such cases, make sure that the request header DOES not include the following:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue>

`*Note:`*

Mandatory parameters are marked in redand placeholder content is marked in <green>.

add

URL:http://<Citrix-ADC-IP-address(NSIP)/nitro/v1/config/crvserver_lbvserver_binding

HTTP Method:PUT

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Content-Type:application/json

Request Payload:

```
{

"crvserver_lbvserver_binding":{

<b>"name":<String_value>,

</b>"lbvserver":<String_value>

}}
```

Response:

HTTP Status Code on Success: 201 Created HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error

delete

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/crvserver_lbvserver_binding/name_value<String>

Query-parameters:

args

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/crvserver_lbvserver_binding/name_value<String>?**args=lbvserver:<String_value>**

HTTP Method:DELETE

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue>

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error

get

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/crvserver_lbvserver_binding/name_value<String>

Query-parameters:

filter

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/crvserver_lbvserver_binding/name_value<String>?**filter=property-name1:property-value1,property-name2:property-value2**

Use this query-parameter to get the filtered set of crvserver_lbvserver_binding resources configured on Citrix ADC. Filtering can be done on any of the properties of the resource.

pagination

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/crvserver_lbvserver_binding/name_value<String>?**pagesize=#no;pageno=#no**

Use this query-parameter to get the crvserver_lbvserver_binding resources in chunks.

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "crvserver_lbvserver_binding": [ {

"name":<String_value>,

"lbvserver":<String_value>,

"hits":<Double_value>

}]}
```

get (all)

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/crvserver_lbvserver_binding

Query-parameters:

bulkbindings

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/crvserver_lbvserver_binding?**bulkbindings=yes**

NITRO allows you to fetch bindings in bulk.

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "crvserver_lbvserver_binding": [ {

"name":<String_value>,

"lbvserver":<String_value>,

"hits":<Double_value>

}]}
```

count

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/crvserver_lbvserver_binding/name_value<String>?**count=yes**

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{"crvserver_lbvserver_binding": [ { "__count": "#no"} ] }
```

crvserver_policymap_binding

Binding object showing the policymap that can be bound to crvserver.

Properties

(click to see Operations)

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
priority	<Double>	Read-write	An unsigned integer that determines the priority of the policy relative to other policies bound to this cache redirection virtual server. The lower the value, higher the priority. Note: This option is available only when binding content switching, filtering, and compression policies to a cache redirection virtual server.
bindpoint	<String>	Read-write	For a rewrite policy, the bind point to which to bind the policy. Note: This parameter applies only to rewrite policies, because content switching

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
			policies are evaluated only at request time. Possible values = REQUEST, RESPONSE, ICA_REQUEST
policyname	<String>	Read-write	Policies bound to this vserver.
labelname	<String>	Read-write	Name of the label to be invoked.
name	<String>	Read-write	Name of the cache redirection virtual server to which to bind the cache redirection policy. Minimum length = 1
targetvserver	<String>	Read-write	The CSW target server names.
gotopriorityexpression	<String>	Read-write	Expression or other value specifying the next policy to be evaluated if the current policy evaluates to TRUE. Specify one of the following values: * NEXT - Evaluate the policy with the next higher priority number. * END - End policy evaluation. * USE_INVOCATION_RESULT - Applicable if this policy invokes another policy label.

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
			If the final goto in the invoked policy label has a value of END, the evaluation stops. If the final goto is anything other than END, the current policy label performs a NEXT. * An expression that evaluates to a number. If you specify an expression, the number to which it evaluates determines the next policy to evaluate, as follows: * If the expression evaluates to a higher numbered priority, the policy with that priority is evaluated next. * If the expression evaluates to the priority of the current policy, the policy with the next higher numbered priority is evaluated next. * If the expression evaluates to a priority number that is numerically higher than the highest numbered priority, policy evaluation ends. An UNDEF event is triggered if: * The

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
			expression is invalid. * The expression evaluates to a priority number that is numerically lower than the current policy's priority. * The expression evaluates to a priority number that is between the current policy's priority number (say, 30) and the highest priority number (say, 100), b ut does not match any configured priority number (for example, the expression evaluates to the number 85). This example assumes that the priority number incr ements by 10 for every successive policy, and therefore a priority number of 85 does not exist in the policy label.
invoke	<Boolean>	Read-write	Invoke a policy label if this policy's rule evaluates to TRUE (valid only for default-syntax policies such as application firewall, transform, integrated cache, rewrite,

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
			responder, and content switching).
labeltype	<String>	Read-write	Type of label to be invoked. Possible values = reqvserver, resvserver, policylabel
__count	<Double>	Read-write	count parameter

Operations

(click to see Properties)

ADD:| DELETE:| GET| GET (ALL)| COUNT

Some options that you can use for each operations:

- **Getting warnings in response:**NITRO allows you to get warnings in an operation by specifying the "warning" query parameter as "yes". For example, to get warnings while connecting to the Citrix ADC appliance, the URL is as follows:

  http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/login?warning=yes

  If any, the warnings are displayed in the response payload with the HTTP code "209 X-NITRO-WARNING".

- **Authenticated access for individual NITRO operations:**NITRO allows you to logon to the Citrix ADC appliance to perform individual operations. You can use this option instead of creating a NITRO session (using the login object) and then using that session to perform all operations,

  To do this, you must specify the username and password in the request header of the NITRO request as follows:

  X-NITRO-USER:<username>

  X-NITRO-PASS:<password>

  Note: In such cases, make sure that the request header DOES not include the following:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue>

`*Note:` *

Mandatory parameters are marked in redand placeholder content is marked in <green>.

add

URL:http://<Citrix-ADC-IP-address(NSIP)/nitro/v1/config/crvserver_policymap_binding

HTTP Method:PUT

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Content-Type:application/json

Request Payload:

```
{

"crvserver_policymap_binding":{

<b>"name":<String_value>,

</b>"policyname":<String_value>,

"targetvserver":<String_value>,

"priority":<Double_value>,

"gotopriorityexpression":<String_value>,

"bindpoint":<String_value>,

"invoke":<Boolean_value>,

"labeltype":<String_value>,

"labelname":<String_value>

}}
```

Response:

HTTP Status Code on Success: 201 Created HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error

delete

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/crvserver_policymap_binding/name_value<String>

Query-parameters:

args

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/crvserver_policymap_binding/name_value<String>?**args=policyname:<String_value>,bindpoint:<String_value>,priority:<Double_value>**

HTTP Method:DELETE

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue>

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error

get

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/crvserver_policymap_binding/name_value<String>

Query-parameters:

filter

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/crvserver_policymap_binding/name_value<String>?**filter=property-name1:property-value1,property-name2:property-value2**

Use this query-parameter to get the filtered set of crvserver_policymap_binding resources configured on Citrix ADC. Filtering can be done on any of the properties of the resource.

pagination

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/crvserver_policymap_binding/name_value<String>?**pagesize=#no;pageno=#no**

Use this query-parameter to get the crvserver_policymap_binding resources in chunks.

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "crvserver_policymap_binding": [ {

"priority":<Double_value>,

"bindpoint":<String_value>,

"policyname":<String_value>,

"labelname":<String_value>,

"name":<String_value>,

"targetvserver":<String_value>,

"gotopriorityexpression":<String_value>,

"invoke":<Boolean_value>,

"labeltype":<String_value>
```

```
}]}
```

get (all)

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/crvserver_policymap_binding

Query-parameters:

bulkbindings

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/crvserver_policymap_binding?**bulkbindings=yes**

NITRO allows you to fetch bindings in bulk.

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "crvserver_policymap_binding": [ {

"priority":<Double_value>,

"bindpoint":<String_value>,

"policyname":<String_value>,

"labelname":<String_value>,
```

```
"name":<String_value>,

"targetvserver":<String_value>,

"gotopriorityexpression":<String_value>,

"invoke":<Boolean_value>,

"labeltype":<String_value>

}]}
```

count

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/crvserver_policymap_binding/name_value<String>?**count=yes**

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{"crvserver_policymap_binding": [ { "__count": "#no"} ] }
```

crvserver_responderpolicy_binding

Binding object showing the responderpolicy that can be bound to crvserver.

Properties

(click to see Operations)

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
priority	<Double>	Read-write	The priority for the policy.
bindpoint	<String>	Read-write	For a rewrite policy, the bind point to which to bind the policy. Note: This parameter applies only to rewrite policies, because content switching policies are evaluated only at request time. Possible values = REQUEST, RESPONSE, ICA_REQUEST
policyname	<String>	Read-write	Policies bound to this vserver.
labelname	<String>	Read-write	Name of the label invoked.
name	<String>	Read-write	Name of the cache redirection virtual server to which to bind the cache redirection policy. Minimum length = 1
targetvserver	<String>	Read-write	Name of the virtual server to which content is forwarded. Applicable only if the policy is a map policy

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
			and the cache redirection virtual server is of type REVERSE. Minimum length = 1
gotopriorityexpression	<String>	Read-write	Expression specifying the priority of the next policy which will get evaluated if the current policy rule evaluates to TRUE.
invoke	<Boolean>	Read-write	Invoke flag.
labeltype	<String>	Read-write	The invocation type. Possible values = reqvserver, resvserver, policylabel
__count	<Double>	Read-write	count parameter

Operations

(click to see Properties)

ADD:| DELETE:| GET| GET (ALL)| COUNT

Some options that you can use for each operations:

- **Getting warnings in response:**NITRO allows you to get warnings in an operation by specifying the "warning" query parameter as "yes". For example, to get warnings while connecting to the Citrix ADC appliance, the URL is as follows:

  http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/login?warning=yes

  If any, the warnings are displayed in the response payload with the HTTP code "209 X-NITRO-WARNING".

- **Authenticated access for individual NITRO operations:**NITRO allows you to logon to the Citrix ADC appliance to perform individual operations. You can use this option instead of creating a NITRO session (using the login object) and then using that session to perform all operations,

  To do this, you must specify the username and password in the request header of the NITRO request as follows:

  X-NITRO-USER:<username>

  X-NITRO-PASS:<password>

  Note: In such cases, make sure that the request header DOES not include the following:

  Cookie:NITRO_AUTH_TOKEN=<tokenvalue>

`*Note:`*

Mandatory parameters are marked in redand placeholder content is marked in <green>.

add

URL:http://<Citrix-ADC-IP-address(NSIP)/nitro/v1/config/crvserver_responderpolicy_binding

HTTP Method:PUT

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Content-Type:application/json

Request Payload:

```
{

"crvserver_responderpolicy_binding":{

<b>"name":<String_value>,

</b>"policyname":<String_value>,

"targetvserver":<String_value>,

"priority":<Double_value>,
```

```
"gotopriorityexpression":<String_value>,

"bindpoint":<String_value>,

"invoke":<Boolean_value>,

"labeltype":<String_value>,

"labelname":<String_value>

}}
```

Response:

HTTP Status Code on Success: 201 Created HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error

delete

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/crvserver_responderpolicy_binding/name_value<String>

Query-parameters:

args

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/crvserver_responderpolicy_binding/name_value<String>?
args=policyname:<String_value>,bindpoint:<String_value>,priority:<Double_value>

HTTP Method:DELETE

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue>

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error

get

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/crvserver_responderpolicy_binding/name_value<String>

Query-parameters:

filter

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/crvserver_responderpolicy_binding/name_value<String>?**filter=property-name1:property-value1,property-name2:property-value2**

Use this query-parameter to get the filtered set of crvserver_responderpolicy_binding resources configured on Citrix ADC. Filtering can be done on any of the properties of the resource.

pagination

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/crvserver_responderpolicy_binding/name_value<String>?**pagesize=#no;pageno=#no**

Use this query-parameter to get the crvserver_responderpolicy_binding resources in chunks.

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "crvserver_responderpolicy_binding": [ {

"priority":<Double_value>,

"bindpoint":<String_value>,
```

```
"policyname":<String_value>,

"labelname":<String_value>,

"name":<String_value>,

"targetvserver":<String_value>,

"gotopriorityexpression":<String_value>,

"invoke":<Boolean_value>,

"labeltype":<String_value>

}]}
```

get (all)

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/crvserver_responderpolicy_binding

Query-parameters:

bulkbindings

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/crvserver_responderpolicy_binding?**bulkbindings=yes**

NITRO allows you to fetch bindings in bulk.

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "crvserver_responderpolicy_binding": [ {

"priority":<Double_value>,

"bindpoint":<String_value>,

"policyname":<String_value>,

"labelname":<String_value>,

"name":<String_value>,

"targetvserver":<String_value>,

"gotopriorityexpression":<String_value>,

"invoke":<Boolean_value>,

"labeltype":<String_value>

}]}
```

count

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/crvserver_responderpolicy_binding/name_value<String>?**count=yes**

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{"crvserver_responderpolicy_binding": [ { "__count": "#no"} ] }
```

crvserver_rewritepolicy_binding

Binding object showing the rewritepolicy that can be bound to crvserver.

Properties

(click to see Operations)

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
priority	<Double>	Read-write	The priority for the policy.
bindpoint	<String>	Read-write	The bindpoint to which the policy is bound. Possible values = REQUEST, RESPONSE, ICA_REQUEST
policyname	<String>	Read-write	Policies bound to this vserver.
labelname	<String>	Read-write	Name of the label invoked.
name	<String>	Read-write	Name of the cache redirection virtual server to which to bind the cache redirection policy. Minimum length = 1

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
targetvserver	<String>	Read-write	Name of the virtual server to which content is forwarded. Applicable only if the policy is a map policy and the cache redirection virtual server is of type REVERSE. Minimum length = 1
gotopriorityexpression	<String>	Read-write	Expression specifying the priority of the next policy which will get evaluated if the current policy rule evaluates to TRUE.
invoke	<Boolean>	Read-write	Invoke flag.
labeltype	<String>	Read-write	The invocation type. Possible values = reqvserver, resvserver, policylabel
__count	<Double>	Read-write	count parameter

Operations

(click to see Properties)

ADD:| DELETE:| GET| GET (ALL)| COUNT

Some options that you can use for each operations:

- **Getting warnings in response:**NITRO allows you to get warnings in an operation by specifying the "warning" query parameter as "yes". For example, to get warnings while connecting to the Citrix ADC appliance, the URL is as follows:

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/login?warning=yes

If any, the warnings are displayed in the response payload with the HTTP code "209 X-NITRO-WARNING".

- **Authenticated access for individual NITRO operations:**NITRO allows you to logon to the Citrix ADC appliance to perform individual operations. You can use this option instead of creating a NITRO session (using the login object) and then using that session to perform all operations,

  To do this, you must specify the username and password in the request header of the NITRO request as follows:

  X-NITRO-USER:<username>

  X-NITRO-PASS:<password>

  Note: In such cases, make sure that the request header DOES not include the following:

  Cookie:NITRO_AUTH_TOKEN=<tokenvalue>

`*Note:`*

Mandatory parameters are marked in redand placeholder content is marked in <green>.

add

URL:http://<Citrix-ADC-IP-address(NSIP)/nitro/v1/config/crvserver_rewritepolicy_binding

HTTP Method:PUT

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Content-Type:application/json

Request Payload:

```
{

"crvserver_rewritepolicy_binding":{

<b>"name":<String_value>,

</b>"policyname":<String_value>,
```

```
"targetvserver":<String_value>,

"priority":<Double_value>,

"gotopriorityexpression":<String_value>,

"bindpoint":<String_value>,

"invoke":<Boolean_value>,

"labeltype":<String_value>,

"labelname":<String_value>

}}
```

Response:

HTTP Status Code on Success: 201 Created HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error

delete

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/crvserver_rewritepolicy_binding/name_value<String>

Query-parameters:

args

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/crvserver_rewritepolicy_binding/name_value<String>?
args=policyname:<String_value>,bindpoint:<String_value>,priority:<Double_value>

HTTP Method:DELETE

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue>

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error

get

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/crvserver_rewritepolicy_binding/name_value<String>

Query-parameters:

filter

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/crvserver_rewritepolicy_binding/name_value<String>?**filter=property-name1:property-value1,property-name2:property-value2**

Use this query-parameter to get the filtered set of crvserver_rewritepolicy_binding resources configured on Citrix ADC. Filtering can be done on any of the properties of the resource.

pagination

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/crvserver_rewritepolicy_binding/name_value<String>?**pagesize=#no;pageno=#no**

Use this query-parameter to get the crvserver_rewritepolicy_binding resources in chunks.

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "crvserver_rewritepolicy_binding": [ {
```

```
"priority":<Double_value>,

"bindpoint":<String_value>,

"policyname":<String_value>,

"labelname":<String_value>,

"name":<String_value>,

"targetvserver":<String_value>,

"gotopriorityexpression":<String_value>,

"invoke":<Boolean_value>,

"labeltype":<String_value>

}]}
```

get (all)

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/crvserver_rewritepolicy_binding

Query-parameters:

bulkbindings

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/crvserver_rewritepolicy_binding?**bulkbindings=yes**

NITRO allows you to fetch bindings in bulk.

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "crvserver_rewritepolicy_binding": [ {

"priority":<Double_value>,

"bindpoint":<String_value>,

"policyname":<String_value>,

"labelname":<String_value>,

"name":<String_value>,

"targetvserver":<String_value>,

"gotopriorityexpression":<String_value>,

"invoke":<Boolean_value>,

"labeltype":<String_value>

}]}
```

count

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/crvserver_rewritepolicy_binding/name_value<String>?**count=yes**

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{"crvserver_rewritepolicy_binding": [ { "__count": "#no"} ] }
```

crvserver_spilloverpolicy_binding

Binding object showing the spilloverpolicy that can be bound to crvserver.

Properties

(click to see Operations)

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
priority	<Double>	Read-write	The priority for the policy.
bindpoint	<String>	Read-write	The bindpoint to which the policy is bound. Possible values = REQUEST, RESPONSE, ICA_REQUEST
policyname	<String>	Read-write	Policies bound to this vserver.
labelname	<String>	Read-write	Name of the label to be invoked.

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
name	<String>	Read-write	Name of the cache redirection virtual server to which to bind the cache redirection policy. Minimum length = 1
targetvserver	<String>	Read-write	Name of the virtual server to which content is forwarded. Applicable only if the policy is a map policy and the cache redirection virtual server is of type REVERSE. Minimum length = 1
gotopriorityexpression	<String>	Read-write	Expression specifying the priority of the next policy which will get evaluated if the current policy rule evaluates to TRUE.
invoke	<Boolean>	Read-write	Invoke a policy label if this policy's rule evaluates to TRUE (valid only for default-syntax policies such as application firewall, transform, integrated cache, rewrite, responder, and content switching).
labeltype	<String>	Read-write	Type of label to be invoked.

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
			Possible values = reqvserver, resvserver, policylabel
__count	<Double>	Read-write	count parameter

Operations

(click to see Properties)

ADD:| DELETE:| GET| GET (ALL)| COUNT

Some options that you can use for each operations:

- **Getting warnings in response:**NITRO allows you to get warnings in an operation by specifying the "warning" query parameter as "yes". For example, to get warnings while connecting to the Citrix ADC appliance, the URL is as follows:

  http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/login?warning=yes

  If any, the warnings are displayed in the response payload with the HTTP code "209 X-NITRO-WARNING".

- **Authenticated access for individual NITRO operations:**NITRO allows you to logon to the Citrix ADC appliance to perform individual operations. You can use this option instead of creating a NITRO session (using the login object) and then using that session to perform all operations,

  To do this, you must specify the username and password in the request header of the NITRO request as follows:

  X-NITRO-USER:<username>

  X-NITRO-PASS:<password>

  Note: In such cases, make sure that the request header DOES not include the following:

  Cookie:NITRO_AUTH_TOKEN=<tokenvalue>

`*Note:`*

Mandatory parameters are marked in redand placeholder content is marked in <green>.

add

URL:http://<Citrix-ADC-IP-address(NSIP)/nitro/v1/config/crvserver_spilloverpolicy_binding

HTTP Method:PUT

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Content-Type:application/json

Request Payload:

```
{

"crvserver_spilloverpolicy_binding":{

<b>"name":<String_value>,

</b>"policyname":<String_value>,

"targetvserver":<String_value>,

"priority":<Double_value>,

"gotopriorityexpression":<String_value>,

"bindpoint":<String_value>,

"invoke":<Boolean_value>,

"labeltype":<String_value>,

"labelname":<String_value>

}}
```

Response:

HTTP Status Code on Success: 201 Created HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error

delete

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/crvserver_spilloverpolicy_binding/name_value<String>

Query-parameters:

args

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/crvserver_spilloverpolicy_binding/name_value<String>?**args=policyname:<String_value>,bindpoint:<String_value>,priority:<Double_value>**

HTTP Method:DELETE

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue>

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error

get

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/crvserver_spilloverpolicy_binding/name_value<String>

Query-parameters:

filter

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/crvserver_spilloverpolicy_binding/name_value<String>?**filter=property-name1:property-value1,property-name2:property-value2**

Use this query-parameter to get the filtered set of crvserver_spilloverpolicy_binding resources configured on Citrix ADC. Filtering can be done on any of the properties of the resource.

pagination

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/crvserver_spilloverpolicy_binding/name_value<String>?**pagesize=#no;pageno=#no**

Use this query-parameter to get the crvserver_spilloverpolicy_binding resources in chunks.

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "crvserver_spilloverpolicy_binding": [ {

"priority":<Double_value>,

"bindpoint":<String_value>,

"policyname":<String_value>,

"labelname":<String_value>,

"name":<String_value>,

"targetvserver":<String_value>,

"gotopriorityexpression":<String_value>,

"invoke":<Boolean_value>,

"labeltype":<String_value>

}]}
```

get (all)

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/crvserver_spilloverpolicy_binding

Query-parameters:

bulkbindings

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/crvserver_spilloverpolicy_binding?**bulkbindings=yes**

NITRO allows you to fetch bindings in bulk.

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "crvserver_spilloverpolicy_binding": [ {

"priority":<Double_value>,

"bindpoint":<String_value>,

"policyname":<String_value>,

"labelname":<String_value>,

"name":<String_value>,

"targetvserver":<String_value>,

"gotopriorityexpression":<String_value>,

"invoke":<Boolean_value>,
```

```
"labeltype":<String_value>

}]}
```

count

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/crvserver_spilloverpolicy_binding/name_value<String>?**count=yes**

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{"crvserver_spilloverpolicy_binding": [ { "__count": "#no"} ] }
```

CI

CI

Content Inspection

NAME	DESCRIPTION
contentinspectionaction	Configuration for Content Inspection action

NAME	DESCRIPTION
contentinspectioncallout	Configuration for Content Inspection callout
contentinspectionglobal_binding	Binding object showing the resources that can be bound to contentinspectionglobal
contentinspectionglobal_contentinspectionpolicy_binding	Binding object showing the contentinspectionpolicy that can be bound to contentinspectionglobal
contentinspectionparameter	Configuration for Contentinspection parameter
contentinspectionpolicy	Configuration for ContentInspection policy
contentinspectionpolicy_binding	Binding object showing the resources that can be bound to contentinspectionpolicy
contentinspectionpolicy_contentinspectionglobal_binding	Binding object showing the contentinspectionglobal that can be bound to contentinspectionpolicy
contentinspectionpolicy_contentinspectionpolicylabel_binding	Binding object showing the contentinspectionpolicylabel that can be bound to contentinspectionpolicy
contentinspectionpolicy_csvserver_binding	Binding object showing the csvserver that can be bound to contentinspectionpolicy
contentinspectionpolicy_lbvserver_binding	Binding object showing the lbvserver that can be bound to contentinspectionpolicy
contentinspectionpolicylabel	Configuration for ContentInspection policy label
contentinspectionpolicylabel_binding	Binding object showing the resources that can be bound to contentinspectionpolicylabel
contentinspectionpolicylabel_contentinspectionpolicy_binding	Binding object showing the contentinspectionpolicy that can be bound to contentinspectionpolicylabel
contentinspectionpolicylabel_policybinding_binding	Binding object showing the policybinding that can be bound to contentinspectionpolicylabel

NAME	DESCRIPTION
contentinspectionprofile	Configuration for ContentInspection profile

contentinspectionaction

Configuration for Content Inspection action resource.

Properties

(click to see Operations)

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
name	<String>	Read-write	Name of the remote service action. Must begin with an ASCII alphabetic or underscore (_) character, and must contain only ASCII alphanumeric, underscore, hash (#), period (.), space, colon (:), at (@), equals (=), and hyphen (-) characters.
type	<String>	Read-write	Type of operation this action is going to perform. following actions are available to configure: * ICAP - forward the incoming request or response to an ICAP server for modification.

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
			* INLINEINSPECTION - forward the incoming or outgoing packets to IPS server for Intrusion Prevention. * MIRROR - Forwards cloned packets for Intrusion Detection. * NOINSPECTION - This does not forward incoming and outgoing packets to the Inspection device. * NSTRACE - capture current and further incoming packets on this transaction. Possible values = ICAP, INLINEINSPECTION, MIRROR, NOINSPECTION
servername	<String>	Read-write	Name of the LB vserver or service. Minimum length = 1

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
			Maximum length = 127
serverip	<String>	Read-write	IP address of remoteService. Minimum length = 1
serverport	<Double>	Read-write	Port of remoteService. Default value: 1344 Minimum value = 1 Maximum value = 65535
icapprofilename	<String>	Read-write	Name of the ICAP profile to be attached to the contentInspection action. Minimum length = 1 Maximum length = 127
ifserverdown	<String>	Read-write	Name of the action to perform if the Vserver representing the remote service is not UP. This is not supported for NOINSPECTION Type. There are also some built-in actions which can be used. These are: * RESET - Reset the client connection by

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
			closing it. The client program, such as a browser, will handle this and may inform the user. The client may then resend the request if desired. * DROP - Drop the request without sending a response to the user. Default value: CONTINUE Possible values = CONTINUE, DROP, RESET
reqtimeout	<Double>	Read-write	Time, in seconds, within which the remote service request must complete. This is not supported for NOINSPECTION action type. If the request does not complete within this time, the specified request timeout action is executed. Zero disables the timeout. Default value: 0 Minimum value = 0

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
			Maximum value = 86400
reqtimeoutaction	<String>	Read-write	Name of the action to perform if the Vserver representing the remote service does not respond. This is not supported for NOINSPECTION action type. There are also some built-in actions which can be used. These are: * RESET - Reset the client connection by closing it. The client program, such as a browser, will handle this and may inform the user. The client may then resend the request if desired. * DROP - Drop the request without sending a response to the user. Default value: BYPASS Possible values = BYPASS, DROP, RESET

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
hits	<Double>	Read-only	The number of times the action has been taken.
referencecount	<Double>	Read-only	The number of references to the action.
undefhits	<Double>	Read-only	The number of times the action resulted in UNDEF.
builtin	<String[]>	Read-only	Flag to determine whether contentinspection action is built-in or not. Possible values = MODIFIABLE, DELETABLE, IMMUTABLE, PARTITION_ALL
__count	<Double>	Read-only	count parameter

Operations

(click to see Properties)

ADD| UPDATE| UNSET| DELETE| GET (ALL)| GET| COUNT

Some options that you can use for each operations:

- **Getting warnings in response:**NITRO allows you to get warnings in an operation by specifying the "warning" query parameter as "yes". For example, to get warnings while connecting to the Citrix ADC appliance, the URL is as follows:

  http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/login?warning=yes

  If any, the warnings are displayed in the response payload with the HTTP code "209 X-NITRO-WARNING".

- **Authenticated access for individual NITRO operations:**NITRO allows you to logon to the Citrix ADC appliance to perform individual operations. You can use this option instead of creating a NITRO session (using the login object) and then using that session to perform all operations,

  To do this, you must specify the username and password in the request header of the NITRO request as follows:

  X-NITRO-USER:<username>

  X-NITRO-PASS:<password>

  Note: In such cases, make sure that the request header DOES not include the following:

  Cookie:NITRO_AUTH_TOKEN=<tokenvalue>

`*Note:`*

Mandatory parameters are marked in redand placeholder content is marked in <green>.

add

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/contentinspectionaction

HTTP Method:POST

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Content-Type:application/json

Request Payload:

```
{"contentinspectionaction":{

<b>"name":<String_value>,

</b><b>"type":<String_value>,

</b>"servername":<String_value>,

"serverip":<String_value>,

"serverport":<Double_value>,
```

```
"icapprofilename":<String_value>,

"ifserverdown":<String_value>,

"reqtimeout":<Double_value>,

"reqtimeoutaction":<String_value>

}}
```

Response:

HTTP Status Code on Success: 201 Created HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error

update

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/contentinspectionaction

HTTP Method:PUT

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Content-Type:application/json

Request Payload:

```
{"contentinspectionaction":{

<b>"name":<String_value>,

</b>"servername":<String_value>,

"serverip":<String_value>,

"serverport":<Double_value>,

"icapprofilename":<String_value>,
```

```
"ifserverdown":<String_value>,

"reqtimeout":<Double_value>,

"reqtimeoutaction":<String_value>

}}
```

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error

unset

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/contentinspectionaction?**action=unset**

HTTP Method:POST

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Content-Type:application/json

Request Payload:

```
{"contentinspectionaction":{

<b>"name":<String_value>,

</b>"serverport":true,

"ifserverdown":true,

"reqtimeout":true

}}
```

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error

delete

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/contentinspectionaction/name_value<String>

HTTP Method:DELETE

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue>

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error

get (all)

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/contentinspectionaction

Query-parameters:

attrs

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/contentinspectionaction?**attrs=property-name1,property-name2**

Use this query parameter to specify the resource details that you want to retrieve.

filter

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/contentinspectionaction?**filter=property-name1:property-val1,property-name2:property-val2**

Use this query-parameter to get the filtered set of contentinspectionaction resources configured on Citrix ADC. Filtering can be done on any of the properties of the resource.

view

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/contentinspectionaction?**view=summary**

Note: By default, the retrieved results are displayed in detail view (?view=detail).

pagination

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/contentinspectionaction?**pagesize=#no;pageno=#no**

Use this query-parameter to get the contentinspectionaction resources in chunks.

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "contentinspectionaction": [ {

"name":<String_value>,

"type":<String_value>,

"servername":<String_value>,

"serverip":<String_value>,

"serverport":<Double_value>,

"icapprofilename":<String_value>,

"ifserverdown":<String_value>,

"reqtimeout":<Double_value>,
```

```
"reqtimeoutaction":<String_value>,

"hits":<Double_value>,

"referencecount":<Double_value>,

"undefhits":<Double_value>,

"builtin":<String[]_value>

}]}
```

get

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/contentinspectionaction/name_value<String>

Query-parameters:

attrs

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/contentinspectionaction/name_value<String>?**attrs=property-name1,property-name2**

Use this query parameter to specify the resource details that you want to retrieve.

view

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/contentinspectionaction/name_value<String>?**view=summary**

Note: By default, the retrieved results are displayed in detail view (?view=detail).

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "contentinspectionaction": [ {

"name":<String_value>,

"type":<String_value>,

"servername":<String_value>,

"serverip":<String_value>,

"serverport":<Double_value>,

"icapprofilename":<String_value>,

"ifserverdown":<String_value>,

"reqtimeout":<Double_value>,

"reqtimeoutaction":<String_value>,

"hits":<Double_value>,

"referencecount":<Double_value>,

"undefhits":<Double_value>,

"builtin":<String[]_value>

}]}
```

count

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/contentinspectionaction?**count=yes**

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "contentinspectionaction": [ { "__count": "#no"} ] }
```

contentinspectionglobal_binding

Binding object which returns the resources bound to contentinspectionglobal.

Properties

(click to see Operations)

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
contentinspectionglobal_contentinspectionpolicy_binding	<contentinspectionglobal_contentinspectionpolicy_binding[]>	Read-only	contentinspectionpolicy that can be bound to contentinspectionglobal.

Operations

(click to see Properties)

GET| GET (ALL)

Some options that you can use for each operations:

- **Getting warnings in response:**NITRO allows you to get warnings in an operation by specifying the "warning" query parameter as "yes". For example, to get warnings while connecting to the Citrix ADC appliance, the URL is as follows:

  http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/login?warning=yes

  If any, the warnings are displayed in the response payload with the HTTP code "209 X-NITRO-WARNING".

- **Authenticated access for individual NITRO operations:**NITRO allows you to logon to the Citrix ADC appliance to perform individual operations. You can use this option instead of creating a NITRO session (using the login object) and then using that session to perform all operations,

  To do this, you must specify the username and password in the request header of the NITRO request as follows:

  X-NITRO-USER:<username>

  X-NITRO-PASS:<password>

  Note: In such cases, make sure that the request header DOES not include the following:

  Cookie:NITRO_AUTH_TOKEN=<tokenvalue>

`*Note:`*

Mandatory parameters are marked in redand placeholder content is marked in <green>.

get

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/contentinspectionglobal_binding

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "contentinspectionglobal_binding": [ {

"contentinspectionglobal_contentinspectionpolicy_binding":<contentinspectionglobal_contentinspectionpolicy_binding[]_value>

}]}
```

get (all)

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/contentinspectionglobal_binding

Query-parameters:

bulkbindings

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/contentinspectionglobal_binding?**bulkbindings=yes**

NITRO allows you to fetch bindings in bulk.

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "contentinspectionglobal_binding": [ {

"contentinspectionglobal_contentinspectionpolicy_binding":<contentinspectionglobal_contentinspectionpolicy_binding[]_value>

}]}
```

contentinspectionglobal_contentinspectionpolicy_binding

Binding object showing the contentinspectionpolicy that can be bound to contentinspectionglobal.

Properties

(click to see Operations)

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
priority	<Double>	Read-write	Specifies the priority of the policy.
globalbindtype	<String>	Read-write	. Default value: SYSTEM_GLOBAL Possible values = SYSTEM_GLOBAL, VPN_GLOBAL, RNAT_GLOBAL
gotopriorityexpression	<String>	Read-write	Expression specifying the priority of the next policy which will get evaluated if the current policy rule evaluates to TRUE.

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
policyname	<String>	Read-write	Name of the contentInspection policy.
type	<String>	Read-write	The bindpoint to which to policy is bound. Possible values = REQ_OVERRIDE, REQ_DEFAULT, RES_OVERRIDE, RES_DEFAULT
labeltype	<String>	Read-write	Type of invocation. Available settings function as follows: * reqvserver - Forward the request to the specified request virtual server. * resvserver - Forward the response to the specified response virtual server. * policylabel - Invoke the specified policy label. Possible values = reqvserver, resvserver, policylabel
labelname	<String>	Read-write	* If labelType is policylabel, name of the policy label to invoke. * If labelType is reqvserver or resvserver, name of the virtual server to

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
			which to forward the request of response.
invoke	<Boolean>	Read-write	Terminate evaluation of policies bound to the current policy label, and then forward the request to the specified virtual server or evaluate the specified policy label.
numpol	<Double>	Read-only	The number of policies bound to the bindpoint.
flowtype	<Double>	Read-only	flowtype of the bound contentInspection policy.
__count	<Double>	Read-only	count parameter

Operations

(click to see Properties)

ADD:| DELETE:| GET| GET (ALL)| COUNT

Some options that you can use for each operations:

- **Getting warnings in response:**NITRO allows you to get warnings in an operation by specifying the "warning" query parameter as "yes". For example, to get warnings while connecting to the Citrix ADC appliance, the URL is as follows:

  http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/login?warning=yes

  If any, the warnings are displayed in the response payload with the HTTP code "209 X-NITRO-WARNING".

- **Authenticated access for individual NITRO operations:**NITRO allows you to logon to the Citrix ADC appliance to perform individual operations. You can use this option instead of creating a NITRO session (using the login object) and then using that session to perform all operations,

  To do this, you must specify the username and password in the request header of the NITRO request as follows:

  X-NITRO-USER:<username>

  X-NITRO-PASS:<password>

  Note: In such cases, make sure that the request header DOES not include the following:

  Cookie:NITRO_AUTH_TOKEN=<tokenvalue>

`*Note:`*

Mandatory parameters are marked in redand placeholder content is marked in <green>.

add

URL:http://<Citrix-ADC-IP-address(NSIP)/nitro/v1/config/contentinspectionglobal_contentinspectionpolicy_binding

HTTP Method:PUT

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Content-Type:application/json

Request Payload:

```
{

"contentinspectionglobal_contentinspectionpolicy_binding":{

<b>"policyname":<String_value>,

</b><b>"priority":<Double_value>,

</b>"gotopriorityexpression":<String_value>,
```

```
"type":<String_value>,

"invoke":<Boolean_value>,

"labeltype":<String_value>,

"labelname":<String_value>

}}
```

Response:

HTTP Status Code on Success: 201 Created HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error

delete

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/contentinspectionglobal_contentinspectionpolicy_binding

Query-parameters:

args

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/contentinspectionglobal_contentinspectionpolicy_binding?**args=policyname:<String_value>,type:<String_value>,priority:<Double_value>**

HTTP Method:DELETE

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue>

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error

get

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/contentinspectionglobal_contentinspectionpolicy_binding

Query-parameters:

filter

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/contentinspectionglobal_contentinspectionpolicy_binding?**filter=property-name1:property-value1,property-name2:property-value2**

Use this query-parameter to get the filtered set of contentinspectionglobal_contentinspectionpolicy_binding resources configured on Citrix ADC. Filtering can be done on any of the properties of the resource.

pagination

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/contentinspectionglobal_contentinspectionpolicy_binding?**pagesize=#no;pageno=#no**

Use this query-parameter to get the contentinspectionglobal_contentinspectionpolicy_binding resources in chunks.

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "contentinspectionglobal_contentinspectionpolicy_binding": [ {
```

```
"priority":<Double_value>,

"globalbindtype":<String_value>,

"gotopriorityexpression":<String_value>,

"policyname":<String_value>,

"type":<String_value>,

"labeltype":<String_value>,

"labelname":<String_value>,

"invoke":<Boolean_value>,

"numpol":<Double_value>,

"flowtype":<Double_value>

}]}
```

get (all)

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/contentinspectionglobal_contentinspectionpolicy_binding

Query-parameters:

bulkbindings

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/contentinspectionglobal_contentinspectionpolicy_binding?**bulkbindings=yes**

NITRO allows you to fetch bindings in bulk.

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "contentinspectionglobal_contentinspectionpolicy_binding": [ {

"priority":<Double_value>,

"globalbindtype":<String_value>,

"gotopriorityexpression":<String_value>,

"policyname":<String_value>,

"type":<String_value>,

"labeltype":<String_value>,

"labelname":<String_value>,

"invoke":<Boolean_value>,

"numpol":<Double_value>,

"flowtype":<Double_value>

}]}
```

count

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/contentinspectionglobal_contentinspectionpolicy_binding?**count=yes**

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "contentinspectionglobal_contentinspectionpolicy_binding":
[ { "__count": "#no"} ] }
```

contentinspectionpolicy

Configuration for ContentInspection policy resource.

Properties

(click to see Operations)

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
name	<String>	Read-write	Name for the contentInspection policy. Must begin with a letter, number, or the underscore character (_), and must contain only letters, numbers, and the hyphen (-), period (.) pound (#), space ( ), at (@), equals (=), colon (:), and underscore

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
			characters. Can be changed after the contentInspection policy is added. The following requirement applies only to the Citrix ADC CLI: If the name includes one or more spaces, enclose the name in double or single quotation marks (for example, "my contentInspection policy" or 'my contentInspection policy').
rule	<String>	Read-write	Expression that the policy uses to determine whether to execute the specified action.
action	<String>	Read-write	Name of the contentInspection action to perform if the request matches this contentInspection policy. There are also some built-in

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
			actions which can be used. These are: * NOINSPECTION - Send the request from the client to the server or response from the server to the client without sending it to Inspection device for Content Inspection. * RESET - Resets the client connection by closing it. The client program, such as a browser, will handle this and may inform the user. The client may then resend the request if desired. * DROP - Drop the request without sending a response to the user.
undefaction	<String>	Read-write	Action to perform if the result of policy evaluation is undefined (UNDEF). An UNDEF event indicates an internal error condition. Only

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
			the above built-in actions can be used.
comment	<String>	Read-write	Any type of information about this contentInspection policy.
logaction	<String>	Read-write	Name of the messagelog action to use for requests that match this policy.
newname	<String>	Read-write	New name for the contentInspection policy. Must begin with a letter, number, or the underscore character (_), and must contain only letters, numbers, and the hyphen (-), period (.) hash (#), space ( ), at (@), equals (=), colon (:), and underscore characters. The following requirement applies only to the Citrix ADC CLI: If the name includes one or more spaces, enclose the name in double or single quotation marks (for example, "my

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
			contentInspection policy" or 'my contentInspection policy'). Minimum length = 1
hits	<Double>	Read-only	Number of hits.
undefhits	<Double>	Read-only	Number of policy UNDEF hits.
__count	<Double>	Read-only	count parameter

Operations

(click to see Properties)

ADD| DELETE| UPDATE| UNSET| RENAME| GET (ALL)| GET| COUNT

Some options that you can use for each operations:

- **Getting warnings in response:**NITRO allows you to get warnings in an operation by specifying the "warning" query parameter as "yes". For example, to get warnings while connecting to the Citrix ADC appliance, the URL is as follows:

  http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/login?warning=yes

  If any, the warnings are displayed in the response payload with the HTTP code "209 X-NITRO-WARNING".

- **Authenticated access for individual NITRO operations:**NITRO allows you to logon to the Citrix ADC appliance to perform individual operations. You can use this option instead of creating a NITRO session (using the login object) and then using that session to perform all operations,

  To do this, you must specify the username and password in the request header of the NITRO request as follows:

  X-NITRO-USER:<username>

  X-NITRO-PASS:<password>

Note: In such cases, make sure that the request header DOES not include the following:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue>

`*Note:` *

Mandatory parameters are marked in redand placeholder content is marked in <green>.

add

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/contentinspectionpolicy

HTTP Method:POST

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Content-Type:application/json

Request Payload:

```
{"contentinspectionpolicy":{

<b>"name":<String_value>,

</b><b>"rule":<String_value>,

</b><b>"action":<String_value>,

</b>"undefaction":<String_value>,

"comment":<String_value>,

"logaction":<String_value>

}}
```

Response:

HTTP Status Code on Success: 201 Created HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error

delete

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/contentinspectionpolicy/name_value<String>

HTTP Method:DELETE

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue>

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error

update

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/contentinspectionpolicy

HTTP Method:PUT

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Content-Type:application/json

Request Payload:

```
{"contentinspectionpolicy":{

<b>"name":<String_value>,

</b>"rule":<String_value>,

"action":<String_value>,

"undefaction":<String_value>,

"comment":<String_value>,

"logaction":<String_value>

}}
```

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error

unset

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/contentinspectionpolicy?**action=unset**

HTTP Method:POST

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Content-Type:application/json

Request Payload:

```
{"contentinspectionpolicy":{

<b>"name":<String_value>,

</b>"undefaction":true,

"comment":true,

"logaction":true

}}
```

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error

rename

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/contentinspectionpolicy?**action=rename**

HTTP Method:POST

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Content-Type:application/json

Request Payload:

```
{"contentinspectionpolicy":{

<b>"name":<String_value>,

</b><b>"newname":<String_value>

</b>}}
```

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error

get (all)

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/contentinspectionpolicy

Query-parameters:

attrs

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/contentinspectionpolicy?**attrs=property-name1,property-name2**

Use this query parameter to specify the resource details that you want to retrieve.

filter

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/contentinspectionpolicy?**filter=property-name1:property-val1,property-name2:property-val2**

Use this query-parameter to get the filtered set of contentinspectionpolicy resources configured on Citrix ADC. Filtering can be done on any of the properties of the resource.

view

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/contentinspectionpolicy?**view=summary**

Note: By default, the retrieved results are displayed in detail view (?view=detail).

pagination

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/contentinspectionpolicy?**pagesize=#no;pageno=#no**

Use this query-parameter to get the contentinspectionpolicy resources in chunks.

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "contentinspectionpolicy": [ {

"name":<String_value>,

"rule":<String_value>,

"action":<String_value>,

"undefaction":<String_value>,

"hits":<Double_value>,
```

```
"undefhits":<Double_value>,

"comment":<String_value>,

"logaction":<String_value>

}]}
```

get

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/contentinspectionpolicy/name_value<String>

Query-parameters:

attrs

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/contentinspectionpolicy/name_value<String>?**attrs=property-name1,property-name2**

Use this query parameter to specify the resource details that you want to retrieve.

view

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/contentinspectionpolicy/name_value<String>?**view=summary**

Note: By default, the retrieved results are displayed in detail view (?view=detail).

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "contentinspectionpolicy": [ {

"name":<String_value>,

"rule":<String_value>,

"action":<String_value>,

"undefaction":<String_value>,

"hits":<Double_value>,

"undefhits":<Double_value>,

"comment":<String_value>,

"logaction":<String_value>

}]}
```

count

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/contentinspectionpolicy?**count=yes**

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "contentinspectionpolicy": [ { "__count": "#no"} ] }
```

contentinspectionpolicy_binding

Binding object which returns the resources bound to contentinspectionpolicy.

Properties

(click to see Operations)

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
name	<String>	Read-write	Name of the contentInspection policy for which to display settings.
contentinspectionpolicy_csvserver_binding	<contentinspectionpolicy_csvserver_binding[]>	Read-only	csvserver that can be bound to contentinspectionpolicy.
contentinspectionpolicy_lbvserver_binding	<contentinspectionpolicy_lbvserver_binding[]>	Read-only	lbvserver that can be bound to contentinspectionpolicy.
contentinspectionpolicy_contentinspectionglobal_binding	<contentinspectionpolicy_contentinspectionglobal_binding[]>	Read-only	contentinspectionglobal that can be bound to contentinspectionpolicy.
contentinspectionpolicy_contentinspectionpolicylabel_binding	<contentinspectionpolicy_contentinspectionpolicylabel_binding[]>	Read-only	contentinspectionpolicylabel that can be bound to contentinspectionpolicy.

Operations

(click to see Properties)

GET| GET (ALL)

Some options that you can use for each operations:

- **Getting warnings in response:**NITRO allows you to get warnings in an operation by specifying the "warning" query parameter as "yes". For example, to get warnings while connecting to the Citrix ADC appliance, the URL is as follows:

  http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/login?warning=yes

  If any, the warnings are displayed in the response payload with the HTTP code "209 X-NITRO-WARNING".

- **Authenticated access for individual NITRO operations:**NITRO allows you to logon to the Citrix ADC appliance to perform individual operations. You can use this option instead of creating a NITRO session (using the login object) and then using that session to perform all operations,

  To do this, you must specify the username and password in the request header of the NITRO request as follows:

  X-NITRO-USER:<username>

  X-NITRO-PASS:<password>

  Note: In such cases, make sure that the request header DOES not include the following:

  Cookie:NITRO_AUTH_TOKEN=<tokenvalue>

`*Note:`*

Mandatory parameters are marked in redand placeholder content is marked in <green>.

get

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/contentinspectionpolicy_binding/name_value<String>

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "contentinspectionpolicy_binding": [ {

"name":<String_value>,

"contentinspectionpolicy_lbvserver_binding":<contentinspectionpolicy_lbvse
rver_binding[]_value>,

"contentinspectionpolicy_csvserver_binding":<contentinspectionpolicy_csvse
rver_binding[]_value>,

"contentinspectionpolicy_contentinspectionpolicylabel_binding":<contentins
pectionpolicy_contentinspectionpolicylabel_binding[]_value>,

"contentinspectionpolicy_contentinspectionglobal_binding":<contentinspecti
onpolicy_contentinspectionglobal_binding[]_value>

}]}
```

get (all)

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/contentinspectionpolicy_binding

Query-parameters:

bulkbindings

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/contentinspectionpolicy_binding?**bulkbindings=yes**

NITRO allows you to fetch bindings in bulk.

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "contentinspectionpolicy_binding": [ {

"name":<String_value>,

"contentinspectionpolicy_lbvserver_binding":<contentinspectionpolicy_lbvse
rver_binding[]_value>,

"contentinspectionpolicy_csvserver_binding":<contentinspectionpolicy_csvse
rver_binding[]_value>,

"contentinspectionpolicy_contentinspectionpolicylabel_binding":<contentins
pectionpolicy_contentinspectionpolicylabel_binding[]_value>,

"contentinspectionpolicy_contentinspectionglobal_binding":<contentinspecti
onpolicy_contentinspectionglobal_binding[]_value>

}]}
```

contentinspectionpolicy_contentinspectionglobal_binding

Binding object showing the contentinspectionglobal that can be bound to contentinspectionpolicy.

Properties

(click to see Operations)

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
boundto	<String>	Read-write	Location where policy is bound.
name	<String>	Read-write	Name of the contentInspection policy for which to display settings.
priority	<Double>	Read-only	Specifies the priority of the policy.
gotopriorityexpression	<String>	Read-only	Expression specifying the priority of the next policy which will get evaluated if the current policy rule evaluates to TRUE.
labeltype	<String>	Read-only	Type of policy label invocation. Possible values = reqvserver, resvserver, policylabel
activepolicy	<Integer>	Read-only	Indicates whether policy is bound or not.
labelname	<String>	Read-only	Name of the label to invoke if the current policy rule evaluates to TRUE.
__count	<Double>	Read-only	count parameter

Operations

(click to see Properties)

GET| GET (ALL)| COUNT

Some options that you can use for each operations:

- **Getting warnings in response:**NITRO allows you to get warnings in an operation by specifying the "warning" query parameter as "yes". For example, to get warnings while connecting to the Citrix ADC appliance, the URL is as follows:

  http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/login?warning=yes

  If any, the warnings are displayed in the response payload with the HTTP code "209 X-NITRO-WARNING".

- **Authenticated access for individual NITRO operations:**NITRO allows you to logon to the Citrix ADC appliance to perform individual operations. You can use this option instead of creating a NITRO session (using the login object) and then using that session to perform all operations,

  To do this, you must specify the username and password in the request header of the NITRO request as follows:

  X-NITRO-USER:<username>

  X-NITRO-PASS:<password>

  Note: In such cases, make sure that the request header DOES not include the following:

  Cookie:NITRO_AUTH_TOKEN=<tokenvalue>

`*Note:`*

Mandatory parameters are marked in redand placeholder content is marked in <green>.

get

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/contentinspectionpolicy_contentinspectionglobal_binding/name_value<String>

Query-parameters:

filter

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/contentinspectionpolicy_contentinspectionglobal_binding/name_value<String>?**filter=property-name1:property-value1,property-name2:property-value2**

Use this query-parameter to get the filtered set of contentinspectionpolicy_contentinspectionglobal_binding resources configured on Citrix ADC. Filtering can be done on any of the properties of the resource.

pagination

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/contentinspectionpolicy_contentinspectionglobal_binding/name_value<String>?**pagesize=#no;pageno=#no**

Use this query-parameter to get the contentinspectionpolicy_contentinspectionglobal_binding resources in chunks.

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "contentinspectionpolicy_contentinspectionglobal_binding": [ {

"boundto":<String_value>,

"name":<String_value>,

"priority":<Double_value>,
```

```
"gotopriorityexpression":<String_value>,

"labeltype":<String_value>,

"activepolicy":<Integer_value>,

"labelname":<String_value>

}]}
```

get (all)

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/contentinspectionpolicy_contentinspectionglobal_binding

Query-parameters:

bulkbindings

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/contentinspectionpolicy_contentinspectionglobal_binding?**bulkbindings=yes**

NITRO allows you to fetch bindings in bulk.

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "contentinspectionpolicy_contentinspectionglobal_binding": [ {

"boundto":<String_value>,

"name":<String_value>,

"priority":<Double_value>,

"gotopriorityexpression":<String_value>,

"labeltype":<String_value>,

"activepolicy":<Integer_value>,

"labelname":<String_value>

}]}
```

count

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/contentinspectionpolicy_contentinspectionglobal_binding/name_value<String>?**count=yes**

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{"contentinspectionpolicy_contentinspectionglobal_binding": [ { "__count":
"#no"} ] }
```


contentinspectionpolicy_contentinspectionpolicylabel_binding

Binding object showing the contentinspectionpolicylabel that can be bound to contentinspectionpolicy.

Properties

(click to see Operations)

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
boundto	<String>	Read-write	Location where policy is bound.
name	<String>	Read-write	Name of the contentInspection policy for which to display settings.
priority	<Double>	Read-only	Specifies the priority of the policy.
gotopriorityexpression	<String>	Read-only	Expression specifying the priority of the next policy which will get evaluated if the current policy rule evaluates to TRUE.
labeltype	<String>	Read-only	Type of policy label invocation. Possible values = reqvserver, resvserver, policylabel

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
activepolicy	<Integer>	Read-only	Indicates whether policy is bound or not.
labelname	<String>	Read-only	Name of the label to invoke if the current policy rule evaluates to TRUE.
__count	<Double>	Read-only	count parameter

Operations

(click to see Properties)

GET| GET (ALL)| COUNT

Some options that you can use for each operations:

- **Getting warnings in response:**NITRO allows you to get warnings in an operation by specifying the "warning" query parameter as "yes". For example, to get warnings while connecting to the Citrix ADC appliance, the URL is as follows:

  http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/login?warning=yes

  If any, the warnings are displayed in the response payload with the HTTP code "209 X-NITRO-WARNING".

- **Authenticated access for individual NITRO operations:**NITRO allows you to logon to the Citrix ADC appliance to perform individual operations. You can use this option instead of creating a NITRO session (using the login object) and then using that session to perform all operations,

  To do this, you must specify the username and password in the request header of the NITRO request as follows:

  X-NITRO-USER:<username>

  X-NITRO-PASS:<password>

  Note: In such cases, make sure that the request header DOES not include the following:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue>

`*Note:`*

Mandatory parameters are marked in redand placeholder content is marked in <green>.

get

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/contentinspectionpolicy_contentinspectionpolicylabel_binding/name_value<String>

Query-parameters:

filter

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/contentinspectionpolicy_contentinspectionpolicylabel_binding/name_value<String>?**filter=property-name1:property-value1,property-name2:property-value2**

Use this query-parameter to get the filtered set of contentinspectionpolicy_contentinspectionpolicylabel_binding resources configured on Citrix ADC. Filtering can be done on any of the properties of the resource.

pagination

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/contentinspectionpolicy_contentinspectionpolicylabel_binding/name_value<String>?**pagesize=#no;pageno=#no**

Use this query-parameter to get the contentinspectionpolicy_contentinspectionpolicylabel_binding resources in chunks.

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "contentinspectionpolicy_contentinspectionpolicylabel_binding": [ {

"boundto":<String_value>,

"name":<String_value>,

"priority":<Double_value>,

"gotopriorityexpression":<String_value>,

"labeltype":<String_value>,

"activepolicy":<Integer_value>,

"labelname":<String_value>

}]}
```

get (all)

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/contentinspectionpolicy_contentinspectionpolicylabel_binding

Query-parameters:

bulkbindings

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/contentinspectionpolicy_contentinspectionpolicylabel_binding?**bulkbindings=yes**

NITRO allows you to fetch bindings in bulk.

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "contentinspectionpolicy_contentinspectionpolicylabel_binding": [ {

"boundto":<String_value>,

"name":<String_value>,

"priority":<Double_value>,

"gotopriorityexpression":<String_value>,

"labeltype":<String_value>,

"activepolicy":<Integer_value>,

"labelname":<String_value>

}]}
```

count

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/contentinspectionpolicy_contentinspectionpolicylabel_binding/name_value<String>?**count=yes**

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{"contentinspectionpolicy_contentinspectionpolicylabel_binding":
[ { "__count": "#no"} ] }
```

contentinspectionpolicy_csvserver_binding

Binding object showing the csvserver that can be bound to contentinspectionpolicy.

Properties

(click to see Operations)

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
boundto	<String>	Read-write	Location where policy is bound.
name	<String>	Read-write	Name of the contentInspection policy for which to display settings.
priority	<Double>	Read-only	Specifies the priority of the policy.
gotopriorityexpression	<String>	Read-only	Expression specifying the priority of the next policy which will get evaluated if the current policy rule evaluates to TRUE.

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
labeltype	<String>	Read-only	Type of policy label invocation. Possible values = reqvserver, resvserver, policylabel
activepolicy	<Integer>	Read-only	Indicates whether policy is bound or not.
labelname	<String>	Read-only	Name of the label to invoke if the current policy rule evaluates to TRUE.
__count	<Double>	Read-only	count parameter

Operations

(click to see Properties)

GET| GET (ALL)| COUNT

Some options that you can use for each operations:

- **Getting warnings in response:**NITRO allows you to get warnings in an operation by specifying the "warning" query parameter as "yes". For example, to get warnings while connecting to the Citrix ADC appliance, the URL is as follows:

  http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/login?warning=yes

  If any, the warnings are displayed in the response payload with the HTTP code "209 X-NITRO-WARNING".
- **Authenticated access for individual NITRO operations:**NITRO allows you to logon to the Citrix ADC appliance to perform individual operations. You can use this option instead of creating a NITRO session (using the login object) and then using that session to perform all operations,

To do this, you must specify the username and password in the request header of the NITRO request as follows:

X-NITRO-USER:<username>

X-NITRO-PASS:<password>

Note: In such cases, make sure that the request header DOES not include the following:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue>

`*Note:`*

Mandatory parameters are marked in redand placeholder content is marked in <green>.

get

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/contentinspectionpolicy_csvserver_binding/name_value<String>

Query-parameters:

filter

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/contentinspectionpolicy_csvserver_binding/name_value<String>?**filter=property-name1:property-value1,property-name2:property-value2**

Use this query-parameter to get the filtered set of contentinspectionpolicy_csvserver_binding resources configured on Citrix ADC. Filtering can be done on any of the properties of the resource.

pagination

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/contentinspectionpolicy_csvserver_binding/name_value<String>?**pagesize=#no;pageno=#no**

Use this query-parameter to get the contentinspectionpolicy_csvserver_binding resources in chunks.

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "contentinspectionpolicy_csvserver_binding": [ {

"boundto":<String_value>,

"name":<String_value>,

"priority":<Double_value>,

"gotopriorityexpression":<String_value>,

"labeltype":<String_value>,

"activepolicy":<Integer_value>,

"labelname":<String_value>

}]}
```

get (all)

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/contentinspectionpolicy_csvserver_binding

Query-parameters:

bulkbindings

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/contentinspectionpolicy_csvserver_binding?**bulkbindings=yes**

NITRO allows you to fetch bindings in bulk.

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "contentinspectionpolicy_csvserver_binding": [ {

"boundto":<String_value>,

"name":<String_value>,

"priority":<Double_value>,

"gotopriorityexpression":<String_value>,

"labeltype":<String_value>,

"activepolicy":<Integer_value>,

"labelname":<String_value>

}]}
```

count

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/contentinspectionpolicy_csvserver_binding/name_value<String>?**count=yes**

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{"contentinspectionpolicy_csvserver_binding": [ { "__count": "#no"} ] }
```

contentinspectionpolicy_lbvserver_binding

Binding object showing the lbvserver that can be bound to contentinspectionpolicy.

Properties

(click to see Operations)

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
boundto	<String>	Read-write	Location where policy is bound.
name	<String>	Read-write	Name of the contentInspection policy for which to display settings.
priority	<Double>	Read-only	Specifies the priority of the policy.
gotopriorityexpression	<String>	Read-only	Expression specifying the priority of the next policy which will get evaluated if the

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
			current policy rule evaluates to TRUE.
labeltype	<String>	Read-only	Type of policy label invocation. Possible values = reqvserver, resvserver, policylabel
activepolicy	<Integer>	Read-only	Indicates whether policy is bound or not.
labelname	<String>	Read-only	Name of the label to invoke if the current policy rule evaluates to TRUE.
__count	<Double>	Read-only	count parameter

Operations

(click to see Properties)

GET| GET (ALL)| COUNT

Some options that you can use for each operations:

- **Getting warnings in response:**NITRO allows you to get warnings in an operation by specifying the "warning" query parameter as "yes". For example, to get warnings while connecting to the Citrix ADC appliance, the URL is as follows:

  http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/login?warning=yes

  If any, the warnings are displayed in the response payload with the HTTP code "209 X-NITRO-WARNING".
- **Authenticated access for individual NITRO operations:**NITRO allows you to logon to the Citrix ADC appliance to perform individual operations. You can use this option instead of

creating a NITRO session (using the login object) and then using that session to perform all operations,

To do this, you must specify the username and password in the request header of the NITRO request as follows:

X-NITRO-USER:<username>

X-NITRO-PASS:<password>

Note: In such cases, make sure that the request header DOES not include the following:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue>

`*Note:`*

Mandatory parameters are marked in redand placeholder content is marked in <green>.

get

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/contentinspectionpolicy_lbvserver_binding/name_value<String>

Query-parameters:

filter

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/contentinspectionpolicy_lbvserver_binding/name_value<String>?**filter=property-name1:property-value1,property-name2:property-value2**

Use this query-parameter to get the filtered set of contentinspectionpolicy_lbvserver_binding resources configured on Citrix ADC. Filtering can be done on any of the properties of the resource.

pagination

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/contentinspectionpolicy_lbvserver_binding/name_value<String>?**pagesize=#no;pageno=#no**

Use this query-parameter to get the contentinspectionpolicy_lbvserver_binding resources in chunks.

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "contentinspectionpolicy_lbvserver_binding": [ {

"boundto":<String_value>,

"name":<String_value>,

"priority":<Double_value>,

"gotopriorityexpression":<String_value>,

"labeltype":<String_value>,

"activepolicy":<Integer_value>,

"labelname":<String_value>

}]}
```

get (all)

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/contentinspectionpolicy_lbvserver_binding

Query-parameters:

bulkbindings

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/contentinspectionpolicy_lbvserver_binding?**bulkbindings=yes**

NITRO allows you to fetch bindings in bulk.

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "contentinspectionpolicy_lbvserver_binding": [ {

"boundto":<String_value>,

"name":<String_value>,

"priority":<Double_value>,

"gotopriorityexpression":<String_value>,

"labeltype":<String_value>,

"activepolicy":<Integer_value>,

"labelname":<String_value>

}]}
```

count

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/contentinspectionpolicy_lbvserver_binding/name_value<String>?**count=yes**

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{"contentinspectionpolicy_lbvserver_binding": [ { "__count": "#no"} ] }
```

contentinspectionpolicylabel

Configuration for ContentInspection policy label resource.

Properties

(click to see Operations)

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
labelname	<String>	Read-write	Name for the contentInspection policy label. Must begin with a letter, number, or the underscore character (_), and must contain only letters, numbers, and the hyphen (-), period (.) hash (#), space ( ), at (@), equals (=), colon (:), and underscore

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
			characters. Cannot be changed after the contentInspection policy label is added. The following requirement applies only to the Citrix ADC CLI: If the name includes one or more spaces, enclose the name in double or single quotation marks (for example, "my contentInspection policy label" or 'my contentInspection policy label').
type	<String>	Read-write	Type of packets (request or response packets) against which to match the policies bound to this policy label. Possible values = REQ, RES
comment	<String>	Read-write	Any comments to preserve information about this contentInspection policy label.
newname	<String>	Read-write	New name for the contentInspection policy label.

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
			Must begin with a letter, number, or the underscore character (_), and must contain only letters, numbers, and the hyphen (-), period (.) hash (#), space ( ), at (@), equals (=), colon (:), and underscore characters. The following requirement applies only to the Citrix ADC CLI: If the name includes one or more spaces, enclose the name in double or single quotation marks (for example, "my policy label" or 'my policy label'). Minimum length = 1
numpol	<Double>	Read-only	Number of polices bound to label.
hits	<Double>	Read-only	Number of times policy label was invoked.
priority	<Double>	Read-only	Specifies the priority of the policy.

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
gotopriorityexpression	<String>	Read-only	Expression specifying the priority of the next policy which will get evaluated if the current policy rule evaluates to TRUE.
labeltype	<String>	Read-only	Type of invocation. Available settings function as follows: * reqvserver - Forward the request to the specified request virtual server. * resvserver - Forward the response to the specified response virtual server. * policylabel - Invoke the specified policy label. Possible values = reqvserver, resvserver, policylabel
invoke_labelname	<String>	Read-only	* If labelType is policylabel, name of the policy label to invoke. * If labelType is reqvserver or resvserver, name of

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
			the virtual server to which to forward the request or response.
flowtype	<Double>	Read-only	Flowtype of the bound contentInspection policy.
isdefault	<Boolean>	Read-only	A value of true is returned if it is a default cipolicylabel.
__count	<Double>	Read-only	count parameter

Operations

(click to see Properties)

ADD| DELETE| RENAME| GET (ALL)| GET| COUNT

Some options that you can use for each operations:

- **Getting warnings in response:**NITRO allows you to get warnings in an operation by specifying the "warning" query parameter as "yes". For example, to get warnings while connecting to the Citrix ADC appliance, the URL is as follows:

  http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/login?warning=yes

  If any, the warnings are displayed in the response payload with the HTTP code "209 X-NITRO-WARNING".

- **Authenticated access for individual NITRO operations:**NITRO allows you to logon to the Citrix ADC appliance to perform individual operations. You can use this option instead of creating a NITRO session (using the login object) and then using that session to perform all operations,

  To do this, you must specify the username and password in the request header of the NITRO request as follows:

  X-NITRO-USER:<username>

X-NITRO-PASS:<password>

Note: In such cases, make sure that the request header DOES not include the following:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue>

`*Note:`*

Mandatory parameters are marked in redand placeholder content is marked in <green>.

add

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/contentinspectionpolicylabel

HTTP Method:POST

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Content-Type:application/json

Request Payload:

```
{"contentinspectionpolicylabel":{

<b>"labelname":<String_value>,

</b><b>"type":<String_value>,

</b>"comment":<String_value>

}}
```

Response:

HTTP Status Code on Success: 201 Created HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error

delete

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/contentinspectionpolicylabel/labelname_value<String>

HTTP Method:DELETE

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue>

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error

rename

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/contentinspectionpolicylabel?**action=rename**

HTTP Method:POST

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Content-Type:application/json

Request Payload:

```
{"contentinspectionpolicylabel":{

<b>"labelname":<String_value>,

</b><b>"newname":<String_value>

</b>}}
```

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error

get (all)

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/contentinspectionpolicylabel

Query-parameters:

attrs

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/contentinspectionpolicylabel?**attrs=property-name1,property-name2**

Use this query parameter to specify the resource details that you want to retrieve.

filter

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/contentinspectionpolicylabel?**filter=property-name1:property-val1,property-name2:property-val2**

Use this query-parameter to get the filtered set of contentinspectionpolicylabel resources configured on Citrix ADC. Filtering can be done on any of the properties of the resource.

view

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/contentinspectionpolicylabel?**view=summary**

Note: By default, the retrieved results are displayed in detail view (?view=detail).

pagination

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/contentinspectionpolicylabel?**pagesize=#no;pageno=#no**

Use this query-parameter to get the contentinspectionpolicylabel resources in chunks.

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "contentinspectionpolicylabel": [ {
"labelname":<String_value>,
"type":<String_value>,
"numpol":<Double_value>,
"hits":<Double_value>,
"priority":<Double_value>,
"gotopriorityexpression":<String_value>,
"labeltype":<String_value>,
"invoke_labelname":<String_value>,
"flowtype":<Double_value>,
"isdefault":<Boolean_value>,
"comment":<String_value>
}]}
```

get

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/contentinspectionpolicylabel/labelname_value<String>

Query-parameters:

attrs

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/contentinspectionpolicylabel/labelname_value<String>?**attrs=property-name1,property-name2**

Use this query parameter to specify the resource details that you want to retrieve.

view

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/contentinspectionpolicylabel/labelname_value<String>?**view=summary**

Note: By default, the retrieved results are displayed in detail view (?view=detail).

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "contentinspectionpolicylabel": [ {

"labelname":<String_value>,

"type":<String_value>,

"numpol":<Double_value>,

"hits":<Double_value>,

"priority":<Double_value>,

"gotopriorityexpression":<String_value>,

"labeltype":<String_value>,

"invoke_labelname":<String_value>,

"flowtype":<Double_value>,

"isdefault":<Boolean_value>,
```

```
"comment":<String_value>

}]}
```

count

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/contentinspectionpolicylabel?**count=yes**

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "contentinspectionpolicylabel": [ { "__count": "#no"} ] }
```

contentinspectionpolicylabel_binding

Binding object which returns the resources bound to contentinspectionpolicylabel.

Properties

(click to see Operations)

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
labelname	<String>	Read-write	Name of the contentInspection policy label.

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
contentinspectionpolicylabel_policybinding_binding	<contentinspectionpolicylabel_policybinding_binding[]>	Read-only	policybinding that can be bound to contentinspectionpolicylabel.
contentinspectionpolicylabel_contentinspectionpolicy_binding	<contentinspectionpolicylabel_contentinspectionpolicy_binding[]>	Read-only	contentinspectionpolicy that can be bound to contentinspectionpolicylabel.

Operations

(click to see Properties)

GET| GET (ALL)

Some options that you can use for each operations:

- **Getting warnings in response:**NITRO allows you to get warnings in an operation by specifying the "warning" query parameter as "yes". For example, to get warnings while connecting to the Citrix ADC appliance, the URL is as follows:

  http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/login?warning=yes

  If any, the warnings are displayed in the response payload with the HTTP code "209 X-NITRO-WARNING".

- **Authenticated access for individual NITRO operations:**NITRO allows you to logon to the Citrix ADC appliance to perform individual operations. You can use this option instead of creating a NITRO session (using the login object) and then using that session to perform all operations,

  To do this, you must specify the username and password in the request header of the NITRO request as follows:

  X-NITRO-USER:<username>

  X-NITRO-PASS:<password>

  Note: In such cases, make sure that the request header DOES not include the following:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue>

`*Note:` *

Mandatory parameters are marked in redand placeholder content is marked in <green>.

get

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/contentinspectionpolicylabel_binding/labelname_value<String>

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "contentinspectionpolicylabel_binding": [ {

"labelname":<String_value>,

"contentinspectionpolicylabel_policybinding_binding":<contentinspectionpolicylabel_policybinding_binding[]_value>,

"contentinspectionpolicylabel_contentinspectionpolicy_binding":<contentinspectionpolicylabel_contentinspectionpolicy_binding[]_value>

}]}
```

get (all)

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/contentinspectionpolicylabel_binding

Query-parameters:

bulkbindings

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/contentinspectionpolicylabel_binding?**bulkbindings=yes**

NITRO allows you to fetch bindings in bulk.

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "contentinspectionpolicylabel_binding": [ {

"labelname":<String_value>,

"contentinspectionpolicylabel_policybinding_binding":<contentinspectionpol
icylabel_policybinding_binding[]_value>,

"contentinspectionpolicylabel_contentinspectionpolicy_binding":<contentins
pectionpolicylabel_contentinspectionpolicy_binding[]_value>

}]}
```

contentinspectionpolicylabel_contentinspectionpolicy_binding

Binding object showing the contentinspectionpolicy that can be bound to contentinspectionpolicylabel.

Properties

(click to see Operations)

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
priority	<Double>	Read-write	Specifies the priority of the policy.
gotopriorityexpression	<String>	Read-write	Expression specifying the priority of the next policy which will get evaluated if the current policy rule evaluates to TRUE.
policyname	<String>	Read-write	Name of the contentInspection policy to bind to the policy label.
labeltype	<String>	Read-write	Type of invocation. Available settings function as follows: * reqvserver - Forward the request to the specified request virtual server. * resvserver - Forward the response to the specified response virtual server. * policylabel - Invoke the specified policy label. Possible values = reqvserver, resvserver, policylabel
labelname	<String>	Read-write	Name of the contentInspection

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
			policy label to which to bind the policy.
invoke_labelname	<String>	Read-write	* If labelType is policylabel, name of the policy label to invoke. * If labelType is reqvserver or resvserver, name of the virtual server to which to forward the request or response.
invoke	<Boolean>	Read-write	Suspend evaluation of policies bound to the current policy label, and then forward the request to the specified virtual server or evaluate the specified policy label.
__count	<Double>	Read-write	count parameter

Operations

(click to see Properties)

ADD:| DELETE:| GET| GET (ALL)| COUNT

Some options that you can use for each operations:

- **Getting warnings in response:**NITRO allows you to get warnings in an operation by specifying the "warning" query parameter as "yes". For example, to get warnings while connecting to the Citrix ADC appliance, the URL is as follows:

  http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/login?warning=yes

  If any, the warnings are displayed in the response payload with the HTTP code "209 X-NITRO-WARNING".

- **Authenticated access for individual NITRO operations:**NITRO allows you to logon to the Citrix ADC appliance to perform individual operations. You can use this option instead of creating a NITRO session (using the login object) and then using that session to perform all operations,

  To do this, you must specify the username and password in the request header of the NITRO request as follows:

  X-NITRO-USER:<username>

  X-NITRO-PASS:<password>

  Note: In such cases, make sure that the request header DOES not include the following:

  Cookie:NITRO_AUTH_TOKEN=<tokenvalue>

`*Note:`*

Mandatory parameters are marked in redand placeholder content is marked in <green>.

add

URL:http://<Citrix-ADC-IP-address(NSIP)/nitro/v1/config/contentinspectionpolicylabel_contentinspectionpolicy_binding

HTTP Method:PUT

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Content-Type:application/json

Request Payload:

```
{

"contentinspectionpolicylabel_contentinspectionpolicy_binding":{

<b>"labelname":<String_value>,

</b><b>"policyname":<String_value>,

</b><b>"priority":<Double_value>,
```

```
</b>"gotopriorityexpression":<String_value>,

"invoke":<Boolean_value>,

"labeltype":<String_value>,

"invoke_labelname":<String_value>

}}
```

Response:

HTTP Status Code on Success: 201 Created HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error

delete

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/contentinspectionpolicylabel_contentinspectionpolicy_binding/labelname_value<String>

Query-parameters:

args

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/contentinspectionpolicylabel_contentinspectionpolicy_binding/labelname_value<String>?**args=policyname:<String_value>,priority:<Double_value>**

HTTP Method:DELETE

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue>

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error

get

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/contentinspectionpolicylabel_contentinspectionpolicy_binding/labelname_value<String>

Query-parameters:

filter

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/contentinspectionpolicylabel_contentinspectionpolicy_binding/labelname_value<String>?**filter=property-name1:property-value1,property-name2:property-value2**

Use this query-parameter to get the filtered set of contentinspectionpolicylabel_contentinspectionpolicy_binding resources configured on Citrix ADC. Filtering can be done on any of the properties of the resource.

pagination

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/contentinspectionpolicylabel_contentinspectionpolicy_binding/labelname_value<String>?**pagesize=#no;pageno=#no**

Use this query-parameter to get the contentinspectionpolicylabel_contentinspectionpolicy_binding resources in chunks.

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "contentinspectionpolicylabel_contentinspectionpolicy_binding": [ {

"priority":<Double_value>,

"gotopriorityexpression":<String_value>,

"policyname":<String_value>,

"labeltype":<String_value>,

"labelname":<String_value>,

"invoke_labelname":<String_value>,

"invoke":<Boolean_value>

}]}
```

get (all)

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/contentinspectionpolicylabel_contentinspectionpolicy_binding

Query-parameters:

bulkbindings

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/contentinspectionpolicylabel_contentinspectionpolicy_binding?**bulkbindings=yes**

NITRO allows you to fetch bindings in bulk.

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "contentinspectionpolicylabel_contentinspectionpolicy_binding": [ {

"priority":<Double_value>,

"gotopriorityexpression":<String_value>,

"policyname":<String_value>,

"labeltype":<String_value>,

"labelname":<String_value>,

"invoke_labelname":<String_value>,

"invoke":<Boolean_value>

}]}
```

count

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/contentinspectionpolicylabel_contentinspectionpolicy_binding/labelname_value<String>?**count=yes**

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{"contentinspectionpolicylabel_contentinspectionpolicy_binding":
[ { "__count": "#no"} ] }
```

contentinspectionpolicylabel_policybinding_binding

Binding object showing the policybinding that can be bound to contentinspectionpolicylabel.

Properties

(click to see Operations)

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
priority	<Double>	Read-write	Specifies the priority of the policy.
gotopriorityexpression	<String>	Read-write	Expression specifying the priority of the next policy which will get evaluated if the current policy rule evaluates to TRUE.
policyname	<String>	Read-write	Name of the contentInspection policy to bind to the policy label.
labeltype	<String>	Read-write	Type of invocation. Available settings function as follows: *

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
			reqvserver - Forward the request to the specified request virtual server. * resvserver - Forward the response to the specified response virtual server. * policylabel - Invoke the specified policy label. Possible values = reqvserver, resvserver, policylabel
labelname	<String>	Read-write	Name of the contentInspection policy label to which to bind the policy.
invoke_labelname	<String>	Read-write	* If labelType is policylabel, name of the policy label to invoke. * If labelType is reqvserver or resvserver, name of the virtual server to which to forward the request or response.
invoke	<Boolean>	Read-write	Suspend evaluation of policies bound to the current policy label, and then forward the request to the specified virtual server or evaluate the specified policy label.

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
__count	<Double>	Read-write	count parameter

Operations

(click to see Properties)

GET| GET (ALL)| COUNT

Some options that you can use for each operations:

- **Getting warnings in response:**NITRO allows you to get warnings in an operation by specifying the "warning" query parameter as "yes". For example, to get warnings while connecting to the Citrix ADC appliance, the URL is as follows:

  http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/login?warning=yes

  If any, the warnings are displayed in the response payload with the HTTP code "209 X-NITRO-WARNING".
- **Authenticated access for individual NITRO operations:**NITRO allows you to logon to the Citrix ADC appliance to perform individual operations. You can use this option instead of creating a NITRO session (using the login object) and then using that session to perform all operations,

  To do this, you must specify the username and password in the request header of the NITRO request as follows:

  X-NITRO-USER:<username>

  X-NITRO-PASS:<password>

  Note: In such cases, make sure that the request header DOES not include the following:

  Cookie:NITRO_AUTH_TOKEN=<tokenvalue>

`*Note:`*

Mandatory parameters are marked in redand placeholder content is marked in <green>.

get

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/contentinspectionpolicylabel_policybinding_binding/labelname_value<String>

Query-parameters:

filter

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/contentinspectionpolicylabel_policybinding_binding/labelname_value<String>?**filter=property-name1:property-value1,property-name2:property-value2**

Use this query-parameter to get the filtered set of contentinspectionpolicylabel_policybinding_binding resources configured on Citrix ADC. Filtering can be done on any of the properties of the resource.

pagination

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/contentinspectionpolicylabel_policybinding_binding/labelname_value<String>?**pagesize=#no;pageno=#no**

Use this query-parameter to get the contentinspectionpolicylabel_policybinding_binding resources in chunks.

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "contentinspectionpolicylabel_policybinding_binding": [ {

"priority":<Double_value>,

"gotopriorityexpression":<String_value>,
```

```
"policyname":<String_value>,

"labeltype":<String_value>,

"labelname":<String_value>,

"invoke_labelname":<String_value>,

"invoke":<Boolean_value>

}]}
```

get (all)

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/contentinspectionpolicylabel_policybinding_binding

Query-parameters:

bulkbindings

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/contentinspectionpolicylabel_policybinding_binding?**bulkbindings=yes**

NITRO allows you to fetch bindings in bulk.

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "contentinspectionpolicylabel_policybinding_binding": [ {

"priority":<Double_value>,

"gotopriorityexpression":<String_value>,

"policyname":<String_value>,

"labeltype":<String_value>,

"labelname":<String_value>,

"invoke_labelname":<String_value>,

"invoke":<Boolean_value>

}]}
```

count

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/contentinspectionpolicylabel_policybinding_binding/labelname_value<String>?**count=yes**

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{"contentinspectionpolicylabel_policybinding_binding": [ { "__count":
"#no"} ] }
```


contentinspectionprofile

Configuration for ContentInspection profile resource.

Properties

(click to see Operations)

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
name	<String>	Read-write	Name of a ContentInspection profile. Must begin with a letter, number, or the underscore \(_\) character. Other characters allowed, after the first character, are the hyphen \(-\), period \(.\), hash \(#\), space \(\), at \(@\), colon \(:\), and equal \(=\) characters. The name of a IPS profile cannot be changed after it is created. CLI Users: If the name includes one or more spaces, enclose the name in double or single quotation marks \(for example, "my

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
			ips profile" or 'my ips profile'\). Minimum length = 1 Maximum length = 127
type	<String>	Read-write	Type of ContentInspection profile. Following types are available to configure: INLINEINSPECTION : To inspect the packets/requests using IPS. MIRROR : To forward cloned packets. Possible values = InlineInspection, Mirror
ingressinterface	<String>	Read-write	Ingress interface for CI profile.It is a mandatory argument while creating an ContentInspection profile of IPS type.
ingressvlan	<Double>	Read-write	Ingress Vlan for CI.
egressinterface	<String>	Read-write	Egress interface for CI profile.It is a mandatory argument while creating an ContentInspection profile of type

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
			INLINEINSPECTION or MIRROR.
egressvlan	<Double>	Read-write	Egress Vlan for CI.
__count	<Double>	Read-only	count parameter

Operations

(click to see Properties)

ADD| DELETE| UPDATE| UNSET| GET (ALL)| GET| COUNT

Some options that you can use for each operations:

- **Getting warnings in response:**NITRO allows you to get warnings in an operation by specifying the "warning" query parameter as "yes". For example, to get warnings while connecting to the Citrix ADC appliance, the URL is as follows:

  http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/login?warning=yes

  If any, the warnings are displayed in the response payload with the HTTP code "209 X-NITRO-WARNING".

- **Authenticated access for individual NITRO operations:**NITRO allows you to logon to the Citrix ADC appliance to perform individual operations. You can use this option instead of creating a NITRO session (using the login object) and then using that session to perform all operations,

  To do this, you must specify the username and password in the request header of the NITRO request as follows:

  X-NITRO-USER:<username>

  X-NITRO-PASS:<password>

  Note: In such cases, make sure that the request header DOES not include the following:

  Cookie:NITRO_AUTH_TOKEN=<tokenvalue>

`*Note:` *

Mandatory parameters are marked in redand placeholder content is marked in <green>.

add

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/contentinspectionprofile

HTTP Method:POST

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Content-Type:application/json

Request Payload:

```
{"contentinspectionprofile":{

<b>"name":<String_value>,

</b><b>"type":<String_value>,

</b>"ingressinterface":<String_value>,

"ingressvlan":<Double_value>,

<b>"egressinterface":<String_value>,

</b>"egressvlan":<Double_value>

}}
```

Response:

HTTP Status Code on Success: 201 Created HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error

delete

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/contentinspectionprofile/name_value<String>

HTTP Method:DELETE

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue>

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error

update

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/contentinspectionprofile

HTTP Method:PUT

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Content-Type:application/json

Request Payload:

```
{"contentinspectionprofile":{

<b>"name":<String_value>,

</b>"egressinterface":<String_value>,

"ingressinterface":<String_value>,

"egressvlan":<Double_value>,

"ingressvlan":<Double_value>

}}
```

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error

unset

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/contentinspectionprofile?**action=unset**

HTTP Method:POST

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Content-Type:application/json

Request Payload:

```
{"contentinspectionprofile":{

<b>"name":<String_value>,

</b>"egressvlan":true,

"ingressvlan":true

}}
```

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error

get (all)

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/contentinspectionprofile

Query-parameters:

attrs

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/contentinspectionprofile?**attrs=property-name1,property-name2**

Use this query parameter to specify the resource details that you want to retrieve.

filter

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/contentinspectionprofile?**filter=property-name1:property-val1,property-name2:property-val2**

Use this query-parameter to get the filtered set of contentinspectionprofile resources configured on Citrix ADC. Filtering can be done on any of the properties of the resource.

view

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/contentinspectionprofile?**view=summary**

Note: By default, the retrieved results are displayed in detail view (?view=detail).

pagination

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/contentinspectionprofile?**pagesize=#no;pageno=#no**

Use this query-parameter to get the contentinspectionprofile resources in chunks.

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "contentinspectionprofile": [ {

"name":<String_value>,

"type":<String_value>,

"egressinterface":<String_value>,
```

```
"ingressinterface":<String_value>,

"egressvlan":<Double_value>,

"ingressvlan":<Double_value>

}]}
```

get

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/contentinspectionprofile/name_value<String>

Query-parameters:

attrs

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/contentinspectionprofile/name_value<String>?**attrs=property-name1,property-name2**

Use this query parameter to specify the resource details that you want to retrieve.

view

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/contentinspectionprofile/name_value<String>?**view=summary**

Note: By default, the retrieved results are displayed in detail view (?view=detail).

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "contentinspectionprofile": [ {

"name":<String_value>,

"type":<String_value>,

"egressinterface":<String_value>,

"ingressinterface":<String_value>,

"egressvlan":<Double_value>,

"ingressvlan":<Double_value>

}]}
```

count

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/contentinspectionprofile?**count=yes**

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "contentinspectionprofile": [ { "__count": "#no"} ] }
```

Cloud

Cloud

Citrix ADC as SD proxy Configuration and cloud discovery commands.

NAME	DESCRIPTION
cloudautoscalegroup	Configuration for Cloud autoscalegroup
cloudcredential	Configuration for cloud credentials
cloudparameter	Configuration for cloud parameter
cloudparaminternal	Configuration for cloud paramInternal
cloudprofile	Configuration for cloud profile
cloudvserverip	Configuration for Cloud virtual server IPs

cloudcredential

Configuration for cloud credentials resource.

Properties

(click to see Operations)

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
tenantidentifier	<String>	Read-write	Tenant ID of the Credentials. Minimum length = 1
applicationid	<String>	Read-write	Application ID of the Credentials. Minimum length = 1

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
applicationsecret	<String>	Read-write	Application Secret of the Credentials. Minimum length = 1
isset	<Integer>	Read-only	Are the credentials set?.

Operations

(click to see Properties)

UPDATE| GET (ALL)

Some options that you can use for each operations:

- **Getting warnings in response:**NITRO allows you to get warnings in an operation by specifying the "warning" query parameter as "yes". For example, to get warnings while connecting to the Citrix ADC appliance, the URL is as follows:

  http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/login?warning=yes

  If any, the warnings are displayed in the response payload with the HTTP code "209 X-NITRO-WARNING".

- **Authenticated access for individual NITRO operations:**NITRO allows you to logon to the Citrix ADC appliance to perform individual operations. You can use this option instead of creating a NITRO session (using the login object) and then using that session to perform all operations,

  To do this, you must specify the username and password in the request header of the NITRO request as follows:

  X-NITRO-USER:<username>

  X-NITRO-PASS:<password>

  Note: In such cases, make sure that the request header DOES not include the following:

  Cookie:NITRO_AUTH_TOKEN=<tokenvalue>

`*Note:`*

Mandatory parameters are marked in redand placeholder content is marked in <green>.

update

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/cloudcredential

HTTP Method:PUT

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Content-Type:application/json

Request Payload:

```
{"cloudcredential":{

<b>"tenantidentifier":<String_value>,

</b><b>"applicationid":<String_value>,

</b><b>"applicationsecret":<String_value>

</b>}}
```

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error

get (all)

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/cloudcredential

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "cloudcredential": [ {

"isset":<Integer_value>

}]}
```

cloudparameter

Configuration for cloud parameter resource.

Properties

(click to see Operations)

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
controllerfqdn	<String>	Read-write	FQDN of the controller to which the Citrix ADC SDProxy Connects. Minimum length = 1
controllerport	<Integer>	Read-write	Port number of the controller to which the Citrix ADC SDProxy connects. Minimum value = 1 Range 1 - 65535

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
			* in CLI is represented as 65535 in NITRO API
instanceid	<String>	Read-write	Instance ID of the customer provided by Trust. Minimum length = 1
customerid	<String>	Read-write	Customer ID of the citrix cloud customer. Minimum length = 1
resourcelocation	<String>	Read-write	Resource Location of the customer provided by Trust. Minimum length = 1
activationcode	<String>	Read-write	Activation code for the NGS Connector instance.
deployment	<String>	Read-write	Describes if the customer is a Staging/Production or Dev Citrix Cloud customer. Possible values = , Production, Staging, Dev
connectorresidence	<String>	Read-write	Identifies whether the connector is located Onprem, Aws or Azure. Possible values = None, Onprem, Aws, Azure, Cpx

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
controlconnectionstatus	<String>	Read-only	Status of the control connection: In case of successful registration to the controller, connection status will be shown as "Registered" else "Unregistered".

Operations

(click to see Properties)

UPDATE| UNSET| GET (ALL)

Some options that you can use for each operations:

- **Getting warnings in response:**NITRO allows you to get warnings in an operation by specifying the "warning" query parameter as "yes". For example, to get warnings while connecting to the Citrix ADC appliance, the URL is as follows:

  http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/login?warning=yes

  If any, the warnings are displayed in the response payload with the HTTP code "209 X-NITRO-WARNING".
- **Authenticated access for individual NITRO operations:**NITRO allows you to logon to the Citrix ADC appliance to perform individual operations. You can use this option instead of creating a NITRO session (using the login object) and then using that session to perform all operations,

  To do this, you must specify the username and password in the request header of the NITRO request as follows:

  X-NITRO-USER:<username>

  X-NITRO-PASS:<password>

  Note: In such cases, make sure that the request header DOES not include the following:

  Cookie:NITRO_AUTH_TOKEN=<tokenvalue>

`*Note:` *

Mandatory parameters are marked in redand placeholder content is marked in <green>.

update

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/cloudparameter

HTTP Method:PUT

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Content-Type:application/json

Request Payload:

```
{"cloudparameter":{

"controllerfqdn":<String_value>,

"controllerport":<Integer_value>,

"instanceid":<String_value>,

"customerid":<String_value>,

"resourcelocation":<String_value>,

"activationcode":<String_value>,

"deployment":<String_value>,

"connectorresidence":<String_value>

}}
```

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error

unset

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/cloudparameter?**action=unset**

HTTP Method:POST

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Content-Type:application/json

Request Payload:

```
{"cloudparameter":{

"instanceid":true,

"customerid":true,

"resourcelocation":true

}}
```

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error

get (all)

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/cloudparameter

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "cloudparameter": [ {

"controllerfqdn":<String_value>,

"controllerport":<Integer_value>,

"instanceid":<String_value>,

"customerid":<String_value>,

"resourcelocation":<String_value>,

"controlconnectionstatus":<String_value>,

"deployment":<String_value>,

"connectorresidence":<String_value>

}]}
```

cloudautoscalegroup

Configuration for Cloud autoscalegroup resource.

Properties

(click to see Operations)

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
name	<String>	Read-write	.

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
azcount	<Integer>	Read-only	Number of Availibility Zone this Autoscale Group is in.
aznames	<String[]>	Read-only	Availibility Zones this Autoscale Group is in.
graceful	<String>	Read-only	Autoscale down policy Set for this Autoscale group. Possible values = YES, NO
__count	<Double>	Read-only	count parameter

Operations

(click to see Properties)

GET (ALL)| GET| COUNT

Some options that you can use for each operations:

- **Getting warnings in response:**NITRO allows you to get warnings in an operation by specifying the "warning" query parameter as "yes". For example, to get warnings while connecting to the Citrix ADC appliance, the URL is as follows:

  http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/login?warning=yes

  If any, the warnings are displayed in the response payload with the HTTP code "209 X-NITRO-WARNING".

- **Authenticated access for individual NITRO operations:**NITRO allows you to logon to the Citrix ADC appliance to perform individual operations. You can use this option instead of creating a NITRO session (using the login object) and then using that session to perform all operations,

  To do this, you must specify the username and password in the request header of the NITRO request as follows:

  X-NITRO-USER:<username>

X-NITRO-PASS:<password>

Note: In such cases, make sure that the request header DOES not include the following:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue>

`*Note:`*

Mandatory parameters are marked in redand placeholder content is marked in <green>.

get (all)

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/cloudautoscalegroup

Query-parameters:

attrs

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/cloudautoscalegroup?**attrs=property-name1,property-name2**

Use this query parameter to specify the resource details that you want to retrieve.

filter

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/cloudautoscalegroup?**filter=property-name1:property-val1,property-name2:property-val2**

Use this query-parameter to get the filtered set of cloudautoscalegroup resources configured on Citrix ADC. Filtering can be done on any of the properties of the resource.

view

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/cloudautoscalegroup?**view=summary**

Note: By default, the retrieved results are displayed in detail view (?view=detail).

pagination

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/cloudautoscalegroup?**pagesize=#no;pageno=#no**

Use this query-parameter to get the cloudautoscalegroup resources in chunks.

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "cloudautoscalegroup": [ {

"name":<String_value>,

"azcount":<Integer_value>,

"aznames":<String[]_value>,

"graceful":<String_value>

}]}
```

get

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/cloudautoscalegroup/name_value<String>

Query-parameters:

attrs

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/cloudautoscalegroup/name_value<String>?**attrs=property-name1,property-name2**

Use this query parameter to specify the resource details that you want to retrieve.

view

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/cloudautoscalegroup/name_value<String>?**view=summary**

Note: By default, the retrieved results are displayed in detail view (?view=detail).

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "cloudautoscalegroup": [ {

"name":<String_value>,

"azcount":<Integer_value>,

"aznames":<String[]_value>,

"graceful":<String_value>

}]}
```

count

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/cloudautoscalegroup?**count=yes**

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "cloudautoscalegroup": [ { "__count": "#no"} ] }
```

cloudparaminternal

Configuration for cloud paramInternal resource.

Properties

(click to see Operations)

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
nonftumode	<String>	Read-write	Indicates if GUI in in FTU mode or not. Possible values = YES, NO
iamperm	<String>	Read-only	Indicates if user has sufficient IAM previliges. Default value: NO Possible values = YES, NO
__count	<Double>	Read-only	count parameter

Operations

(click to see Properties)

UPDATE| GET (ALL)| COUNT

Some options that you can use for each operations:

- **Getting warnings in response:**NITRO allows you to get warnings in an operation by specifying the "warning" query parameter as "yes". For example, to get warnings while connecting to the Citrix ADC appliance, the URL is as follows:

  http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/login?warning=yes

  If any, the warnings are displayed in the response payload with the HTTP code "209 X-NITRO-WARNING".

- **Authenticated access for individual NITRO operations:**NITRO allows you to logon to the Citrix ADC appliance to perform individual operations. You can use this option instead of creating a NITRO session (using the login object) and then using that session to perform all operations,

  To do this, you must specify the username and password in the request header of the NITRO request as follows:

  X-NITRO-USER:<username>

  X-NITRO-PASS:<password>

  Note: In such cases, make sure that the request header DOES not include the following:

  Cookie:NITRO_AUTH_TOKEN=<tokenvalue>

`*Note:`*

Mandatory parameters are marked in redand placeholder content is marked in <green>.

update

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/cloudparaminternal

HTTP Method:PUT

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Content-Type:application/json

Request Payload:

```
{"cloudparaminternal":{

"nonftumode":<String_value>

}}
```

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error

get (all)

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/cloudparaminternal

Query-parameters:

attrs

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/cloudparaminternal?**attrs=property-name1,property-name2**

Use this query parameter to specify the resource details that you want to retrieve.

filter

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/cloudparaminternal?**filter=property-name1:property-val1,property-name2:property-val2**

Use this query-parameter to get the filtered set of cloudparaminternal resources configured on Citrix ADC. Filtering can be done on any of the properties of the resource.

view

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/cloudparaminternal?**view=summary**

Note: By default, the retrieved results are displayed in detail view (?view=detail).

pagination

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/cloudparaminternal?**pagesize=#no;pageno=#no**

Use this query-parameter to get the cloudparaminternal resources in chunks.

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "cloudparaminternal": [ {

"nonftumode":<String_value>,

"iamperm":<String_value>

}]}
```

count

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/cloudparaminternal?**count=yes**

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "cloudparaminternal": [ { "__count": "#no"} ] }
```

cloudprofile

Configuration for cloud profile resource.

Properties

(click to see Operations)

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
name	<String>	Read-write	Name for the Cloud profile. Must begin with an ASCII alphanumeric or underscore (_) character, and must contain only ASCII alphanumeric, underscore, hash (#), period (.), space, colon (:), at (@), equals (=), and hyphen (-) characters. Cannot be changed after the profile is created. Minimum length = 1

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
			Maximum length = 127
type	<String>	Read-write	Type of cloud profile that you want to create, stats or Vserver. Possible values = autoscale, stats
vservername	<String>	Read-write	Name for the virtual server. Must begin with an ASCII alphanumeric or underscore (_) character, and must contain only ASCII alphanumeric, underscore, hash (#), period (.), space, colon (:), at sign (@), equal sign (=), and hyphen (-) characters. Can be changed after the virtual server is created. CLI Users: If the name includes one or more spaces, enclose the name in double or single quotation marks (for example, "my vserver" or 'my vserver'). . Minimum length = 1

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
servicetype	<String>	Read-write	Protocol used by the service (also called the service type). Possible values = HTTP, FTP, TCP, UDP, SSL, SSL_BRIDGE, SSL_TCP, DTLS, NNTP, DNS, DHCPRA, ANY, SIP_UDP, SIP_TCP, SIP_SSL, DNS_TCP, RTSP, PUSH, SSL_PUSH, RADIUS, RDP, MYSQL, MSSQL, DIAMETER, SSL_DIAMETER, TFTP, ORACLE, SMPP, SYSLOGTCP, SYSLOGUDP, FIX, SSL_FIX, PROXY, USER_TCP, USER_SSL_TCP, QUIC, IPFIX, LOGSTREAM
ipaddress	<String>	Read-write	IPv4 or IPv6 address to assign to the virtual server.
port	<Integer>	Read-write	Port number for the virtual server. Range 1 - 65535

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
			* in CLI is represented as 65535 in NITRO API
servicegroupname	<String>	Read-write	servicegroups bind to this server.
boundservicegroups vctype	<String>	Read-write	The type of bound service. Possible values = HTTP, FTP, TCP, UDP, SSL, SSL_BRIDGE, SSL_TCP, DTLS, NNTP, RPCSVR, DNS, ADNS, SNMP, RTSP, DHCPRA, ANY, SIP_UDP, SIP_TCP, SIP_SSL, DNS_TCP, ADNS_TCP, MYSQL, MSSQL, ORACLE, RADIUS, RADIUSListener, RDP, DIAMETER, SSL_DIAMETER, TFTP, SMPP, PPTP, GRE, SYSLOGTCP, SYSLOGUDP, FIX, SSL_FIX, USER_TCP, USER_SSL_TCP, QUIC, IPFIX, LOGSTREAM

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
vsvrbindsvcport	<Integer>	Read-write	The port number to be used for the bound service. Range 1 - 65535 * in CLI is represented as 65535 in NITRO API
graceful	<String>	Read-write	Indicates graceful shutdown of the service. System will wait for all outstanding connections to this service to be closed before disabling the service. Default value: NO Possible values = YES, NO
delay	<Double>	Read-write	Time, in seconds, after which all the services configured on the server are disabled.
__count	<Double>	Read-only	count parameter

Operations

(click to see Properties)

ADD| DELETE| GET (ALL)| GET| COUNT

Some options that you can use for each operations:

- **Getting warnings in response:**NITRO allows you to get warnings in an operation by specifying the "warning" query parameter as "yes". For example, to get warnings while connecting to the Citrix ADC appliance, the URL is as follows:

  http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/login?warning=yes

  If any, the warnings are displayed in the response payload with the HTTP code "209 X-NITRO-WARNING".

- **Authenticated access for individual NITRO operations:**NITRO allows you to logon to the Citrix ADC appliance to perform individual operations. You can use this option instead of creating a NITRO session (using the login object) and then using that session to perform all operations,

  To do this, you must specify the username and password in the request header of the NITRO request as follows:

  X-NITRO-USER:<username>

  X-NITRO-PASS:<password>

  Note: In such cases, make sure that the request header DOES not include the following:

  Cookie:NITRO_AUTH_TOKEN=<tokenvalue>

`*Note:`*

Mandatory parameters are marked in redand placeholder content is marked in <green>.

add

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/cloudprofile

HTTP Method:POST

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Content-Type:application/json

Request Payload:

```
{"cloudprofile":{
```

```
<b>"name":<String_value>,

</b><b>"type":<String_value>,

</b><b>"vservername":<String_value>,

</b><b>"servicetype":<String_value>,

</b><b>"ipaddress":<String_value>,

</b><b>"port":<Integer_value>,

</b><b>"servicegroupname":<String_value>,

</b><b>"boundservicegroupsvctype":<String_value>,

</b><b>"vsvrbindsvcport":<Integer_value>,

</b>"graceful":<String_value>,

"delay":<Double_value>

}}
```

Response:

HTTP Status Code on Success: 201 Created HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error

delete

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/cloudprofile/name_value<String>

HTTP Method:DELETE

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue>

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error

get (all)

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/cloudprofile

Query-parameters:

attrs

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/cloudprofile?**attrs=property-name1,property-name2**

Use this query parameter to specify the resource details that you want to retrieve.

filter

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/cloudprofile?**filter=property-name1:property-val1,property-name2:property-val2**

Use this query-parameter to get the filtered set of cloudprofile resources configured on Citrix ADC. Filtering can be done on any of the properties of the resource.

view

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/cloudprofile?**view=summary**

Note: By default, the retrieved results are displayed in detail view (?view=detail).

pagination

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/cloudprofile?**pagesize=#no;pageno=#no**

Use this query-parameter to get the cloudprofile resources in chunks.

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "cloudprofile": [ {

"name":<String_value>,

"type":<String_value>,

"vservername":<String_value>,

"servicetype":<String_value>,

"ipaddress":<String_value>,

"port":<Integer_value>,

"servicegroupname":<String_value>,

"boundservicegroupsvctype":<String_value>,

"vsvrbindsvcport":<Integer_value>,

"graceful":<String_value>,

"delay":<Double_value>

}]}
```

get

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/cloudprofile/name_value<String>

Query-parameters:

attrs

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/cloudprofile/name_value<String>?**attrs=property-name1,property-name2**

Use this query parameter to specify the resource details that you want to retrieve.

view

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/cloudprofile/name_value<String>?**view=summary**

Note: By default, the retrieved results are displayed in detail view (?view=detail).

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "cloudprofile": [ {

"name":<String_value>,

"type":<String_value>,

"vservername":<String_value>,

"servicetype":<String_value>,

"ipaddress":<String_value>,

"port":<Integer_value>,
```

```
"servicegroupname":<String_value>,

"boundservicegroupsvctype":<String_value>,

"vsvrbindsvcport":<Integer_value>,

"graceful":<String_value>,

"delay":<Double_value>

}]}
```

count

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/cloudprofile?**count=yes**

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "cloudprofile": [ { "__count": "#no"} ] }
```

cloudvserverip

Configuration for Cloud virtual server IPs resource.

Properties

(click to see Operations)

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
ipaddress	<String>	Read-only	VIPS on a given ENI.
__count	<Double>	Read-only	count parameter

Operations

(click to see Properties)

GET (ALL)| COUNT

Some options that you can use for each operations:

- **Getting warnings in response:**NITRO allows you to get warnings in an operation by specifying the "warning" query parameter as "yes". For example, to get warnings while connecting to the Citrix ADC appliance, the URL is as follows:

  http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/login?warning=yes

  If any, the warnings are displayed in the response payload with the HTTP code "209 X-NITRO-WARNING".

- **Authenticated access for individual NITRO operations:**NITRO allows you to logon to the Citrix ADC appliance to perform individual operations. You can use this option instead of creating a NITRO session (using the login object) and then using that session to perform all operations,

  To do this, you must specify the username and password in the request header of the NITRO request as follows:

  X-NITRO-USER:<username>

  X-NITRO-PASS:<password>

  Note: In such cases, make sure that the request header DOES not include the following:

  Cookie:NITRO_AUTH_TOKEN=<tokenvalue>

`*Note:`*

Mandatory parameters are marked in redand placeholder content is marked in <green>.

get (all)

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/cloudvserverip

Query-parameters:

attrs

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/cloudvserverip?**attrs=property-name1,property-name2**

Use this query parameter to specify the resource details that you want to retrieve.

filter

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/cloudvserverip?**filter=property-name1:property-val1,property-name2:property-val2**

Use this query-parameter to get the filtered set of cloudvserverip resources configured on Citrix ADC. Filtering can be done on any of the properties of the resource.

view

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/cloudvserverip?**view=summary**

Note: By default, the retrieved results are displayed in detail view (?view=detail).

pagination

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/cloudvserverip?**pagesize=#no;pageno=#no**

Use this query-parameter to get the cloudvserverip resources in chunks.

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "cloudvserverip": [ {

"ipaddress":<String_value>

}]}
```

count

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/cloudvserverip?**count=yes**

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "cloudvserverip": [ { "__count": "#no"} ] }
```

Cluster

Cluster

Cluster commands.

NAME	DESCRIPTION
cluster	
clusterfiles	Configuration for files
clusterinstance	Configuration for cluster instance
clusterinstance_binding	Binding object showing the resources that can be bound to clusterinstance
clusterinstance_clusternode_binding	Binding object showing the clusternode that can be bound to clusterinstance
clusternode	Configuration for cluster node
clusternode_binding	Binding object showing the resources that can be bound to clusternode
clusternode_routemonitor_binding	Binding object showing the routemonitor that can be bound to clusternode
clusternodegroup	Configuration for Node group object type
clusternodegroup_authenticationvserver_binding	Binding object showing the authenticationvserver that can be bound to clusternodegroup
clusternodegroup_binding	Binding object showing the resources that can be bound to clusternodegroup
clusternodegroup_clusternode_binding	Binding object showing the clusternode that can be bound to clusternodegroup
clusternodegroup_crvserver_binding	Binding object showing the crvserver that can be bound to clusternodegroup
clusternodegroup_csvserver_binding	Binding object showing the csvserver that can be bound to clusternodegroup
clusternodegroup_gslbsite_binding	Binding object showing the gslbsite that can be bound to clusternodegroup

NAME	DESCRIPTION
clusternodegroup_gslbvserver_binding	Binding object showing the gslbvserver that can be bound to clusternodegroup
clusternodegroup_lbvserver_binding	Binding object showing the lbvserver that can be bound to clusternodegroup
clusternodegroup_nslimitidentifier_binding	Binding object showing the nslimitidentifier that can be bound to clusternodegroup
clusternodegroup_service_binding	Binding object showing the service that can be bound to clusternodegroup
clusternodegroup_streamidentifier_binding	Binding object showing the streamidentifier that can be bound to clusternodegroup
clusternodegroup_vpnvserver_binding	Binding object showing the vpnvserver that can be bound to clusternodegroup
clusterpropstatus	
clustersync	Configuration for sync

cluster

Configuration for 0 resource.

Properties

(click to see Operations)

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
clip	<String>	Read-write	Cluster IP address to which to add the node.
password	<String>	Read-write	Password for the nsroot account of the configuration coordinator (CCO).

Operations

(click to see Properties)

JOIN

Some options that you can use for each operations:

- **Getting warnings in response:**NITRO allows you to get warnings in an operation by specifying the "warning" query parameter as "yes". For example, to get warnings while connecting to the Citrix ADC appliance, the URL is as follows:

  http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/login?warning=yes

  If any, the warnings are displayed in the response payload with the HTTP code "209 X-NITRO-WARNING".

- **Authenticated access for individual NITRO operations:**NITRO allows you to logon to the Citrix ADC appliance to perform individual operations. You can use this option instead of creating a NITRO session (using the login object) and then using that session to perform all operations,

  To do this, you must specify the username and password in the request header of the NITRO request as follows:

  X-NITRO-USER:<username>

  X-NITRO-PASS:<password>

  Note: In such cases, make sure that the request header DOES not include the following:

  Cookie:NITRO_AUTH_TOKEN=<tokenvalue>

`*Note:`*

Mandatory parameters are marked in redand placeholder content is marked in <green>.

join

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/cluster?**action=join**

HTTP Method:POST

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Content-Type:application/json

Request Payload:

```
{"cluster":{

<b>"clip":<String_value>,

</b><b>"password":<String_value>

</b>}}
```

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error

clusterfiles

Configuration for files resource.

Properties

(click to see Operations)

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
mode	<String[]>	Read-write	The directories and files to be synchronized. The available settings function as follows: Mode Paths all /nsconfig/ssl/ /var/netscaler/ssl/ /var/vpn/bookmark/

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
			/nsconfig/dns/ /nsconfig/htmlinjection/ /netscaler/htmlinjection/ens/ /nsconfig/monitors/ /nsconfig/nstemplates/ /nsconfig/ssh/ /nsconfig/rc.Citrix ADC /nsconfig/resolv.conf /nsconfig/inetd.conf /nsconfig/syslog.conf /nsconfig/ntp.conf /nsconfig/httpd.conf /nsconfig/sshd_config /nsconfig/hosts /nsconfig/enckey /var/nslw.bin/etc/krb5.conf /var/nslw.bin/etc/krb5.keytab

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
			/var/lib/likewise/db/ /var/download/ /var/wi/tomcat/webapps/ /var/wi/tomcat/conf/Catalina/localhost/ /var/wi/java_home/lib/security/cacerts /var/wi/java_home/jre/lib/security/cacerts /var/netscaler/locdb/ ssl /nsconfig/ssl/ /var/netscaler/ssl/ bookmarks /var/vpn/bookmark/ dns /nsconfig/dns/ htmlinjection /nsconfig/htmlinjection/ imports /var/download/ misc /nsconfig/license/ /nsconfig/rc.conf

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
			all_plus_misc Includes *all* files and /nsconfig/license/ and /nsconfig/rc.conf. Default value: all. Possible values = all, bookmarks, ssl, htmlinjection, imports, misc, dns, krb, AAA, app_catalog, all_plus_misc, all_minus_misc

Operations

(click to see Properties)

SYNC

Some options that you can use for each operations:

- **Getting warnings in response:**NITRO allows you to get warnings in an operation by specifying the "warning" query parameter as "yes". For example, to get warnings while connecting to the Citrix ADC appliance, the URL is as follows:

  http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/login?warning=yes

  If any, the warnings are displayed in the response payload with the HTTP code "209 X-NITRO-WARNING".
- **Authenticated access for individual NITRO operations:**NITRO allows you to logon to the Citrix ADC appliance to perform individual operations. You can use this option instead of creating a NITRO session (using the login object) and then using that session to perform all operations,

To do this, you must specify the username and password in the request header of the NITRO request as follows:

X-NITRO-USER:<username>

X-NITRO-PASS:<password>

Note: In such cases, make sure that the request header DOES not include the following:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue>

`*Note:`*

Mandatory parameters are marked in redand placeholder content is marked in <green>.

sync

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/clusterfiles?**action=sync**

HTTP Method:POST

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Content-Type:application/json

Request Payload:

```
{"clusterfiles":{

"mode":<String[]_value>

}}
```

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error

clusterinstance

Configuration for cluster instance resource.

Properties

(click to see Operations)

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
clid	<Double>	Read-write	Unique number that identifies the cluster. Minimum value = 1 Maximum value = 16
deadinterval	<Double>	Read-write	Amount of time, in seconds, after which nodes that do not respond to the heartbeats are assumed to be down.If the value is less than 3 sec, set the helloInterval parameter to 200 msec. Default value: 3 Minimum value = 1 Maximum value = 60
hellointerval	<Double>	Read-write	Interval, in milliseconds, at which heartbeats are sent to each cluster node to check the health status.Set the value to 200 msec, if the deadInterval parameter is less than 3 sec. Default value: 200

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
			Minimum value = 200 Maximum value = 1000
preemption	<String>	Read-write	Preempt a cluster node that is configured as a SPARE if an ACTIVE node becomes available. Default value: DISABLED Possible values = ENABLED, DISABLED
quorumtype	<String>	Read-write	Quorum Configuration Choices - "Majority" (recommended) requires majority of nodes to be online for the cluster to be UP. "None" relaxes this requirement. Default value: MAJORITY Possible values = MAJORITY, NONE
inc	<String>	Read-write	This option is required if the cluster nodes reside on different networks.

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
			Default value: DISABLED Possible values = ENABLED, DISABLED
processlocal	<String>	Read-write	By turning on this option packets destined to a service in a cluster will not under go any steering. Default value: DISABLED Possible values = ENABLED, DISABLED
retainconnectionsoncluster	<String>	Read-write	This option enables you to retain existing connections on a node joining a Cluster system or when a node is being configured for passive timeout. By default, this option is disabled. Default value: NO Possible values = YES, NO
nodegroup	<String>	Read-write	The node group in a Cluster system used for transition from L2 to L3.

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
adminstate	<String>	Read-only	Cluster Admin State. Default value: DISABLED Possible values = ENABLED, DISABLED
propstate	<String>	Read-only	Enable/Disable the execution of commands on the cluster. This will not impact the execution of commands on individual cluster nodes by using the NSIP. Default value: ENABLED Possible values = ENABLED, DISABLED
validmtu	<Double>	Read-only	Correct MTU value that has to be set on backplane.
heterogeneousflag	<String>	Read-only	This field diplay if heterogeneous is detected in the cluster system. Default value: NO Possible values = YES, NO
operationalstate	<String>	Read-only	Cluster Operational State.

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
			Possible values = ENABLED, DISABLED
status	<String>	Read-only	Cluster Operational State. Possible values = DOWN, UP, PARTIAL-UP, UNKNOWN
rsskeymismatch	<Boolean>	Read-only	This argument is used to determine if there is a RSS key mismatch at cluster instance level.
penummismatch	<Boolean>	Read-only	This argument is used to determine if there is a PE number mismatch at cluster instance level.
nodegroupstatewarning	<Boolean>	Read-only	This argumnt is used to determind whether all the cluster nodes are bound to nodegroup with state set.
licensemismatch	<Boolean>	Read-only	This argument is used to determine if there is a License mismatch at cluster instance level.
jumbonotsupported	<Boolean>	Read-only	This argument is used to determine if Jumbo framework is not supported at

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
			cluster instance level.
clusternoheartbeatonnode	<Boolean>	Read-only	HB is not seen on the backplane interface of member node.
clusternolinksetmbf	<Boolean>	Read-only	MBF is enabled but linkset is not configured .
clusternospottedip	<Boolean>	Read-only	There is no spotted SNIP or MIP.
nodepenummismatch	<Boolean>	Read-only	This argument is used to determine if there is a PE missmatch at cluster node level.
operationalpropstate	<String>	Read-only	Cluster Operational Propagation State. Default value: ENABLED Possible values = UNKNOWN, ENABLED, DISABLED, AUTO DISABLED
__count	<Double>	Read-only	count parameter

Operations

(click to see Properties)

ADD| DELETE| UPDATE| UNSET| ENABLE| DISABLE| GET (ALL)| GET| COUNT

Some options that you can use for each operations:

- **Getting warnings in response:**NITRO allows you to get warnings in an operation by specifying the "warning" query parameter as "yes". For example, to get warnings while connecting to the Citrix ADC appliance, the URL is as follows:

  http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/login?warning=yes

  If any, the warnings are displayed in the response payload with the HTTP code "209 X-NITRO-WARNING".

- **Authenticated access for individual NITRO operations:**NITRO allows you to logon to the Citrix ADC appliance to perform individual operations. You can use this option instead of creating a NITRO session (using the login object) and then using that session to perform all operations,

  To do this, you must specify the username and password in the request header of the NITRO request as follows:

  X-NITRO-USER:<username>

  X-NITRO-PASS:<password>

  Note: In such cases, make sure that the request header DOES not include the following:

  Cookie:NITRO_AUTH_TOKEN=<tokenvalue>

`*Note:`*

Mandatory parameters are marked in redand placeholder content is marked in <green>.

add

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/clusterinstance

HTTP Method:POST

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Content-Type:application/json

Request Payload:

```
{"clusterinstance":{
```

```
<b>"clid":<Double_value>,

</b>"deadinterval":<Double_value>,

"hellointerval":<Double_value>,

"preemption":<String_value>,

"quorumtype":<String_value>,

"inc":<String_value>,

"processlocal":<String_value>,

"retainconnectionsoncluster":<String_value>

}}
```

Response:

HTTP Status Code on Success: 201 Created HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error

delete

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/clusterinstance/clid_value<Double>

HTTP Method:DELETE

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue>

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error

update

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/clusterinstance

HTTP Method:PUT

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Content-Type:application/json

Request Payload:

```
{"clusterinstance":{

<b>"clid":<Double_value>,

</b>"deadinterval":<Double_value>,

"hellointerval":<Double_value>,

"preemption":<String_value>,

"quorumtype":<String_value>,

"inc":<String_value>,

"processlocal":<String_value>,

"nodegroup":<String_value>,

"retainconnectionsoncluster":<String_value>

}}
```

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error

unset

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/clusterinstance?**action=unset**

HTTP Method:POST

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Content-Type:application/json

Request Payload:

```
{"clusterinstance":{

<b>"clid":<Double_value>,

</b>"deadinterval":true,

"hellointerval":true,

"preemption":true,

"quorumtype":true,

"inc":true,

"processlocal":true,

"nodegroup":true,

"retainconnectionsoncluster":true

}}
```

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error

enable

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/clusterinstance?**action=enable**

HTTP Method:POST

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Content-Type:application/json

Request Payload:

```
{"clusterinstance":{

<b>"clid":<Double_value>

</b>}}
```

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error

disable

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/clusterinstance?**action=disable**

HTTP Method:POST

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Content-Type:application/json

Request Payload:

```
{"clusterinstance":{

<b>"clid":<Double_value>

</b>}}
```

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error

get (all)

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/clusterinstance

Query-parameters:

attrs

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/clusterinstance?**attrs=property-name1,property-name2**

Use this query parameter to specify the resource details that you want to retrieve.

filter

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/clusterinstance?**filter=property-name1:property-val1,property-name2:property-val2**

Use this query-parameter to get the filtered set of clusterinstance resources configured on Citrix ADC. Filtering can be done on any of the properties of the resource.

view

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/clusterinstance?**view=summary**

Note: By default, the retrieved results are displayed in detail view (?view=detail).

pagination

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/clusterinstance?**pagesize=#no;pageno=#no**

Use this query-parameter to get the clusterinstance resources in chunks.

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "clusterinstance": [ {

"clid":<Double_value>,

"deadinterval":<Double_value>,

"hellointerval":<Double_value>,

"preemption":<String_value>,

"adminstate":<String_value>,

"quorumtype":<String_value>,

"propstate":<String_value>,

"inc":<String_value>,

"nodegroup":<String_value>,

"processlocal":<String_value>,

"validmtu":<Double_value>,

"retainconnectionsoncluster":<String_value>,

"heterogeneousflag":<String_value>,

"operationalstate":<String_value>,

"status":<String_value>,

"rsskeymismatch":<Boolean_value>,
```

```
"penummismatch":<Boolean_value>,

"nodegroupstatewarning":<Boolean_value>,

"licensemismatch":<Boolean_value>,

"jumbonotsupported":<Boolean_value>,

"clusternoheartbeatonnode":<Boolean_value>,

"clusternolinksetmbf":<Boolean_value>,

"clusternospottedip":<Boolean_value>,

"nodepenummismatch":<Boolean_value>,

"operationalpropstate":<String_value>

}]}
```

get

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/clusterinstance/clid_value<Double>

Query-parameters:

attrs

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/clusterinstance/clid_value<Double>?**attrs=property-name1,property-name2**

Use this query parameter to specify the resource details that you want to retrieve.

view

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/clusterinstance/clid_value<Double>?**view=summary**

Note: By default, the retrieved results are displayed in detail view (?view=detail).

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "clusterinstance": [ {

"clid":<Double_value>,

"deadinterval":<Double_value>,

"hellointerval":<Double_value>,

"preemption":<String_value>,

"adminstate":<String_value>,

"quorumtype":<String_value>,

"propstate":<String_value>,

"inc":<String_value>,

"nodegroup":<String_value>,

"processlocal":<String_value>,

"validmtu":<Double_value>,

"retainconnectionsoncluster":<String_value>,

"heterogeneousflag":<String_value>,

"operationalstate":<String_value>,
```

```
"status":<String_value>,

"rsskeymismatch":<Boolean_value>,

"penummismatch":<Boolean_value>,

"nodegroupstatewarning":<Boolean_value>,

"licensemismatch":<Boolean_value>,

"jumbonotsupported":<Boolean_value>,

"clusternoheartbeatonnode":<Boolean_value>,

"clusternolinksetmbf":<Boolean_value>,

"clusternospottedip":<Boolean_value>,

"nodepenummismatch":<Boolean_value>,

"operationalpropstate":<String_value>

}]}
```

count

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/clusterinstance?**count=yes**

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "clusterinstance": [ { "__count": "#no"} ] }
```

clusterinstance_binding

Binding object which returns the resources bound to clusterinstance.

Properties

(click to see Operations)

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
clid	<Double>	Read-write	Unique number that identifies the cluster. Minimum value = 1 Maximum value = 16
clusterinstance_clusternode_binding	<clusterinstance_clusternode_binding[]>	Read-only	clusternode that can be bound to clusterinstance.

Operations

(click to see Properties)

GET| GET (ALL)

Some options that you can use for each operations:

- **Getting warnings in response:**NITRO allows you to get warnings in an operation by specifying the "warning" query parameter as "yes". For example, to get warnings while connecting to the Citrix ADC appliance, the URL is as follows:

  http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/login?warning=yes

  If any, the warnings are displayed in the response payload with the HTTP code "209 X-NITRO-WARNING".

- **Authenticated access for individual NITRO operations:**NITRO allows you to logon to the Citrix ADC appliance to perform individual operations. You can use this option instead of creating a NITRO session (using the login object) and then using that session to perform all operations,

  To do this, you must specify the username and password in the request header of the NITRO request as follows:

  X-NITRO-USER:<username>

  X-NITRO-PASS:<password>

  Note: In such cases, make sure that the request header DOES not include the following:

  Cookie:NITRO_AUTH_TOKEN=<tokenvalue>

`*Note:`*

Mandatory parameters are marked in redand placeholder content is marked in <green>.

get

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/clusterinstance_binding/clid_value<Double>

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "clusterinstance_binding": [ {
```

```
"clid":<Double_value>,

"clusterinstance_clusternode_binding":<clusterinstance_clusternode_binding
[]_value>

}]}
```

get (all)

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/clusterinstance_binding

Query-parameters:

bulkbindings

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/clusterinstance_binding?**bulkbindings=yes**

NITRO allows you to fetch bindings in bulk.

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "clusterinstance_binding": [ {

"clid":<Double_value>,

"clusterinstance_clusternode_binding":<clusterinstance_clusternode_binding
[]_value>
```

```
}]}
```

clusterinstance_clusternode_binding

Binding object showing the clusternode that can be bound to clusterinstance.

Properties

(click to see Operations)

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
nodeid	<Double>	Read-write	The unique number that identiies a cluster. Minimum value = 0 Maximum value = 31
clid	<Double>	Read-write	Unique number that identifies the cluster. Minimum value = 1 Maximum value = 16
state	<String>	Read-only	Active, Spare or Passive. An active node serves traffic. A spare node serves as a backup for active nodes. A passive node does not serve traffic. This may be useful during temporary maintenance activity where it is desirable that the node takes part in the

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
			consensus protocol, but not serve traffic. Default value: PASSIVE Possible values = ACTIVE, SPARE, PASSIVE
nodersskeymismatch	<Boolean>	Read-only	This argument is used to determine if there is a RSS key mismatch at cluster node level.
nodelicensemismatch	<Boolean>	Read-only	This argument is used to determine if there is a License mismatch at cluster node level.
effectivestate	<String>	Read-only	Node effective health state. Possible values = UP, NOT UP, UNKNOWN, INIT
islocalnode	<Boolean>	Read-only	This argument is used to determine whether it is local node.
isconfigurationcoordinator	<Boolean>	Read-only	This argument is used to determine whether the node is configuration coordinator (CCO).
nodejumbonotsupported	<Boolean>	Read-only	This argument is used to determine if Jumbo framework

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
			not supported at cluster node level.
health	<String>	Read-only	Node Health state. Possible values = UNKNOWN, INIT, DOWN, UP, Some enabled and HAMON interfaces of the node are down, All interfaces of the node are down or disabled, SSL card(s) of the node have failed, Route Monitor(s) of the node have failed, Service state is being synchronized with the cluster, The backplane interface is either not set or it is down or is disabled, The CLAG member(s) of the node are down, Persistence sessions are being synchronized with the cluster, The Syn Cookie is being synchronized with the cluster, Unknown Health, AAA keys are being

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
			sychronized with the cluster, Cluster health is not up due to config sync is in progress
clusterhealth	<String>	Read-only	Node clusterd state. Possible values = UP, Configurations of the node are lagging behind the cluster by more than 256 commands, The config sync operation has exceeded the time limit of 60 seconds, The config sync operation (full sync) is in progress, The node is not in sync with the cluster configurations as sync is disabled on this node, The execution of a configuration command has failed on this node, UNKNOWN
masterstate	<String>	Read-only	Master state. Possible values = INACTIVE, ACTIVE, UNKNOWN

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
ipaddress	<String>	Read-only	The IP Address of the node.
__count	<Double>	Read-only	count parameter

Operations

(click to see Properties)

GET| GET (ALL)| COUNT

Some options that you can use for each operations:

- **Getting warnings in response:**NITRO allows you to get warnings in an operation by specifying the "warning" query parameter as "yes". For example, to get warnings while connecting to the Citrix ADC appliance, the URL is as follows:

  http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/login?warning=yes

  If any, the warnings are displayed in the response payload with the HTTP code "209 X-NITRO-WARNING".

- **Authenticated access for individual NITRO operations:**NITRO allows you to logon to the Citrix ADC appliance to perform individual operations. You can use this option instead of creating a NITRO session (using the login object) and then using that session to perform all operations,

  To do this, you must specify the username and password in the request header of the NITRO request as follows:

  X-NITRO-USER:<username>

  X-NITRO-PASS:<password>

  Note: In such cases, make sure that the request header DOES not include the following:

  Cookie:NITRO_AUTH_TOKEN=<tokenvalue>

`*Note:`*

Mandatory parameters are marked in redand placeholder content is marked in <green>.

get

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/clusterinstance_clusternode_binding/clid_value<Double>

Query-parameters:

filter

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/clusterinstance_clusternode_binding/clid_value<Double>?**filter=property-name1:property-value1,property-name2:property-value2**

Use this query-parameter to get the filtered set of clusterinstance_clusternode_binding resources configured on Citrix ADC. Filtering can be done on any of the properties of the resource.

pagination

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/clusterinstance_clusternode_binding/clid_value<Double>?**pagesize=#no;pageno=#no**

Use this query-parameter to get the clusterinstance_clusternode_binding resources in chunks.

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "clusterinstance_clusternode_binding": [ {

"nodeid":<Double_value>,

"clid":<Double_value>,
```

```
"state":<String_value>,

"nodersskeymismatch":<Boolean_value>,

"nodelicensemismatch":<Boolean_value>,

"effectivestate":<String_value>,

"islocalnode":<Boolean_value>,

"isconfigurationcoordinator":<Boolean_value>,

"nodejumbonotsupported":<Boolean_value>,

"health":<String_value>,

"clusterhealth":<String_value>,

"masterstate":<String_value>,

"ipaddress":<String_value>

}]}
```

get (all)

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/clusterinstance_clusternode_binding

Query-parameters:

bulkbindings

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/clusterinstance_clusternode_binding?**bulkbindings=yes**

NITRO allows you to fetch bindings in bulk.

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "clusterinstance_clusternode_binding": [ {

"nodeid":<Double_value>,

"clid":<Double_value>,

"state":<String_value>,

"nodersskeymismatch":<Boolean_value>,

"nodelicensemismatch":<Boolean_value>,

"effectivestate":<String_value>,

"islocalnode":<Boolean_value>,

"isconfigurationcoordinator":<Boolean_value>,

"nodejumbonotsupported":<Boolean_value>,

"health":<String_value>,

"clusterhealth":<String_value>,

"masterstate":<String_value>,

"ipaddress":<String_value>

}]}
```

count

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/clusterinstance_clusternode_binding/clid_value<Double>?**count=yes**

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{"clusterinstance_clusternode_binding": [ { "__count": "#no"} ] }
```

clusternode

Configuration for cluster node resource.

Properties

(click to see Operations)

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
nodeid	<Double>	Read-write	Unique number that identifies the cluster node. Minimum value = 0 Maximum value = 31

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
ipaddress	<String>	Read-write	Citrix ADC IP (NSIP) address of the appliance to add to the cluster. Must be an IPv4 address. Minimum length = 1
state	<String>	Read-write	Admin state of the cluster node. The available settings function as follows: ACTIVE - The node serves traffic. SPARE - The node does not serve traffic unless an ACTIVE node goes down. PASSIVE - The node does not serve traffic, unless you change its state. PASSIVE state is useful during temporary maintenance activities in which you want the node to take part in the consensus protocol but not to serve traffic. Default value: PASSIVE

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
			Possible values = ACTIVE, SPARE, PASSIVE
backplane	<String>	Read-write	Interface through which the node communicates with the other nodes in the cluster. Must be specified in the three-tuple form n/c/u, where n represents the node ID and c/u refers to the interface on the appliance. Minimum length = 1
priority	<Double>	Read-write	Preference for selecting a node as the configuration coordinator. The node with the lowest priority value is selected as the configuration coordinator. When the current configuration coordinator goes down, the node with the next lowest priority is made the new configuration coordinator. When the original node comes back up, it will preempt the new configuration

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
			coordinator and take over as the configuration coordinator. Note: When priority is not configured for any of the nodes or if multiple nodes have the same priority, the cluster elects one of the nodes as the configuration coordinator. Default value: 31 Minimum value = 0 Maximum value = 31
nodegroup	<String>	Read-write	The default node group in a Cluster system. Default value: DEFAULT_NG Minimum length = 1
delay	<Double>	Read-write	Applicable for Passive node and node becomes passive after this timeout (in minutes). Default value: 0 Minimum value = 0

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
			Maximum value = 1440
tunnelmode	<String>	Read-write	To set the tunnel mode. Default value: NONE Possible values = NONE, GRE, UDP
clearnodegroupconfig	<String>	Read-write	Option to remove nodegroup config. Default value: YES Possible values = YES, NO
clusterhealth	<String>	Read-only	Node clusterd state. Possible values = UP, Configurations of the node are lagging behind the cluster by more than 256 commands, The config sync operation has exceeded the time limit of 60 seconds, The config sync operation (full sync) is in progress, The node is not in sync with the cluster configurations as sync is disabled on this node, The

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
			execution of a configuration command has failed on this node, UNKNOWN
effectivestate	<String>	Read-only	Node effective health state. Possible values = UP, NOT UP, UNKNOWN, INIT
operationalsyncstate	<String>	Read-only	Node Operational Reconciliation state. Default value: ENABLED Possible values = ENABLED, SUCCESS, IN PROGRESS, FAILED, INCREMENTAL SYNC DISABLED, DISABLED, UNKNOWN
masterstate	<String>	Read-only	Node Master state. Possible values = INACTIVE, ACTIVE, UNKNOWN
health	<String>	Read-only	Node Health state. Possible values = UNKNOWN, INIT, DOWN, UP, Some enabled and HAMON interfaces

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
			of the node are down, All interfaces of the node are down or disabled, SSL card(s) of the node have failed, Route Monitor(s) of the node have failed, Service state is being synchronized with the cluster, The backplane interface is either not set or it is down or is disabled, The CLAG member(s) of the node are down, Persistence sessions are being synchronized with the cluster, The Syn Cookie is being synchronized with the cluster, Unknown Health, AAA keys are being sychronized with the cluster, Cluster health is not up due to config sync is in progress
syncstate	<String>	Read-only	Enable/Disable the synchronization of

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
			cluster configurations on the node. Default value: ENABLED Possible values = ENABLED, DISABLED
isconfigurationcoordinator	<Boolean>	Read-only	This argument is used to determine whether the node is configuration coordinator (CCO).
islocalnode	<Boolean>	Read-only	This argument is used to determine whether it is local node.
nodersskeymismatch	<Boolean>	Read-only	This argument is used to determine if there is a RSS key mismatch at cluster node level.
nodelicensemismatch	<Boolean>	Read-only	This argument is used to determine if there is a License mismatch at cluster node level.
nodejumbonotsupported	<Boolean>	Read-only	This argument is used to determine if Jumbo framework not supported at cluster node level.
nodelist	<Double[]>	Read-only	Nodelist for displaying Heartbeat

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
			not seen interfaces on a cluster node.
ifaceslist	<String[]>	Read-only	Interface list corresponding to nodelist for Heartbeat not seen interfaces on a cluster node.
enabledifaces	<String>	Read-only	Enabled Interfaces on a cluster node.
disabledifaces	<String>	Read-only	Disabled Interfaces on a cluster node.
partialfailifaces	<String>	Read-only	Partial Failure Interfaces on a cluster node.
hamonifaces	<String>	Read-only	Hamon Interfaces on a cluster node.
name	<String>	Read-only	Name of the state specific nodegroup. Minimum length = 1
cfgflags	<Double>	Read-only	Flag indicates whether the node is bound to cluster nodegroup.
routemonitor	<String>	Read-only	The IP address (IPv4 or IPv6).
netmask	<String>	Read-only	The netmask.
__count	<Double>	Read-only	count parameter

Operations

(click to see Properties)

ADD| UPDATE| UNSET| DELETE| GET (ALL)| GET| COUNT

Some options that you can use for each operations:

- **Getting warnings in response:**NITRO allows you to get warnings in an operation by specifying the "warning" query parameter as "yes". For example, to get warnings while connecting to the Citrix ADC appliance, the URL is as follows:

  http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/login?warning=yes

  If any, the warnings are displayed in the response payload with the HTTP code "209 X-NITRO-WARNING".

- **Authenticated access for individual NITRO operations:**NITRO allows you to logon to the Citrix ADC appliance to perform individual operations. You can use this option instead of creating a NITRO session (using the login object) and then using that session to perform all operations,

  To do this, you must specify the username and password in the request header of the NITRO request as follows:

  X-NITRO-USER:<username>

  X-NITRO-PASS:<password>

  Note: In such cases, make sure that the request header DOES not include the following:

  Cookie:NITRO_AUTH_TOKEN=<tokenvalue>

`*Note:`*

Mandatory parameters are marked in redand placeholder content is marked in <green>.

add

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/clusternode

HTTP Method:POST

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Content-Type:application/json

Request Payload:

```
{"clusternode":{

<b>"nodeid":<Double_value>,

</b><b>"ipaddress":<String_value>,

</b>"state":<String_value>,

"backplane":<String_value>,

"priority":<Double_value>,

"nodegroup":<String_value>,

"delay":<Double_value>,

"tunnelmode":<String_value>

}}
```

Response:

HTTP Status Code on Success: 201 Created HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error

update

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/clusternode

HTTP Method:PUT

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Content-Type:application/json

Request Payload:

```
{"clusternode":{

<b>"nodeid":<Double_value>,

</b>"state":<String_value>,

"backplane":<String_value>,

"priority":<Double_value>,

"delay":<Double_value>,

"tunnelmode":<String_value>

}}
```

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error

unset

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/clusternode?**action=unset**

HTTP Method:POST

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Content-Type:application/json

Request Payload:

```
{"clusternode":{

<b>"nodeid":<Double_value>,
```

```
</b>"state":true,

"backplane":true,

"priority":true,

"delay":true,

"tunnelmode":true

}}
```

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error

delete

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/clusternode/nodeid_value<Double>

HTTP Method:DELETE

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue>

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error

get (all)

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/clusternode

Query-parameters:

attrs

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/clusternode?**attrs=property-name1,property-name2**

Use this query parameter to specify the resource details that you want to retrieve.

filter

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/clusternode?**filter=property-name1:property-val1,property-name2:property-val2**

Use this query-parameter to get the filtered set of clusternode resources configured on Citrix ADC. Filtering can be done on any of the properties of the resource.

view

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/clusternode?**view=summary**

Note: By default, the retrieved results are displayed in detail view (?view=detail).

pagination

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/clusternode?**pagesize=#no;pageno=#no**

Use this query-parameter to get the clusternode resources in chunks.

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "clusternode": [ {
```

```
"nodeid":<Double_value>,

"ipaddress":<String_value>,

"clusterhealth":<String_value>,

"effectivestate":<String_value>,

"operationalsyncstate":<String_value>,

"masterstate":<String_value>,

"health":<String_value>,

"state":<String_value>,

"backplane":<String_value>,

"syncstate":<String_value>,

"priority":<Double_value>,

"isconfigurationcoordinator":<Boolean_value>,

"islocalnode":<Boolean_value>,

"nodersskeymismatch":<Boolean_value>,

"nodelicensemismatch":<Boolean_value>,

"nodejumbonotsupported":<Boolean_value>,

"nodelist":<Double[]_value>,

"ifaceslist":<String[]_value>,

"enabledifaces":<String_value>,

"disabledifaces":<String_value>,

"partialfailifaces":<String_value>,

"hamonifaces":<String_value>,
```

```
"nodegroup":<String_value>,

"name":<String_value>,

"cfgflags":<Double_value>,

"routemonitor":<String_value>,

"netmask":<String_value>,

"delay":<Double_value>,

"tunnelmode":<String_value>

}]}
```

get

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/clusternode/nodeid_value<Double>

Query-parameters:

attrs

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/clusternode/nodeid_value<Double>?**attrs=property-name1,property-name2**

Use this query parameter to specify the resource details that you want to retrieve.

view

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/clusternode/nodeid_value<Double>?**view=summary**

Note: By default, the retrieved results are displayed in detail view (?view=detail).

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "clusternode": [ {

"nodeid":<Double_value>,

"ipaddress":<String_value>,

"clusterhealth":<String_value>,

"effectivestate":<String_value>,

"operationalsyncstate":<String_value>,

"masterstate":<String_value>,

"health":<String_value>,

"state":<String_value>,

"backplane":<String_value>,

"syncstate":<String_value>,

"priority":<Double_value>,

"isconfigurationcoordinator":<Boolean_value>,

"islocalnode":<Boolean_value>,

"nodersskeymismatch":<Boolean_value>,

"nodelicensemismatch":<Boolean_value>,

"nodejumbonotsupported":<Boolean_value>,
```

```
"nodelist":<Double[]_value>,

"ifaceslist":<String[]_value>,

"enabledifaces":<String_value>,

"disabledifaces":<String_value>,

"partialfailifaces":<String_value>,

"hamonifaces":<String_value>,

"nodegroup":<String_value>,

"name":<String_value>,

"cfgflags":<Double_value>,

"routemonitor":<String_value>,

"netmask":<String_value>,

"delay":<Double_value>,

"tunnelmode":<String_value>

}]}
```

count

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/clusternode?**count=yes**

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "clusternode": [ { "__count": "#no"} ] }
```

clusternodegroup

Configuration for Node group object type resource.

Properties

(click to see Operations)

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
name	<String>	Read-write	Name of the nodegroup. The name uniquely identifies the nodegroup on the cluster. Minimum length = 1
strict	<String>	Read-write	Specifies whether cluster nodes, that are not part of the nodegroup, will be used as backup for the nodegroup. * Enabled - When one of the nodes goes down, no other cluster node is picked up to replace

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
			it. When the node comes up, it will continue being part of the nodegroup. * Disabled - When one of the nodes goes down, a non-nodegroup cluster node is picked up and acts as part of the nodegroup. When the original node of the nodegroup comes up, the backup node will be replaced. Default value: NO Possible values = YES, NO
sticky	<String>	Read-write	Only one node can be bound to nodegroup with this option enabled. It specifies whether to prempt the traffic for the entities bound to nodegroup when owner node goes down and rejoins the cluster. * Enabled - When owner node goes down, backup node will become the

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
			owner node and takes the traffic for the entities bound to the nodegroup. When bound node rejoins the cluster, traffic for the entities bound to nodegroup will not be steered back to this bound node. Current owner will have the ownership till it goes down. * Disabled - When one of the nodes goes down, a non-nodegroup cluster node is picked up and acts as part of the nodegroup. When the original node of the nodegroup comes up, the backup node will be replaced. Default value: NO Possible values = YES, NO
state	<String>	Read-write	State of the nodegroup. All the nodes binding to this nodegroup must

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
			have the same state. ACTIVE/SPARE/ PASSIVE. Possible values = ACTIVE, SPARE, PASSIVE
priority	<Double>	Read-write	Priority of Nodegroup. This priority is used for all the nodes bound to the nodegroup for Nodegroup selection. Minimum value = 0 Maximum value = 31
currentnodemask	<Double>	Read-only	Bitmap of current nodes in this nodegroup.
backupnodemask	<Double>	Read-only	Bitmap of backup nodes in this nodegroup.
boundedentitiescntfrompe	<Double>	Read-only	Count of bounded entities to this nodegroup accoding to PE.
activelist	<Double[]>	Read-only	Active node list of this nodegroup.
backuplist	<Double[]>	Read-only	Backup node list of this nodegroup.
__count	<Double>	Read-only	count parameter

Operations

(click to see Properties)

ADD| UPDATE| UNSET| DELETE| GET (ALL)| GET| COUNT

Some options that you can use for each operations:

- **Getting warnings in response:**NITRO allows you to get warnings in an operation by specifying the "warning" query parameter as "yes". For example, to get warnings while connecting to the Citrix ADC appliance, the URL is as follows:

  http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/login?warning=yes

  If any, the warnings are displayed in the response payload with the HTTP code "209 X-NITRO-WARNING".

- **Authenticated access for individual NITRO operations:**NITRO allows you to logon to the Citrix ADC appliance to perform individual operations. You can use this option instead of creating a NITRO session (using the login object) and then using that session to perform all operations,

  To do this, you must specify the username and password in the request header of the NITRO request as follows:

  X-NITRO-USER:<username>

  X-NITRO-PASS:<password>

  Note: In such cases, make sure that the request header DOES not include the following:

  Cookie:NITRO_AUTH_TOKEN=<tokenvalue>

`*Note:`*

Mandatory parameters are marked in redand placeholder content is marked in <green>.

add

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/clusternodegroup

HTTP Method:POST

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Content-Type:application/json

Request Payload:

```
{"clusternodegroup":{

<b>"name":<String_value>,

</b>"strict":<String_value>,

"sticky":<String_value>,

"state":<String_value>,

"priority":<Double_value>

}}
```

Response:

HTTP Status Code on Success: 201 Created HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error

update

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/clusternodegroup

HTTP Method:PUT

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Content-Type:application/json

Request Payload:

```
{"clusternodegroup":{

<b>"name":<String_value>,

</b>"strict":<String_value>,
```

```
"state":<String_value>,

"priority":<Double_value>

}}
```

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error

unset

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/clusternodegroup?**action=unset**

HTTP Method:POST

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Content-Type:application/json

Request Payload:

```
{"clusternodegroup":{

<b>"name":<String_value>,

</b>"strict":true

}}
```

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error

delete

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/clusternodegroup/name_value<String>

HTTP Method:DELETE

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue>

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error

get (all)

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/clusternodegroup

Query-parameters:

attrs

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/clusternodegroup?**attrs=property-name1,property-name2**

Use this query parameter to specify the resource details that you want to retrieve.

filter

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/clusternodegroup?**filter=property-name1:property-val1,property-name2:property-val2**

Use this query-parameter to get the filtered set of clusternodegroup resources configured on Citrix ADC. Filtering can be done on any of the properties of the resource.

view

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/clusternodegroup?**view=summary**

Note: By default, the retrieved results are displayed in detail view (?view=detail).

pagination

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/clusternodegroup?**pagesize=#no;pageno=#no**

Use this query-parameter to get the clusternodegroup resources in chunks.

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "clusternodegroup": [ {

"name":<String_value>,

"strict":<String_value>,

"sticky":<String_value>,

"currentnodemask":<Double_value>,

"backupnodemask":<Double_value>,

"boundedentitiescntfrompe":<Double_value>,

"activelist":<Double[]_value>,

"backuplist":<Double[]_value>,

"state":<String_value>,

"priority":<Double_value>

}]}
```

get

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/clusternodegroup/name_value<String>

Query-parameters:

attrs

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/clusternodegroup/name_value<String>?**attrs=property-name1,property-name2**

Use this query parameter to specify the resource details that you want to retrieve.

view

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/clusternodegroup/name_value<String>?**view=summary**

Note: By default, the retrieved results are displayed in detail view (?view=detail).

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "clusternodegroup": [ {

"name":<String_value>,

"strict":<String_value>,
```

```
"sticky":<String_value>,

"currentnodemask":<Double_value>,

"backupnodemask":<Double_value>,

"boundedentitiescntfrompe":<Double_value>,

"activelist":<Double[]_value>,

"backuplist":<Double[]_value>,

"state":<String_value>,

"priority":<Double_value>

}]}
```

count

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/clusternodegroup?**count=yes**

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "clusternodegroup": [ { "__count": "#no"} ] }
```

clusternodegroup_authenticationvserver_binding

Binding object showing the authenticationvserver that can be bound to clusternodegroup.

Properties

(click to see Operations)

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
vserver	<String>	Read-write	vserver that need to be bound to this nodegroup.
name	<String>	Read-write	Name of the nodegroup. The name uniquely identifies the nodegroup on the cluster. Minimum length = 1
__count	<Double>	Read-write	count parameter

Operations

(click to see Properties)

ADD:| DELETE:| GET| GET (ALL)| COUNT

Some options that you can use for each operations:

- **Getting warnings in response:**NITRO allows you to get warnings in an operation by specifying the "warning" query parameter as "yes". For example, to get warnings while connecting to the Citrix ADC appliance, the URL is as follows:

  http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/login?warning=yes

  If any, the warnings are displayed in the response payload with the HTTP code "209 X-NITRO-WARNING".

- **Authenticated access for individual NITRO operations:**NITRO allows you to logon to the Citrix ADC appliance to perform individual operations. You can use this option instead of creating a NITRO session (using the login object) and then using that session to perform all operations,

  To do this, you must specify the username and password in the request header of the NITRO request as follows:

  X-NITRO-USER:<username>

  X-NITRO-PASS:<password>

  Note: In such cases, make sure that the request header DOES not include the following:

  Cookie:NITRO_AUTH_TOKEN=<tokenvalue>

`*Note:`*

Mandatory parameters are marked in redand placeholder content is marked in <green>.

add

URL:http://<Citrix-ADC-IP-address(NSIP)/nitro/v1/config/clusternodegroup_authenticationvserver_binding

HTTP Method:PUT

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Content-Type:application/json

Request Payload:

```
{

"clusternodegroup_authenticationvserver_binding":{

<b>"name":<String_value>,

</b>"vserver":<String_value>

}}
```

Response:

HTTP Status Code on Success: 201 Created HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error

delete

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/clusternodegroup_authenticationvserver_binding

Query-parameters:

args

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/clusternodegroup_authenticationvserver_binding?**args=name:<String_value>,vserver:<String_value>**

HTTP Method:DELETE

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue>

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error

get

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/clusternodegroup_authenticationvserver_binding

Query-parameters:

filter

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/clusternodegroup_authenticationvserver_binding?**filter=property-name1:property-value1,property-name2:property-value2**

Use this query-parameter to get the filtered set of clusternodegroup_authenticationvserver_binding resources configured on Citrix ADC. Filtering can be done on any of the properties of the resource.

pagination

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/clusternodegroup_authenticationvserver_binding?**pagesize=#no;pageno=#no**

Use this query-parameter to get the clusternodegroup_authenticationvserver_binding resources in chunks.

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "clusternodegroup_authenticationvserver_binding": [ {

"vserver":<String_value>,

"name":<String_value>

}]}
```

get (all)

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/clusternodegroup_authenticationvserver_binding

Query-parameters:

bulkbindings

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/clusternodegroup_authenticationvserver_binding?**bulkbindings=yes**

NITRO allows you to fetch bindings in bulk.

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "clusternodegroup_authenticationvserver_binding": [ {

"vserver":<String_value>,

"name":<String_value>

}]}
```

count

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/clusternodegroup_authenticationvserver_binding?**args=;count=yes**

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "clusternodegroup_authenticationvserver_binding": [ { "__count":
"#no"} ] }
```

clusternodegroup_binding

Binding object which returns the resources bound to clusternodegroup.

Properties

(click to see Operations)

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
name	<String>	Read-write	Name of the nodegroup to be displayed. If a name is not provided, information about all nodegroups is displayed. Minimum length = 1
clusternodegroup_csvserver_binding	<clusternodegroup_csvserver_binding[]>	Read-only	csvserver that can be bound to clusternodegroup.
clusternodegroup_gslbsite_binding	<clusternodegroup_gslbsite_binding[]>	Read-only	gslbsite that can be bound to clusternodegroup.

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
clusternodegroup_nslimitidentifier_binding	<clusternodegroup_nslimitidentifier_binding[]>	Read-only	nslimitidentifier that can be bound to clusternodegroup.
clusternodegroup_clusternode_binding	<clusternodegroup_clusternode_binding[]>	Read-only	clusternode that can be bound to clusternodegroup.
clusternodegroup_lbvserver_binding	<clusternodegroup_lbvserver_binding[]>	Read-only	lbvserver that can be bound to clusternodegroup.
clusternodegroup_vpnvserver_binding	<clusternodegroup_vpnvserver_binding[]>	Read-only	vpnvserver that can be bound to clusternodegroup.
clusternodegroup_service_binding	<clusternodegroup_service_binding[]>	Read-only	service that can be bound to clusternodegroup.
clusternodegroup_streamidentifier_binding	<clusternodegroup_streamidentifier_binding[]>	Read-only	streamidentifier that can be bound to clusternodegroup.
clusternodegroup_gslbvserver_binding	<clusternodegroup_gslbvserver_binding[]>	Read-only	gslbvserver that can be bound to clusternodegroup.
clusternodegroup_authenticationvserver_binding	<clusternodegroup_authenticationvserver_binding[]>	Read-only	authenticationvserver that can be bound to clusternodegroup.
clusternodegroup_crvserver_binding	<clusternodegroup_crvserver_binding[]>	Read-only	crvserver that can be bound to clusternodegroup.

Operations

(click to see Properties)

GET| GET (ALL)

Some options that you can use for each operations:

- **Getting warnings in response:**NITRO allows you to get warnings in an operation by specifying the "warning" query parameter as "yes". For example, to get warnings while connecting to the Citrix ADC appliance, the URL is as follows:

  http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/login?warning=yes

  If any, the warnings are displayed in the response payload with the HTTP code "209 X-NITRO-WARNING".

- **Authenticated access for individual NITRO operations:**NITRO allows you to logon to the Citrix ADC appliance to perform individual operations. You can use this option instead of creating a NITRO session (using the login object) and then using that session to perform all operations,

  To do this, you must specify the username and password in the request header of the NITRO request as follows:

  X-NITRO-USER:<username>

  X-NITRO-PASS:<password>

  Note: In such cases, make sure that the request header DOES not include the following:

  Cookie:NITRO_AUTH_TOKEN=<tokenvalue>

`*Note:`*

Mandatory parameters are marked in redand placeholder content is marked in <green>.

get

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/clusternodegroup_binding/name_value<String>

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "clusternodegroup_binding": [ {

"name":<String_value>,

"clusternodegroup_csvserver_binding":<clusternodegroup_csvserver_binding[]
_value>,

"clusternodegroup_vpnvserver_binding":<clusternodegroup_vpnvserver_binding
[]_value>,

"clusternodegroup_crvserver_binding":<clusternodegroup_crvserver_binding[]
_value>,

"clusternodegroup_nslimitidentifier_binding":<clusternodegroup_nslimitiden
tifier_binding[]_value>,

"clusternodegroup_gslbsite_binding":<clusternodegroup_gslbsite_binding[]_v
alue>,

"clusternodegroup_clusternode_binding":<clusternodegroup_clusternode_bindi
ng[]_value>,

"clusternodegroup_authenticationvserver_binding":<clusternodegroup_authent
icationvserver_binding[]_value>,

"clusternodegroup_lbvserver_binding":<clusternodegroup_lbvserver_binding[]
_value>,

"clusternodegroup_gslbvserver_binding":<clusternodegroup_gslbvserver_bindi
ng[]_value>,

"clusternodegroup_streamidentifier_binding":<clusternodegroup_streamidenti
fier_binding[]_value>,
```

```
"clusternodegroup_service_binding":<clusternodegroup_service_binding[]_val
ue>

}]}
```

get (all)

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/clusternodegroup_binding

Query-parameters:

bulkbindings

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/clusternodegroup_binding?**bulkbindings=yes**

NITRO allows you to fetch bindings in bulk.

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "clusternodegroup_binding": [ {

"name":<String_value>,

"clusternodegroup_csvserver_binding":<clusternodegroup_csvserver_binding[]
_value>,
```

```
"clusternodegroup_vpnvserver_binding":<clusternodegroup_vpnvserver_binding
[]_value>,

"clusternodegroup_crvserver_binding":<clusternodegroup_crvserver_binding[]
_value>,

"clusternodegroup_nslimitidentifier_binding":<clusternodegroup_nslimitiden
tifier_binding[]_value>,

"clusternodegroup_gslbsite_binding":<clusternodegroup_gslbsite_binding[]_v
alue>,

"clusternodegroup_clusternode_binding":<clusternodegroup_clusternode_bindi
ng[]_value>,

"clusternodegroup_authenticationvserver_binding":<clusternodegroup_authent
icationvserver_binding[]_value>,

"clusternodegroup_lbvserver_binding":<clusternodegroup_lbvserver_binding[]
_value>,

"clusternodegroup_gslbvserver_binding":<clusternodegroup_gslbvserver_bindi
ng[]_value>,

"clusternodegroup_streamidentifier_binding":<clusternodegroup_streamidenti
fier_binding[]_value>,

"clusternodegroup_service_binding":<clusternodegroup_service_binding[]_val
ue>

}]}
```

clusternodegroup_clusternode_binding

Binding object showing the clusternode that can be bound to clusternodegroup.

Properties

(click to see Operations)

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
name	<String>	Read-write	Name of the nodegroup. The name uniquely identifies the nodegroup on the cluster. Minimum length = 1
node	<Double>	Read-write	Nodes in the nodegroup. Minimum value = 0 Maximum value = 31
__count	<Double>	Read-write	count parameter

Operations

(click to see Properties)

ADD:| DELETE:| GET| GET (ALL)| COUNT

Some options that you can use for each operations:

- **Getting warnings in response:**NITRO allows you to get warnings in an operation by specifying the "warning" query parameter as "yes". For example, to get warnings while connecting to the Citrix ADC appliance, the URL is as follows:

  http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/login?warning=yes

  If any, the warnings are displayed in the response payload with the HTTP code "209 X-NITRO-WARNING".

- **Authenticated access for individual NITRO operations:**NITRO allows you to logon to the Citrix ADC appliance to perform individual operations. You can use this option instead of creating a NITRO session (using the login object) and then using that session to perform all operations,

  To do this, you must specify the username and password in the request header of the NITRO request as follows:

X-NITRO-USER:<username>

X-NITRO-PASS:<password>

Note: In such cases, make sure that the request header DOES not include the following:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue>

`*Note:`*

Mandatory parameters are marked in redand placeholder content is marked in <green>.

add

URL:http://<Citrix-ADC-IP-address(NSIP)/nitro/v1/config/clusternodegroup_clusternode_binding

HTTP Method:PUT

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Content-Type:application/json

Request Payload:

```
{

"clusternodegroup_clusternode_binding":{

<b>"name":<String_value>,

</b>"node":<Double_value>

}}
```

Response:

HTTP Status Code on Success: 201 Created HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error

delete

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/clusternodegroup_clusternode_binding

Query-parameters:

args

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/clusternodegroup_clusternode_binding?**args=name:<String_value>,node:<Double_value>**

HTTP Method:DELETE

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue>

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error

get

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/clusternodegroup_clusternode_binding

Query-parameters:

filter

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/clusternodegroup_clusternode_binding?**filter=property-name1:property-value1,property-name2:property-value2**

Use this query-parameter to get the filtered set of clusternodegroup_clusternode_binding resources configured on Citrix ADC. Filtering can be done on any of the properties of the resource.

pagination

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/clusternodegroup_clusternode_binding?**pagesize=#no;pageno=#no**

Use this query-parameter to get the clusternodegroup_clusternode_binding resources in chunks.

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "clusternodegroup_clusternode_binding": [ {

"name":<String_value>,

"node":<Double_value>

}]}
```

get (all)

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/clusternodegroup_clusternode_binding

Query-parameters:

bulkbindings

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/clusternodegroup_clusternode_binding?**bulkbindings=yes**

NITRO allows you to fetch bindings in bulk.

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "clusternodegroup_clusternode_binding": [ {

"name":<String_value>,

"node":<Double_value>

}]}
```

count

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/clusternodegroup_clusternode_binding?**args=;count=yes**

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "clusternodegroup_clusternode_binding": [ { "__count": "#no"} ] }
```

clusternodegroup_crvserver_binding

Binding object showing the crvserver that can be bound to clusternodegroup.

Properties

(click to see Operations)

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
vserver	<String>	Read-write	vserver that need to be bound to this nodegroup.
name	<String>	Read-write	Name of the nodegroup. The name uniquely identifies the nodegroup on the cluster. Minimum length = 1
__count	<Double>	Read-write	count parameter

Operations

(click to see Properties)

ADD:| DELETE:| GET| GET (ALL)| COUNT

Some options that you can use for each operations:

- **Getting warnings in response:**NITRO allows you to get warnings in an operation by specifying the "warning" query parameter as "yes". For example, to get warnings while connecting to the Citrix ADC appliance, the URL is as follows:

  http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/login?warning=yes

  If any, the warnings are displayed in the response payload with the HTTP code "209 X-NITRO-WARNING".

- **Authenticated access for individual NITRO operations:**NITRO allows you to logon to the Citrix ADC appliance to perform individual operations. You can use this option instead of creating a NITRO session (using the login object) and then using that session to perform all operations,

  To do this, you must specify the username and password in the request header of the NITRO request as follows:

  X-NITRO-USER:<username>

  X-NITRO-PASS:<password>

  Note: In such cases, make sure that the request header DOES not include the following:

  Cookie:NITRO_AUTH_TOKEN=<tokenvalue>

`*Note:`*

Mandatory parameters are marked in redand placeholder content is marked in <green>.

add

URL:http://<Citrix-ADC-IP-address(NSIP)/nitro/v1/config/clusternodegroup_crvserver_binding

HTTP Method:PUT

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Content-Type:application/json

Request Payload:

```
{

"clusternodegroup_crvserver_binding":{

<b>"name":<String_value>,

</b>"vserver":<String_value>

}}
```

Response:

HTTP Status Code on Success: 201 Created HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error

delete

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/clusternodegroup_crvserver_binding

Query-parameters:

args

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/clusternodegroup_crvserver_binding?**args=name:<String_value>,vserver:<String_value>**

HTTP Method:DELETE

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue>

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error

get

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/clusternodegroup_crvserver_binding

Query-parameters:

filter

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/clusternodegroup_crvserver_binding?**filter=property-name1:property-value1,property-name2:property-value2**

Use this query-parameter to get the filtered set of clusternodegroup_crvserver_binding resources configured on Citrix ADC. Filtering can be done on any of the properties of the resource.

pagination

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/clusternodegroup_crvserver_binding?**pagesize=#no;pageno=#no**

Use this query-parameter to get the clusternodegroup_crvserver_binding resources in chunks.

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "clusternodegroup_crvserver_binding": [ {

"vserver":<String_value>,

"name":<String_value>

}]}
```

get (all)

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/clusternodegroup_crvserver_binding

Query-parameters:

bulkbindings

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/clusternodegroup_crvserver_binding?**bulkbindings=yes**

NITRO allows you to fetch bindings in bulk.

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "clusternodegroup_crvserver_binding": [ {

"vserver":<String_value>,

"name":<String_value>

}]}
```

count

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/clusternodegroup_crvserver_binding?**args=;count=yes**

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "clusternodegroup_crvserver_binding": [ { "__count": "#no"} ] }
```

clusternodegroup_csvserver_binding

Binding object showing the csvserver that can be bound to clusternodegroup.

Properties

(click to see Operations)

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
vserver	<String>	Read-write	vserver that need to be bound to this nodegroup.
name	<String>	Read-write	Name of the nodegroup. The name uniquely identifies the nodegroup on the cluster. Minimum length = 1
__count	<Double>	Read-write	count parameter

Operations

(click to see Properties)

ADD:| DELETE:| GET| GET (ALL)| COUNT

Some options that you can use for each operations:

- **Getting warnings in response:**NITRO allows you to get warnings in an operation by specifying the "warning" query parameter as "yes". For example, to get warnings while connecting to the Citrix ADC appliance, the URL is as follows:

  http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/login?warning=yes

  If any, the warnings are displayed in the response payload with the HTTP code "209 X-NITRO-WARNING".

- **Authenticated access for individual NITRO operations:**NITRO allows you to logon to the Citrix ADC appliance to perform individual operations. You can use this option instead of creating a NITRO session (using the login object) and then using that session to perform all operations,

  To do this, you must specify the username and password in the request header of the NITRO request as follows:

  X-NITRO-USER:<username>

  X-NITRO-PASS:<password>

  Note: In such cases, make sure that the request header DOES not include the following:

  Cookie:NITRO_AUTH_TOKEN=<tokenvalue>

`*Note:`*

Mandatory parameters are marked in redand placeholder content is marked in <green>.

add

URL:http://<Citrix-ADC-IP-address(NSIP)/nitro/v1/config/clusternodegroup_csvserver_binding

HTTP Method:PUT

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Content-Type:application/json

Request Payload:

```
{
```

```
"clusternodegroup_csvserver_binding":{

<b>"name":<String_value>,

</b>"vserver":<String_value>

}}
```

Response:

HTTP Status Code on Success: 201 Created HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error

delete

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/clusternodegroup_csvserver_binding

Query-parameters:

args

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/clusternodegroup_csvserver_binding?**args=name:<String_value>,vserver:<String_value>**

HTTP Method:DELETE

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue>

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error

get

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/clusternodegroup_csvserver_binding

Query-parameters:

filter

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/clusternodegroup_csvserver_binding?**filter=property-name1:property-value1,property-name2:property-value2**

Use this query-parameter to get the filtered set of clusternodegroup_csvserver_binding resources configured on Citrix ADC. Filtering can be done on any of the properties of the resource.

pagination

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/clusternodegroup_csvserver_binding?**pagesize=#no;pageno=#no**

Use this query-parameter to get the clusternodegroup_csvserver_binding resources in chunks.

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "clusternodegroup_csvserver_binding": [ {

"vserver":<String_value>,

"name":<String_value>

}]}
```

get (all)

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/clusternodegroup_csvserver_binding

Query-parameters:

bulkbindings

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/clusternodegroup_csvserver_binding?**bulkbindings=yes**

NITRO allows you to fetch bindings in bulk.

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "clusternodegroup_csvserver_binding": [ {

"vserver":<String_value>,

"name":<String_value>

}]}
```

count

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/clusternodegroup_csvserver_binding?**args=;count=yes**

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "clusternodegroup_csvserver_binding": [ { "__count": "#no"} ] }
```

clusternodegroup_gslbsite_binding

Binding object showing the gslbsite that can be bound to clusternodegroup.

Properties

(click to see Operations)

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
gslbsite	<String>	Read-write	vserver that need to be bound to this nodegroup.
name	<String>	Read-write	Name of the nodegroup. The name uniquely identifies the nodegroup on the cluster. Minimum length = 1
__count	<Double>	Read-write	count parameter

Operations

(click to see Properties)

ADD:| DELETE:| GET| GET (ALL)| COUNT

Some options that you can use for each operations:

- **Getting warnings in response:**NITRO allows you to get warnings in an operation by specifying the "warning" query parameter as "yes". For example, to get warnings while connecting to the Citrix ADC appliance, the URL is as follows:

  http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/login?warning=yes

  If any, the warnings are displayed in the response payload with the HTTP code "209 X-NITRO-WARNING".

- **Authenticated access for individual NITRO operations:**NITRO allows you to logon to the Citrix ADC appliance to perform individual operations. You can use this option instead of creating a NITRO session (using the login object) and then using that session to perform all operations,

  To do this, you must specify the username and password in the request header of the NITRO request as follows:

  X-NITRO-USER:<username>

  X-NITRO-PASS:<password>

  Note: In such cases, make sure that the request header DOES not include the following:

  Cookie:NITRO_AUTH_TOKEN=<tokenvalue>

`*Note:`*

Mandatory parameters are marked in redand placeholder content is marked in <green>.

add

URL:http://<Citrix-ADC-IP-address(NSIP)/nitro/v1/config/clusternodegroup_gslbsite_binding

HTTP Method:PUT

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Content-Type:application/json

Request Payload:

```
{

"clusternodegroup_gslbsite_binding":{

<b>"name":<String_value>,

</b>"gslbsite":<String_value>

}}
```

Response:

HTTP Status Code on Success: 201 Created HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error

delete

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/clusternodegroup_gslbsite_binding

Query-parameters:

args

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/clusternodegroup_gslbsite_binding?**args=name:<String_value>,gslbsite:<String_value>**

HTTP Method:DELETE

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue>

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error

get

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/clusternodegroup_gslbsite_binding

Query-parameters:

filter

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/clusternodegroup_gslbsite_binding?**filter=property-name1:property-value1,property-name2:property-value2**

Use this query-parameter to get the filtered set of clusternodegroup_gslbsite_binding resources configured on Citrix ADC. Filtering can be done on any of the properties of the resource.

pagination

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/clusternodegroup_gslbsite_binding?**pagesize=#no;pageno=#no**

Use this query-parameter to get the clusternodegroup_gslbsite_binding resources in chunks.

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "clusternodegroup_gslbsite_binding": [ {

"gslbsite":<String_value>,

"name":<String_value>
```

```
}]}
```

get (all)

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/clusternodegroup_gslbsite_binding

Query-parameters:

bulkbindings

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/clusternodegroup_gslbsite_binding?**bulkbindings=yes**

NITRO allows you to fetch bindings in bulk.

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "clusternodegroup_gslbsite_binding": [ {

"gslbsite":<String_value>,

"name":<String_value>

}]}
```

count

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/clusternodegroup_gslbsite_binding?**args=;count=yes**

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "clusternodegroup_gslbsite_binding": [ { "__count": "#no"} ] }
```

clusternodegroup_gslbvserver_binding

Binding object showing the gslbvserver that can be bound to clusternodegroup.

Properties

(click to see Operations)

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
vserver	<String>	Read-write	vserver that need to be bound to this nodegroup.
name	<String>	Read-write	Name of the nodegroup. The name uniquely identifies the

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
			nodegroup on the cluster. Minimum length = 1
__count	<Double>	Read-write	count parameter

Operations

(click to see Properties)

ADD:| DELETE:| GET| GET (ALL)| COUNT

Some options that you can use for each operations:

- **Getting warnings in response:**NITRO allows you to get warnings in an operation by specifying the "warning" query parameter as "yes". For example, to get warnings while connecting to the Citrix ADC appliance, the URL is as follows:

  http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/login?warning=yes

  If any, the warnings are displayed in the response payload with the HTTP code "209 X-NITRO-WARNING".

- **Authenticated access for individual NITRO operations:**NITRO allows you to logon to the Citrix ADC appliance to perform individual operations. You can use this option instead of creating a NITRO session (using the login object) and then using that session to perform all operations,

  To do this, you must specify the username and password in the request header of the NITRO request as follows:

  X-NITRO-USER:<username>

  X-NITRO-PASS:<password>

  Note: In such cases, make sure that the request header DOES not include the following:

  Cookie:NITRO_AUTH_TOKEN=<tokenvalue>

`*Note:`*

Mandatory parameters are marked in redand placeholder content is marked in <green>.

add

URL:http://<Citrix-ADC-IP-address(NSIP)/nitro/v1/config/clusternodegroup_gslbvserver_binding

HTTP Method:PUT

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Content-Type:application/json

Request Payload:

```
{

"clusternodegroup_gslbvserver_binding":{

<b>"name":<String_value>,

</b>"vserver":<String_value>

}}
```

Response:

HTTP Status Code on Success: 201 Created HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error

delete

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/clusternodegroup_gslbvserver_binding

Query-parameters:

args

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/clusternodegroup_gslbvserver_binding?**args=name:<String_value>,vserver:<String_value>**

HTTP Method:DELETE

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue>

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error

get

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/clusternodegroup_gslbvserver_binding

Query-parameters:

filter

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/clusternodegroup_gslbvserver_binding?**filter=property-name1:property-value1,property-name2:property-value2**

Use this query-parameter to get the filtered set of clusternodegroup_gslbvserver_binding resources configured on Citrix ADC. Filtering can be done on any of the properties of the resource.

pagination

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/clusternodegroup_gslbvserver_binding?**pagesize=#no;pageno=#no**

Use this query-parameter to get the clusternodegroup_gslbvserver_binding resources in chunks.

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "clusternodegroup_gslbvserver_binding": [ {

"vserver":<String_value>,

"name":<String_value>

}]}
```

get (all)

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/clusternodegroup_gslbvserver_binding

Query-parameters:

bulkbindings

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/clusternodegroup_gslbvserver_binding?**bulkbindings=yes**

NITRO allows you to fetch bindings in bulk.

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "clusternodegroup_gslbvserver_binding": [ {
```

```
"vserver":<String_value>,

"name":<String_value>

}]}
```

count

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/clusternodegroup_gslbvserver_binding?**args=;count=yes**

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "clusternodegroup_gslbvserver_binding": [ { "__count": "#no"} ] }
```

clusternodegroup_lbvserver_binding

Binding object showing the lbvserver that can be bound to clusternodegroup.

Properties

(click to see Operations)

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
vserver	<String>	Read-write	vserver that need to be bound to this nodegroup.
name	<String>	Read-write	Name of the nodegroup. The name uniquely identifies the nodegroup on the cluster. Minimum length = 1
__count	<Double>	Read-write	count parameter

Operations

(click to see Properties)

ADD:| DELETE:| GET| GET (ALL)| COUNT

Some options that you can use for each operations:

- **Getting warnings in response:**NITRO allows you to get warnings in an operation by specifying the "warning" query parameter as "yes". For example, to get warnings while connecting to the Citrix ADC appliance, the URL is as follows:

  http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/login?warning=yes

  If any, the warnings are displayed in the response payload with the HTTP code "209 X-NITRO-WARNING".

- **Authenticated access for individual NITRO operations:**NITRO allows you to logon to the Citrix ADC appliance to perform individual operations. You can use this option instead of creating a NITRO session (using the login object) and then using that session to perform all operations,

  To do this, you must specify the username and password in the request header of the NITRO request as follows:

  X-NITRO-USER:<username>

  X-NITRO-PASS:<password>

Note: In such cases, make sure that the request header DOES not include the following:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue>

`*Note:` *

Mandatory parameters are marked in redand placeholder content is marked in <green>.

add

URL:http://<Citrix-ADC-IP-address(NSIP)/nitro/v1/config/clusternodegroup_lbvserver_binding

HTTP Method:PUT

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Content-Type:application/json

Request Payload:

```
{

"clusternodegroup_lbvserver_binding":{

<b>"name":<String_value>,

</b>"vserver":<String_value>

}}
```

Response:

HTTP Status Code on Success: 201 Created HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error

delete

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/clusternodegroup_lbvserver_binding

Query-parameters:

args

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/clusternodegroup_lbvserver_binding?**args=name:<String_value>,vserver:<String_value>**

HTTP Method:DELETE

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue>

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error

get

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/clusternodegroup_lbvserver_binding

Query-parameters:

filter

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/clusternodegroup_lbvserver_binding?**filter=property-name1:property-value1,property-name2:property-value2**

Use this query-parameter to get the filtered set of clusternodegroup_lbvserver_binding resources configured on Citrix ADC. Filtering can be done on any of the properties of the resource.

pagination

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/clusternodegroup_lbvserver_binding?**pagesize=#no;pageno=#no**

Use this query-parameter to get the clusternodegroup_lbvserver_binding resources in chunks.

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "clusternodegroup_lbvserver_binding": [ {

"vserver":<String_value>,

"name":<String_value>

}]}
```

get (all)

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/clusternodegroup_lbvserver_binding

Query-parameters:

bulkbindings

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/clusternodegroup_lbvserver_binding?**bulkbindings=yes**

NITRO allows you to fetch bindings in bulk.

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "clusternodegroup_lbvserver_binding": [ {

"vserver":<String_value>,

"name":<String_value>

}]}
```

count

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/clusternodegroup_lbvserver_binding?**args=;count=yes**

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "clusternodegroup_lbvserver_binding": [ { "__count": "#no"} ] }
```

clusternodegroup_nslimitidentifier_binding

Binding object showing the nslimitidentifier that can be bound to clusternodegroup.

Properties

(click to see Operations)

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
name	<String>	Read-write	Name of the nodegroup to which you want to bind a cluster node or an entity. Minimum length = 1
identifiername	<String>	Read-write	stream identifier and rate limit identifier that need to be bound to this nodegroup.
__count	<Double>	Read-write	count parameter

Operations

(click to see Properties)

ADD:| DELETE:| GET| GET (ALL)| COUNT

Some options that you can use for each operations:

- **Getting warnings in response:**NITRO allows you to get warnings in an operation by specifying the "warning" query parameter as "yes". For example, to get warnings while connecting to the Citrix ADC appliance, the URL is as follows:

  http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/login?warning=yes

  If any, the warnings are displayed in the response payload with the HTTP code "209 X-NITRO-WARNING".
- **Authenticated access for individual NITRO operations:**NITRO allows you to logon to the Citrix ADC appliance to perform individual operations. You can use this option instead of

creating a NITRO session (using the login object) and then using that session to perform all operations,

To do this, you must specify the username and password in the request header of the NITRO request as follows:

X-NITRO-USER:<username>

X-NITRO-PASS:<password>

Note: In such cases, make sure that the request header DOES not include the following:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue>

`*Note:`*

Mandatory parameters are marked in redand placeholder content is marked in <green>.

add

URL:http://<Citrix-ADC-IP-address(NSIP)/nitro/v1/config/clusternodegroup_nslimitidentifier_binding

HTTP Method:PUT

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Content-Type:application/json

Request Payload:

```
{

"clusternodegroup_nslimitidentifier_binding":{

<b>"name":<String_value>,

</b>"identifiername":<String_value>

}}
```

Response:

HTTP Status Code on Success: 201 Created HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error

delete

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/clusternodegroup_nslimitidentifier_binding/name_value<String>

Query-parameters:

args

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/clusternodegroup_nslimitidentifier_binding/name_value<String>?**args=identifiername:<String_value>**

HTTP Method:DELETE

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue>

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error

get

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/clusternodegroup_nslimitidentifier_binding/name_value<String>

Query-parameters:

filter

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/clusternodegroup_nslimitidentifier_binding/name_value<String>?**filter=property-name1:property-value1,property-name2:property-value2**

Use this query-parameter to get the filtered set of clusternodegroup_nslimitidentifier_binding resources configured on Citrix ADC. Filtering can be done on any of the properties of the resource.

pagination

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/clusternodegroup_nslimitidentifier_binding/name_value<String>?**pagesize=#no;pageno=#no**

Use this query-parameter to get the clusternodegroup_nslimitidentifier_binding resources in chunks.

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "clusternodegroup_nslimitidentifier_binding": [ {

"name":<String_value>,

"identifiername":<String_value>

}]}
```

get (all)

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/clusternodegroup_nslimitidentifier_binding

Query-parameters:

bulkbindings

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/clusternodegroup_nslimitidentifier_binding?**bulkbindings=yes**

NITRO allows you to fetch bindings in bulk.

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "clusternodegroup_nslimitidentifier_binding": [ {

"name":<String_value>,

"identifiername":<String_value>

}]}
```

count

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/clusternodegroup_nslimitidentifier_binding/name_value<String>?**count=yes**

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{"clusternodegroup_nslimitidentifier_binding": [ { "__count": "#no"} ] }
```

clusternodegroup_service_binding

Binding object showing the service that can be bound to clusternodegroup.

Properties

(click to see Operations)

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
name	<String>	Read-write	Name of the nodegroup. The name uniquely identifies the nodegroup on the cluster. Minimum length = 1
service	<String>	Read-write	name of the service bound to this nodegroup.
__count	<Double>	Read-write	count parameter

Operations

(click to see Properties)

ADD:| DELETE:| GET| GET (ALL)| COUNT

Some options that you can use for each operations:

- **Getting warnings in response:**NITRO allows you to get warnings in an operation by specifying the "warning" query parameter as "yes". For example, to get warnings while connecting to the Citrix ADC appliance, the URL is as follows:

  http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/login?warning=yes

  If any, the warnings are displayed in the response payload with the HTTP code "209 X-NITRO-WARNING".

- **Authenticated access for individual NITRO operations:**NITRO allows you to logon to the Citrix ADC appliance to perform individual operations. You can use this option instead of creating a NITRO session (using the login object) and then using that session to perform all operations,

  To do this, you must specify the username and password in the request header of the NITRO request as follows:

  X-NITRO-USER:<username>

  X-NITRO-PASS:<password>

  Note: In such cases, make sure that the request header DOES not include the following:

  Cookie:NITRO_AUTH_TOKEN=<tokenvalue>

`*Note:`*

Mandatory parameters are marked in redand placeholder content is marked in <green>.

add

URL:http://<Citrix-ADC-IP-address(NSIP)/nitro/v1/config/clusternodegroup_service_binding

HTTP Method:PUT

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Content-Type:application/json

Request Payload:

```
{

"clusternodegroup_service_binding":{

<b>"name":<String_value>,

</b>"service":<String_value>

}}
```

Response:

HTTP Status Code on Success: 201 Created HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error

delete

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/clusternodegroup_service_binding

Query-parameters:

args

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/clusternodegroup_service_binding?**args=name:<String_value>,service:<String_value>**

HTTP Method:DELETE

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue>

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error

get

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/clusternodegroup_service_binding

Query-parameters:

filter

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/clusternodegroup_service_binding?**filter=property-name1:property-value1,property-name2:property-value2**

Use this query-parameter to get the filtered set of clusternodegroup_service_binding resources configured on Citrix ADC. Filtering can be done on any of the properties of the resource.

pagination

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/clusternodegroup_service_binding?**pagesize=#no;pageno=#no**

Use this query-parameter to get the clusternodegroup_service_binding resources in chunks.

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "clusternodegroup_service_binding": [ {

"name":<String_value>,

"service":<String_value>

}]}
```

get (all)

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/clusternodegroup_service_binding

Query-parameters:

bulkbindings

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/clusternodegroup_service_binding?**bulkbindings=yes**

NITRO allows you to fetch bindings in bulk.

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "clusternodegroup_service_binding": [ {

"name":<String_value>,

"service":<String_value>

}]}
```

count

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/clusternodegroup_service_binding?**args=;count=yes**

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "clusternodegroup_service_binding": [ { "__count": "#no"} ] }
```

clusternodegroup_streamidentifier_binding

Binding object showing the streamidentifier that can be bound to clusternodegroup.

Properties

(click to see Operations)

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
name	<String>	Read-write	Name of the nodegroup to which you want to bind a cluster node or an entity. Minimum length = 1
identifiername	<String>	Read-write	stream identifier and rate limit identifier that need to be bound to this nodegroup.

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
__count	<Double>	Read-write	count parameter

Operations

(click to see Properties)

ADD:| DELETE:| GET| GET (ALL)| COUNT

Some options that you can use for each operations:

- **Getting warnings in response:**NITRO allows you to get warnings in an operation by specifying the "warning" query parameter as "yes". For example, to get warnings while connecting to the Citrix ADC appliance, the URL is as follows:

  http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/login?warning=yes

  If any, the warnings are displayed in the response payload with the HTTP code "209 X-NITRO-WARNING".
- **Authenticated access for individual NITRO operations:**NITRO allows you to logon to the Citrix ADC appliance to perform individual operations. You can use this option instead of creating a NITRO session (using the login object) and then using that session to perform all operations,

  To do this, you must specify the username and password in the request header of the NITRO request as follows:

  X-NITRO-USER:<username>

  X-NITRO-PASS:<password>

  Note: In such cases, make sure that the request header DOES not include the following:

  Cookie:NITRO_AUTH_TOKEN=<tokenvalue>

`*Note:`*

Mandatory parameters are marked in redand placeholder content is marked in <green>.

add

URL:http://<Citrix-ADC-IP-address(NSIP)/nitro/v1/config/clusternodegroup_streamidentifier_binding

HTTP Method:PUT

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Content-Type:application/json

Request Payload:

```
{

"clusternodegroup_streamidentifier_binding":{

<b>"name":<String_value>,

</b>"identifiername":<String_value>

}}
```

Response:

HTTP Status Code on Success: 201 Created HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error

delete

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/clusternodegroup_streamidentifier_binding/name_value<String>

Query-parameters:

args

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/clusternodegroup_streamidentifier_binding/name_value<String>?**args=identifiername:<String_value>**

HTTP Method:DELETE

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue>

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error

get

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/clusternodegroup_streamidentifier_binding/name_value<String>

Query-parameters:

filter

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/clusternodegroup_streamidentifier_binding/name_value<String>?**filter=property-name1:property-value1,property-name2:property-value2**

Use this query-parameter to get the filtered set of clusternodegroup_streamidentifier_binding resources configured on Citrix ADC. Filtering can be done on any of the properties of the resource.

pagination

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/clusternodegroup_streamidentifier_binding/name_value<String>?**pagesize=#no;pageno=#no**

Use this query-parameter to get the clusternodegroup_streamidentifier_binding resources in chunks.

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "clusternodegroup_streamidentifier_binding": [ {

"name":<String_value>,

"identifiername":<String_value>

}]}
```

get (all)

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/clusternodegroup_streamidentifier_binding

Query-parameters:

bulkbindings

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/clusternodegroup_streamidentifier_binding?**bulkbindings=yes**

NITRO allows you to fetch bindings in bulk.

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "clusternodegroup_streamidentifier_binding": [ {
```

```
"name":<String_value>,

"identifiername":<String_value>

}]}
```

count

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/clusternodegroup_streamidentifier_binding/name_value<String>?**count=yes**

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{"clusternodegroup_streamidentifier_binding": [ { "__count": "#no"} ] }
```

clusternodegroup_vpnvserver_binding

Binding object showing the vpnvserver that can be bound to clusternodegroup.

Properties

(click to see Operations)

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
vserver	<String>	Read-write	vserver that need to be bound to this nodegroup.
name	<String>	Read-write	Name of the nodegroup. The name uniquely identifies the nodegroup on the cluster. Minimum length = 1
__count	<Double>	Read-write	count parameter

Operations

(click to see Properties)

ADD:| DELETE:| GET| GET (ALL)| COUNT

Some options that you can use for each operations:

- **Getting warnings in response:**NITRO allows you to get warnings in an operation by specifying the "warning" query parameter as "yes". For example, to get warnings while connecting to the Citrix ADC appliance, the URL is as follows:

  http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/login?warning=yes

  If any, the warnings are displayed in the response payload with the HTTP code "209 X-NITRO-WARNING".

- **Authenticated access for individual NITRO operations:**NITRO allows you to logon to the Citrix ADC appliance to perform individual operations. You can use this option instead of creating a NITRO session (using the login object) and then using that session to perform all operations,

  To do this, you must specify the username and password in the request header of the NITRO request as follows:

  X-NITRO-USER:<username>

  X-NITRO-PASS:<password>

Note: In such cases, make sure that the request header DOES not include the following:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue>

`*Note:` *

Mandatory parameters are marked in redand placeholder content is marked in <green>.

add

URL:http://<Citrix-ADC-IP-address(NSIP)/nitro/v1/config/clusternodegroup_vpnvserver_binding

HTTP Method:PUT

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Content-Type:application/json

Request Payload:

```
{

"clusternodegroup_vpnvserver_binding":{

<b>"name":<String_value>,

</b>"vserver":<String_value>

}}
```

Response:

HTTP Status Code on Success: 201 Created HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error

delete

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/clusternodegroup_vpnvserver_binding

Query-parameters:

args

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/clusternodegroup_vpnvserver_binding?**args=name:<String_value>,vserver:<String_value>**

HTTP Method:DELETE

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue>

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error

get

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/clusternodegroup_vpnvserver_binding

Query-parameters:

filter

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/clusternodegroup_vpnvserver_binding?**filter=property-name1:property-value1,property-name2:property-value2**

Use this query-parameter to get the filtered set of clusternodegroup_vpnvserver_binding resources configured on Citrix ADC. Filtering can be done on any of the properties of the resource.

pagination

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/clusternodegroup_vpnvserver_binding?**pagesize=#no;pageno=#no**

Use this query-parameter to get the clusternodegroup_vpnvserver_binding resources in chunks.

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "clusternodegroup_vpnvserver_binding": [ {

"vserver":<String_value>,

"name":<String_value>

}]}
```

get (all)

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/clusternodegroup_vpnvserver_binding

Query-parameters:

bulkbindings

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/clusternodegroup_vpnvserver_binding?**bulkbindings=yes**

NITRO allows you to fetch bindings in bulk.

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "clusternodegroup_vpnvserver_binding": [ {

"vserver":<String_value>,

"name":<String_value>

}]}
```

count

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/clusternodegroup_vpnvserver_binding?**args=;count=yes**

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "clusternodegroup_vpnvserver_binding": [ { "__count": "#no"} ] }
```

clusternode_binding

Binding object which returns the resources bound to clusternode.

Properties

(click to see Operations)

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
nodeid	<Double>	Read-write	ID of the cluster node for which to display information. If an ID is not provided, information about all nodes is shown. Default value: 255 Minimum value = 0 Maximum value = 31
clusternode_routemonitor_binding	<clusternode_routemonitor_binding[]>	Read-only	routemonitor that can be bound to clusternode.

Operations

(click to see Properties)

GET| GET (ALL)

Some options that you can use for each operations:

- **Getting warnings in response:**NITRO allows you to get warnings in an operation by specifying the "warning" query parameter as "yes". For example, to get warnings while connecting to the Citrix ADC appliance, the URL is as follows:

  http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/login?warning=yes

  If any, the warnings are displayed in the response payload with the HTTP code "209 X-NITRO-WARNING".

- **Authenticated access for individual NITRO operations:**NITRO allows you to logon to the Citrix ADC appliance to perform individual operations. You can use this option instead of creating a NITRO session (using the login object) and then using that session to perform all operations,

  To do this, you must specify the username and password in the request header of the NITRO request as follows:

  X-NITRO-USER:<username>

  X-NITRO-PASS:<password>

  Note: In such cases, make sure that the request header DOES not include the following:

  Cookie:NITRO_AUTH_TOKEN=<tokenvalue>

`*Note:`*

Mandatory parameters are marked in redand placeholder content is marked in <green>.

get

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/clusternode_binding/nodeid_value<Double>

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "clusternode_binding": [ {
```

```
"nodeid":<Double_value>,

"clusternode_routemonitor_binding":<clusternode_routemonitor_binding[]_val
ue>

}]}
```

get (all)

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/clusternode_binding

Query-parameters:

bulkbindings

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/clusternode_binding?**bulkbindings=yes**

NITRO allows you to fetch bindings in bulk.

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "clusternode_binding": [ {

"nodeid":<Double_value>,

"clusternode_routemonitor_binding":<clusternode_routemonitor_binding[]_val
ue>
```

```
}]}
```

clusternode_routemonitor_binding

Binding object showing the routemonitor that can be bound to clusternode.

Properties

(click to see Operations)

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
nodeid	<Double>	Read-write	A number that uniquely identifies the cluster node. . Minimum value = 0 Maximum value = 31
routemonitor	<String>	Read-write	The IP address (IPv4 or IPv6).
netmask	<String>	Read-write	The netmask.
routemonstate	<Double>	Read-only	Current routemonstate.
__count	<Double>	Read-only	count parameter

Operations

(click to see Properties)

ADD:| DELETE:| GET| GET (ALL)| COUNT

Some options that you can use for each operations:

- **Getting warnings in response:**NITRO allows you to get warnings in an operation by specifying the "warning" query parameter as "yes". For example, to get warnings while connecting to the Citrix ADC appliance, the URL is as follows:

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/login?warning=yes

If any, the warnings are displayed in the response payload with the HTTP code "209 X-NITRO-WARNING".

- **Authenticated access for individual NITRO operations:**NITRO allows you to logon to the Citrix ADC appliance to perform individual operations. You can use this option instead of creating a NITRO session (using the login object) and then using that session to perform all operations,

  To do this, you must specify the username and password in the request header of the NITRO request as follows:

  X-NITRO-USER:<username>

  X-NITRO-PASS:<password>

  Note: In such cases, make sure that the request header DOES not include the following:

  Cookie:NITRO_AUTH_TOKEN=<tokenvalue>

`*Note:`*

Mandatory parameters are marked in redand placeholder content is marked in <green>.

add

URL:http://<Citrix-ADC-IP-address(NSIP)/nitro/v1/config/clusternode_routemonitor_binding

HTTP Method:PUT

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Content-Type:application/json

Request Payload:

```
{

"clusternode_routemonitor_binding":{

<b>"nodeid":<Double_value>,

</b><b>"routemonitor":<String_value>,
```

```
</b>"netmask":<String_value>

}}
```

Response:

HTTP Status Code on Success: 201 Created HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error

delete

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/clusternode_routemonitor_binding/nodeid_value<Double>

Query-parameters:

args

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/clusternode_routemonitor_binding/nodeid_value<Double>?**args=routemonitor:<String_value>,netmask:<String_value>**

HTTP Method:DELETE

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue>

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error

get

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/clusternode_routemonitor_binding/nodeid_value<Double>

Query-parameters:

filter

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/clusternode_routemonitor_binding/nodeid_value<Double>?**filter=property-name1:property-value1,property-name2:property-value2**

Use this query-parameter to get the filtered set of clusternode_routemonitor_binding resources configured on Citrix ADC. Filtering can be done on any of the properties of the resource.

pagination

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/clusternode_routemonitor_binding/nodeid_value<Double>?**pagesize=#no;pageno=#no**

Use this query-parameter to get the clusternode_routemonitor_binding resources in chunks.

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "clusternode_routemonitor_binding": [ {

"nodeid":<Double_value>,

"routemonitor":<String_value>,

"netmask":<String_value>,

"routemonstate":<Double_value>

}]}
```

get (all)

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/clusternode_routemonitor_binding

Query-parameters:

bulkbindings

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/clusternode_routemonitor_binding?**bulkbindings=yes**

NITRO allows you to fetch bindings in bulk.

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "clusternode_routemonitor_binding": [ {

"nodeid":<Double_value>,

"routemonitor":<String_value>,

"netmask":<String_value>,

"routemonstate":<Double_value>

}]}
```

count

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/clusternode_routemonitor_binding/nodeid_value<Double>?**count=yes**

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{"clusternode_routemonitor_binding": [ { "__count": "#no"} ] }
```

clusterpropstatus

Configuration for 0 resource.

Properties

(click to see Operations)

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
nodeid	<Double>	Read-write	Unique number that identifies the cluster node. Minimum value = 0

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
			Maximum value = 31
numpropcmdfailed	<Double>	Read-only	Number of propogated comands failed on this node. Minimum value = 0 Maximum value = 20
cmdstrs	<String>	Read-only	.
__count	<Double>	Read-only	count parameter

Operations

(click to see Properties)

CLEAR| GET (ALL)| COUNT

Some options that you can use for each operations:

- **Getting warnings in response:**NITRO allows you to get warnings in an operation by specifying the "warning" query parameter as "yes". For example, to get warnings while connecting to the Citrix ADC appliance, the URL is as follows:

  http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/login?warning=yes

  If any, the warnings are displayed in the response payload with the HTTP code "209 X-NITRO-WARNING".

- **Authenticated access for individual NITRO operations:**NITRO allows you to logon to the Citrix ADC appliance to perform individual operations. You can use this option instead of creating a NITRO session (using the login object) and then using that session to perform all operations,

  To do this, you must specify the username and password in the request header of the NITRO request as follows:

  X-NITRO-USER:<username>

X-NITRO-PASS:<password>

Note: In such cases, make sure that the request header DOES not include the following:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue>

`*Note:`*

Mandatory parameters are marked in redand placeholder content is marked in <green>.

clear

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/clusterpropstatus?**action=clear**

HTTP Method:POST

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Content-Type:application/json

Request Payload:

```
{"clusterpropstatus":{

}}
```

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error

get (all)

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/clusterpropstatus

Query-parameters:

args

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/clusterpropstatus?**args=nodeid:<Double_value>**

Use this query-parameter to get clusterpropstatus resources based on additional properties.

attrs

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/clusterpropstatus?**attrs=property-name1,property-name2**

Use this query parameter to specify the resource details that you want to retrieve.

filter

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/clusterpropstatus?**filter=property-name1:property-val1,property-name2:property-val2**

Use this query-parameter to get the filtered set of clusterpropstatus resources configured on Citrix ADC. Filtering can be done on any of the properties of the resource.

view

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/clusterpropstatus?**view=summary**

Note: By default, the retrieved results are displayed in detail view (?view=detail).

pagination

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/clusterpropstatus?**pagesize=#no;pageno=#no**

Use this query-parameter to get the clusterpropstatus resources in chunks.

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "clusterpropstatus": [ {

nodeid:<Double_value>"numpropcmdfailed":<Double_value>,

"cmdstrs":<String_value>

}]}
```

count

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/clusterpropstatus?**count=yes**

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "clusterpropstatus": [ { "__count": "#no"} ] }
```

clustersync

Configuration for sync resource.

Operations

(click to see Properties)

FORCE

Some options that you can use for each operations:

- **Getting warnings in response:**NITRO allows you to get warnings in an operation by specifying the "warning" query parameter as "yes". For example, to get warnings while connecting to the Citrix ADC appliance, the URL is as follows:

  http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/login?warning=yes

  If any, the warnings are displayed in the response payload with the HTTP code "209 X-NITRO-WARNING".

- **Authenticated access for individual NITRO operations:**NITRO allows you to logon to the Citrix ADC appliance to perform individual operations. You can use this option instead of creating a NITRO session (using the login object) and then using that session to perform all operations,

  To do this, you must specify the username and password in the request header of the NITRO request as follows:

  X-NITRO-USER:<username>

  X-NITRO-PASS:<password>

  Note: In such cases, make sure that the request header DOES not include the following:

  Cookie:NITRO_AUTH_TOKEN=<tokenvalue>

`*Note:`*

Mandatory parameters are marked in redand placeholder content is marked in <green>.

Force

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/clustersync?**action=Force**

HTTP Method:POST

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Content-Type:application/json

Request Payload:

```
{"clustersync":{

}}
```

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error

Compression

Compression

Compression configuration. The system's feature for compressing HTTP responses to compression-aware browsers.

NAME	DESCRIPTION
cmpaction	Configuration for compression action
cmpglobal_binding	Binding object showing the resources that can be bound to cmpglobal
cmpglobal_cmppolicy_binding	Binding object showing the cmppolicy that can be bound to cmpglobal
cmpparameter	Configuration for CMP parameter
cmppolicy	Configuration for compression policy
cmppolicy_binding	Binding object showing the resources that can be bound to cmppolicy
cmppolicy_cmpglobal_binding	Binding object showing the cmpglobal that can be bound to cmppolicy
cmppolicy_cmppolicylabel_binding	Binding object showing the cmppolicylabel that can be bound to cmppolicy

NAME	DESCRIPTION
cmppolicy_crvserver_binding	Binding object showing the crvserver that can be bound to cmppolicy
cmppolicy_csvserver_binding	Binding object showing the csvserver that can be bound to cmppolicy
cmppolicy_lbvserver_binding	Binding object showing the lbvserver that can be bound to cmppolicy
cmppolicylabel	Configuration for compression policy label
cmppolicylabel_binding	Binding object showing the resources that can be bound to cmppolicylabel
cmppolicylabel_cmppolicy_binding	Binding object showing the cmppolicy that can be bound to cmppolicylabel
cmppolicylabel_policybinding_binding	Binding object showing the policybinding that can be bound to cmppolicylabel

cmpaction

Configuration for compression action resource.

Properties

(click to see Operations)

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
name	<String>	Read-write	Name of the compression action. Must begin with an ASCII alphabetic or underscore (_) character, and must contain only ASCII alphanumeric, underscore, hash (#), period (.), space, colon (:), at (@), equals (=), and

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
			hyphen (-) characters. Can be changed after the action is added. The following requirement applies only to the Citrix ADC CLI: If the name includes one or more spaces, enclose the name in double or single quotation marks (for example, "my cmp action" or 'my cmp action'). Minimum length = 1
cmptype	<String>	Read-write	Type of compression performed by this action. Available settings function as follows: * COMPRESS - Apply GZIP or DEFLATE compression to the response, depending on the request header. Prefer GZIP. * GZIP - Apply GZIP compression.

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
			* DEFLATE - Apply DEFLATE compression. * NOCOMPRESS - Do not compress the response if the request matches a policy that uses this action. Possible values = compress, gzip, deflate, nocompress
addvaryheader	<String>	Read-write	Control insertion of the Vary header in HTTP responses compressed by Citrix ADC. Intermediate caches store different versions of the response for different values of the headers present in the Vary response header. Default value: GLOBAL Possible values = GLOBAL, DISABLED, ENABLED
varyheadervalue	<String>	Read-write	The value of the HTTP Vary header for compressed responses.

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
			Minimum length = 1
deltatype	<String>	Read-write	The type of delta action (if delta type compression action is defined). Default value: PERURL Possible values = PERURL, PERPOLICY
newname	<String>	Read-write	New name for the compression action. Must begin with an ASCII alphabetic or underscore (_) character, and must contain only ASCII alphanumeric, underscore, hash (#), period (.), space, colon (:), at (@), equals (=), and hyphen (-) characters. Choose a name that can be correlated with the function that the action performs. The following requirement applies only to the Citrix ADC CLI:

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
			If the name includes one or more spaces, enclose the name in double or single quotation marks (for example, "my cmp action" or 'my cmp action'). Minimum length = 1
isdefault	<Boolean>	Read-only	A value of true is returned if it is a default policy.
__count	<Double>	Read-only	count parameter

Operations

(click to see Properties)

ADD| DELETE| UPDATE| UNSET| RENAME| GET (ALL)| GET| COUNT

Some options that you can use for each operations:

- **Getting warnings in response:**NITRO allows you to get warnings in an operation by specifying the "warning" query parameter as "yes". For example, to get warnings while connecting to the Citrix ADC appliance, the URL is as follows:

  http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/login?warning=yes

  If any, the warnings are displayed in the response payload with the HTTP code "209 X-NITRO-WARNING".

- **Authenticated access for individual NITRO operations:**NITRO allows you to logon to the Citrix ADC appliance to perform individual operations. You can use this option instead of creating a NITRO session (using the login object) and then using that session to perform all operations,

  To do this, you must specify the username and password in the request header of the NITRO request as follows:

X-NITRO-USER:<username>

X-NITRO-PASS:<password>

Note: In such cases, make sure that the request header DOES not include the following:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue>

`*Note:`*

Mandatory parameters are marked in redand placeholder content is marked in <green>.

add

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/cmpaction

HTTP Method:POST

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Content-Type:application/json

Request Payload:

```
{"cmpaction":{

<b>"name":<String_value>,

</b><b>"cmptype":<String_value>,

</b>"addvaryheader":<String_value>,

"varyheadervalue":<String_value>,

"deltatype":<String_value>

}}
```

Response:

HTTP Status Code on Success: 201 Created HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error

delete

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/cmpaction/name_value<String>

HTTP Method:DELETE

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue>

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error

update

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/cmpaction

HTTP Method:PUT

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Content-Type:application/json

Request Payload:

```
{"cmpaction":{

<b>"name":<String_value>,

</b>"cmptype":<String_value>,

"addvaryheader":<String_value>,

"varyheadervalue":<String_value>

}}
```

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error

unset

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/cmpaction?**action=unset**

HTTP Method:POST

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Content-Type:application/json

Request Payload:

```
{"cmpaction":{

<b>"name":<String_value>,

</b>"addvaryheader":true

}}
```

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error

rename

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/cmpaction?**action=rename**

HTTP Method:POST

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Content-Type:application/json

Request Payload:

```
{"cmpaction":{

<b>"name":<String_value>,

</b><b>"newname":<String_value>

</b>}}
```

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error

get (all)

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/cmpaction

Query-parameters:

attrs

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/cmpaction?**attrs=property-name1,property-name2**

Use this query parameter to specify the resource details that you want to retrieve.

filter

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/cmpaction?**filter=property-name1:property-val1,property-name2:property-val2**

Use this query-parameter to get the filtered set of cmpaction resources configured on Citrix ADC. Filtering can be done on any of the properties of the resource.

view

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/cmpaction?**view=summary**

Note: By default, the retrieved results are displayed in detail view (?view=detail).

pagination

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/cmpaction?**pagesize=#no;pageno=#no**

Use this query-parameter to get the cmpaction resources in chunks.

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "cmpaction": [ {

"name":<String_value>,

"cmptype":<String_value>,

"deltatype":<String_value>,

"addvaryheader":<String_value>,

"varyheadervalue":<String_value>,

"isdefault":<Boolean_value>

}]}
```

get

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/cmpaction/name_value<String>

Query-parameters:

attrs

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/cmpaction/name_value<String>?**attrs=property-name1,property-name2**

Use this query parameter to specify the resource details that you want to retrieve.

view

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/cmpaction/name_value<String>?**view=summary**

Note: By default, the retrieved results are displayed in detail view (?view=detail).

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "cmpaction": [ {

"name":<String_value>,

"cmptype":<String_value>,
```

```
"deltatype":<String_value>,

"addvaryheader":<String_value>,

"varyheadervalue":<String_value>,

"isdefault":<Boolean_value>

}]}
```

count

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/cmpaction?**count=yes**

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "cmpaction": [ { "__count": "#no"} ] }
```

cmpglobal_binding

Binding object which returns the resources bound to cmpglobal.

Properties

(click to see Operations)

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
cmpglobal_cmppolicy_binding	<cmpglobal_cmppolicy_binding[]>	Read-only	cmppolicy that can be bound to cmpglobal.

Operations

(click to see Properties)

GET| GET (ALL)

Some options that you can use for each operations:

- **Getting warnings in response:**NITRO allows you to get warnings in an operation by specifying the "warning" query parameter as "yes". For example, to get warnings while connecting to the Citrix ADC appliance, the URL is as follows:

  http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/login?warning=yes

  If any, the warnings are displayed in the response payload with the HTTP code "209 X-NITRO-WARNING".

- **Authenticated access for individual NITRO operations:**NITRO allows you to logon to the Citrix ADC appliance to perform individual operations. You can use this option instead of creating a NITRO session (using the login object) and then using that session to perform all operations,

  To do this, you must specify the username and password in the request header of the NITRO request as follows:

  X-NITRO-USER:<username>

  X-NITRO-PASS:<password>

  Note: In such cases, make sure that the request header DOES not include the following:

  Cookie:NITRO_AUTH_TOKEN=<tokenvalue>

`*Note:`*

Mandatory parameters are marked in redand placeholder content is marked in <green>.

get

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/cmpglobal_binding

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "cmpglobal_binding": [ {

"cmpglobal_cmppolicy_binding":<cmpglobal_cmppolicy_binding[]_value>

}]}
```

get (all)

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/cmpglobal_binding

Query-parameters:

bulkbindings

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/cmpglobal_binding?**bulkbindings=yes**

NITRO allows you to fetch bindings in bulk.

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "cmpglobal_binding": [ {

"cmpglobal_cmppolicy_binding":<cmpglobal_cmppolicy_binding[]_value>

}]}
```

cmpglobal_cmppolicy_binding

Binding object showing the cmppolicy that can be bound to cmpglobal.

Properties

(click to see Operations)

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
priority	<Double>	Read-write	Positive integer specifying the priority of the policy. The lower the number, the higher the priority. By default, polices within a label are evaluated in the order of their priority numbers. In the configuration utility, you can click the Priority field and edit the priority level or

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
			drag the entry to a new position in the list. If you drag the entry to a new position, the priority level is updated automatically.
globalbindtype	<String>	Read-write	. Default value: SYSTEM_GLOBAL Possible values = SYSTEM_GLOBAL, VPN_GLOBAL, RNAT_GLOBAL
policyname	<String>	Read-write	The name of the globally bound HTTP compression policy.
labelname	<String>	Read-write	Name of the label to invoke if the current policy rule evaluates to TRUE. Applicable only to advanced (default-syntax) policies.
state	<String>	Read-write	The current state of the policy binding. This attribute is relevant only for CLASSIC policies. Possible values = ENABLED, DISABLED
gotopriorityexpression	<String>	Read-write	Expression or other value specifying the priority of the next

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
			policy, within the policy label, to evaluate if the current policy evaluates to TRUE. Specify one of the following values: * NEXT - Evaluate the policy with the next higher numbered priority. * END - Stop evaluation. * USE_INVOCATION_RESULT - Applicable if this policy invokes another policy label. If the final goto in the invoked policy label has a value of END, the evaluation stops. If the final goto is anything other than END, the current policy label performs a NEXT. * An expression that evaluates to a number. If you specify an expression, it's evaluation result determines the next policy to evaluate, as follows: * If the expression evaluates to a higher numbered priority, that policy is evaluated next. * If the expression evaluates to the

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
			priority of the current policy, the policy with the next higher priority number is evaluated next. * If the expression evaluates to a priority number that is numerically higher than the highest priority number, policy evaluation ends. An UNDEF event is triggered if: * The expression is invalid. * The expression evaluates to a priority number that is numerically lower than the current policy's priority. * The expression evaluates to a priority number that is between the current policy's priority number (say, 30) and the highest priority number (say, 100), but does not match any configured priority number (for example, the expression evaluates to the number 85). This example assumes that the priority number increments by 10 for every successive

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
			policy, and therefore a priority number of 85 does not exist in the policy label.
invoke	<Boolean>	Read-write	Invoke policies bound to a virtual server or a policy label. After the invoked policies are evaluated, the flow returns to the policy with the next priority. Applicable only for default-syntax policies.
type	<String>	Read-write	Bind point to which the policy is bound. Possible values = REQ_OVERRIDE, REQ_DEFAULT, RES_OVERRIDE, RES_DEFAULT
labeltype	<String>	Read-write	Type of policy label invocation. This argument is relevant only for advanced (default-syntax) policies. Possible values = reqvserver, resvserver, policylabel
numpol	<Double>	Read-only	The number of policies bound to the bindpoint.

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
policytype	<String>	Read-only	Policy type (Classic/ Advanced) to be bound.Used for display. Possible values = Classic Policy, Advanced Policy
__count	<Double>	Read-only	count parameter

Operations

(click to see Properties)

ADD:| DELETE:| GET| GET (ALL)| COUNT

Some options that you can use for each operations:

- **Getting warnings in response:**NITRO allows you to get warnings in an operation by specifying the "warning" query parameter as "yes". For example, to get warnings while connecting to the Citrix ADC appliance, the URL is as follows:

  http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/login?warning=yes

  If any, the warnings are displayed in the response payload with the HTTP code "209 X-NITRO-WARNING".

- **Authenticated access for individual NITRO operations:**NITRO allows you to logon to the Citrix ADC appliance to perform individual operations. You can use this option instead of creating a NITRO session (using the login object) and then using that session to perform all operations,

  To do this, you must specify the username and password in the request header of the NITRO request as follows:

  X-NITRO-USER:<username>

  X-NITRO-PASS:<password>

  Note: In such cases, make sure that the request header DOES not include the following:

  Cookie:NITRO_AUTH_TOKEN=<tokenvalue>

`*Note:` *

Mandatory parameters are marked in redand placeholder content is marked in <green>.

add

URL:http://<Citrix-ADC-IP-address(NSIP)/nitro/v1/config/cmpglobal_cmppolicy_binding

HTTP Method:PUT

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Content-Type:application/json

Request Payload:

```
{

"cmpglobal_cmppolicy_binding":{

<b>"policyname":<String_value>,

</b>"priority":<Double_value>,

"state":<String_value>,

"gotopriorityexpression":<String_value>,

"type":<String_value>,

"invoke":<Boolean_value>,

"labeltype":<String_value>,

"labelname":<String_value>

}}
```

Response:

HTTP Status Code on Success: 201 Created HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error

delete

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/cmpglobal_cmppolicy_binding

Query-parameters:

args

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/cmpglobal_cmppolicy_binding?**args=policyname:<String_value>,type:<String_value>,priority:<Double_value>**

HTTP Method:DELETE

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue>

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error

get

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/cmpglobal_cmppolicy_binding

Query-parameters:

filter

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/cmpglobal_cmppolicy_binding?**filter=property-name1:property-value1,property-name2:property-value2**

Use this query-parameter to get the filtered set of cmpglobal_cmppolicy_binding resources configured on Citrix ADC. Filtering can be done on any of the properties of the resource.

pagination

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/cmpglobal_cmppolicy_binding?**pagesize=#no;pageno=#no**

Use this query-parameter to get the cmpglobal_cmppolicy_binding resources in chunks.

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "cmpglobal_cmppolicy_binding": [ {

"priority":<Double_value>,

"globalbindtype":<String_value>,

"policyname":<String_value>,

"labelname":<String_value>,

"state":<String_value>,

"gotopriorityexpression":<String_value>,

"invoke":<Boolean_value>,

"type":<String_value>,

"labeltype":<String_value>,

"numpol":<Double_value>,

"policytype":<String_value>
```

```
}]}
```

get (all)

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/cmpglobal_cmppolicy_binding

Query-parameters:

bulkbindings

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/cmpglobal_cmppolicy_binding?**bulkbindings=yes**

NITRO allows you to fetch bindings in bulk.

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "cmpglobal_cmppolicy_binding": [ {

"priority":<Double_value>,

"globalbindtype":<String_value>,

"policyname":<String_value>,

"labelname":<String_value>,
```

```
"state":<String_value>,

"gotopriorityexpression":<String_value>,

"invoke":<Boolean_value>,

"type":<String_value>,

"labeltype":<String_value>,

"numpol":<Double_value>,

"policytype":<String_value>

}]}
```

count

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/cmpglobal_cmppolicy_binding?**count=yes**

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "cmpglobal_cmppolicy_binding": [ { "__count": "#no"} ] }
```

cmpparameter

Configuration for CMP parameter resource.

Properties

(click to see Operations)

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
cmplevel	<String>	Read-write	Specify a compression level. Available settings function as follows: * Optimal - Corresponds to a gzip GZIP level of 5-7. * Best speed - Corresponds to a gzip level of 1. * Best compression - Corresponds to a gzip level of 9. Default value: optimal Possible values = optimal, bestspeed, bestcompression
quantumsize	<Double>	Read-write	Minimum quantum of data to be filled before compression begins. Default value: 57344 Minimum value = 8

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
			Maximum value = 63488
servercmp	<String>	Read-write	Allow the server to send compressed data to the Citrix ADC. With the default setting, the Citrix ADC appliance handles all compression. Default value: ON Possible values = ON, OFF
heurexpiry	<String>	Read-write	Heuristic basefile expiry. Default value: OFF Possible values = ON, OFF
heurexpirythres	<Double>	Read-write	Threshold compression ratio for heuristic basefile expiry, multiplied by 100. For example, to set the threshold ratio to 1.25, specify 125. Default value: 100 Minimum value = 1 Maximum value = 1000
heurexpiryhistwt	<Double>	Read-write	For heuristic basefile expiry, weightage to be given to historical

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
			delta compression ratio, specified as percentage. For example, to give 25% weightage to historical ratio (and therefore 75% weightage to the ratio for current delta compression transaction), specify 25. Default value: 50 Minimum value = 1 Maximum value = 100
minressize	<Double>	Read-write	Smallest response size, in bytes, to be compressed.
cmpbypasspct	<Double>	Read-write	Citrix ADC CPU threshold after which compression is not performed. Range: 0 - 100. Default value: 100 Minimum value = 0 Maximum value = 100
cmponpush	<String>	Read-write	Citrix ADC does not wait for the quantum to be filled before starting to compress data. Upon receipt of a packet with a PUSH flag, the

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
			appliance immediately begins compression of the accumulated packets. Default value: DISABLED Possible values = ENABLED, DISABLED
policytype	<String>	Read-write	Type of policy. Available settings function as follows: * Classic - Classic policies evaluate basic characteristics of traffic and other data. Deprecated. * Advanced - Advanced policies (which have been renamed as default syntax policies) can perform the same type of evaluations as classic policies. They also enable you to analyze more data (for example, the body of an HTTP request) and to configure more operations in the policy rule (for example,

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
			transforming data in the body of a request into an HTTP header). Possible values = CLASSIC, ADVANCED
addvaryheader	<String>	Read-write	Control insertion of the Vary header in HTTP responses compressed by Citrix ADC. Intermediate caches store different versions of the response for different values of the headers present in the Vary response header. Default value: DISABLED Possible values = ENABLED, DISABLED
varyheadervalue	<String>	Read-write	The value of the HTTP Vary header for compressed responses. If this argument is not specified, a default value of "Accept-Encoding" will be used. Minimum length = 1

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
externalcache	<String>	Read-write	Enable insertion of Cache-Control: private response directive to indicate response message is intended for a single user and must not be cached by a shared or proxy cache. Default value: NO Possible values = YES, NO

Operations

(click to see Properties)

UPDATE| UNSET| GET (ALL)

Some options that you can use for each operations:

- **Getting warnings in response:**NITRO allows you to get warnings in an operation by specifying the "warning" query parameter as "yes". For example, to get warnings while connecting to the Citrix ADC appliance, the URL is as follows:

  http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/login?warning=yes

  If any, the warnings are displayed in the response payload with the HTTP code "209 X-NITRO-WARNING".

- **Authenticated access for individual NITRO operations:**NITRO allows you to logon to the Citrix ADC appliance to perform individual operations. You can use this option instead of creating a NITRO session (using the login object) and then using that session to perform all operations,

  To do this, you must specify the username and password in the request header of the NITRO request as follows:

  X-NITRO-USER:<username>

  X-NITRO-PASS:<password>

Note: In such cases, make sure that the request header DOES not include the following:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue>

`*Note:` *

Mandatory parameters are marked in redand placeholder content is marked in <green>.

update

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/cmpparameter

HTTP Method:PUT

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Content-Type:application/json

Request Payload:

```
{"cmpparameter":{

"cmplevel":<String_value>,

"quantumsize":<Double_value>,

"servercmp":<String_value>,

"heurexpiry":<String_value>,

"heurexpirythres":<Double_value>,

"heurexpiryhistwt":<Double_value>,

"minressize":<Double_value>,

"cmpbypasspct":<Double_value>,

"cmponpush":<String_value>,

"policytype":<String_value>,

"addvaryheader":<String_value>,
```

```
"varyheadervalue":<String_value>,

"externalcache":<String_value>

}}
```

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error

unset

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/cmpparameter?**action=unset**

HTTP Method:POST

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Content-Type:application/json

Request Payload:

```
{"cmpparameter":{

"policytype":true,

"cmplevel":true,

"quantumsize":true,

"servercmp":true,

"heurexpiry":true,

"heurexpirythres":true,

"heurexpiryhistwt":true,
```

```
"minressize":true,

"cmpbypasspct":true,

"cmponpush":true,

"addvaryheader":true,

"varyheadervalue":true,

"externalcache":true

}}
```

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error

get (all)

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/cmpparameter

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "cmpparameter": [ {
"cmplevel":<String_value>,
"quantumsize":<Double_value>,
"servercmp":<String_value>,
"heurexpiry":<String_value>,
"heurexpirythres":<Double_value>,
"heurexpiryhistwt":<Double_value>,
"minressize":<Double_value>,
"cmpbypasspct":<Double_value>,
"cmponpush":<String_value>,
"policytype":<String_value>,
"addvaryheader":<String_value>,
"varyheadervalue":<String_value>,
"externalcache":<String_value>
}]}
```

cmppolicy

Configuration for compression policy resource.

Properties

(click to see Operations)

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
name	<String>	Read-write	Name of the HTTP compression policy. Must begin with an ASCII alphabetic or underscore (_) character, and must contain only ASCII alphanumeric, underscore, hash (#), period (.), space, colon (:), at (@), equals (=), and hyphen (-) characters. Can be changed after the policy is created. The following requirement applies only to the Citrix ADC CLI: If the name includes one or more spaces, enclose the name in double or single quotation marks (for example, "my cmp policy" or 'my cmp policy'). Minimum length = 1
rule	<String>	Read-write	Expression that determines which HTTP requests or responses match the compression policy.

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
			The following requirements apply only to the Citrix ADC CLI: * If the expression includes one or more spaces, enclose the entire expression in double quotation marks. * If the expression itself includes double quotation marks, escape the quotations by using the \ character. * Alternatively, you can use single quotation marks to enclose the rule, in which case you do not have to escape the double quotation marks.
resaction	<String>	Read-write	The built-in or user-defined compression action to apply to the response when the policy matches a request or response. Minimum length = 1

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
newname	<String>	Read-write	New name for the compression policy. Must begin with an ASCII alphabetic or underscore (_) character, and must contain only ASCII alphanumeric, underscore, hash (#), period (.), space, colon (:), at (@), equals (=), and hyphen (-) characters. Choose a name that reflects the function that the policy performs. The following requirement applies only to the Citrix ADC CLI: If the name includes one or more spaces, enclose the name in double or single quotation marks (for example, "my cmp policy" or 'my cmp policy'). Minimum length = 1
expressiontype	<String>	Read-only	Type of policy (Classic/Advanced) .

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
			Possible values = Classic Policy, Advanced Policy
reqaction	<String>	Read-only	The compression action to be performed on requests.
hits	<Double>	Read-only	Number of hits.
txbytes	<Double>	Read-only	Number of bytes transferred.
rxbytes	<Double>	Read-only	Number of bytes received.
clientttlb	<Double>	Read-only	Total client TTLB value.
clienttransactions	<Double>	Read-only	Number of client transactions.
serverttlb	<Double>	Read-only	Total server TTLB value.
servertransactions	<Double>	Read-only	Number of server transactions.
description	<String>	Read-only	Description of the policy.
isdefault	<Boolean>	Read-only	A value of true is returned if it is a default policy.
__count	<Double>	Read-only	count parameter

Operations

(click to see Properties)

ADD| DELETE| UPDATE| RENAME| GET (ALL)| GET| COUNT

Some options that you can use for each operations:

- **Getting warnings in response:**NITRO allows you to get warnings in an operation by specifying the "warning" query parameter as "yes". For example, to get warnings while connecting to the Citrix ADC appliance, the URL is as follows:

  http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/login?warning=yes

  If any, the warnings are displayed in the response payload with the HTTP code "209 X-NITRO-WARNING".

- **Authenticated access for individual NITRO operations:**NITRO allows you to logon to the Citrix ADC appliance to perform individual operations. You can use this option instead of creating a NITRO session (using the login object) and then using that session to perform all operations,

  To do this, you must specify the username and password in the request header of the NITRO request as follows:

  X-NITRO-USER:<username>

  X-NITRO-PASS:<password>

  Note: In such cases, make sure that the request header DOES not include the following:

  Cookie:NITRO_AUTH_TOKEN=<tokenvalue>

`*Note:`*

Mandatory parameters are marked in redand placeholder content is marked in <green>.

add

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/cmppolicy

HTTP Method:POST

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Content-Type:application/json

Request Payload:

```
{"cmppolicy":{

<b>"name":<String_value>,

</b><b>"rule":<String_value>,

</b><b>"resaction":<String_value>

</b>}}
```

Response:

HTTP Status Code on Success: 201 Created HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error

delete

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/cmppolicy/name_value<String>

HTTP Method:DELETE

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue>

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error

update

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/cmppolicy

HTTP Method:PUT

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Content-Type:application/json

Request Payload:

```
{"cmppolicy":{

<b>"name":<String_value>,

</b>"rule":<String_value>,

"resaction":<String_value>

}}
```

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error

rename

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/cmppolicy?**action=rename**

HTTP Method:POST

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Content-Type:application/json

Request Payload:

```
{"cmppolicy":{

<b>"name":<String_value>,

</b><b>"newname":<String_value>

</b>}}
```

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error

get (all)

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/cmppolicy

Query-parameters:

attrs

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/cmppolicy?**attrs=property-name1,property-name2**

Use this query parameter to specify the resource details that you want to retrieve.

filter

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/cmppolicy?**filter=property-name1:property-val1,property-name2:property-val2**

Use this query-parameter to get the filtered set of cmppolicy resources configured on Citrix ADC. Filtering can be done on any of the properties of the resource.

view

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/cmppolicy?**view=summary**

Note: By default, the retrieved results are displayed in detail view (?view=detail).

pagination

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/cmppolicy?**pagesize=#no;pageno=#no**

Use this query-parameter to get the cmppolicy resources in chunks.

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "cmppolicy": [ {

"name":<String_value>,

"expressiontype":<String_value>,

"rule":<String_value>,

"reqaction":<String_value>,

"resaction":<String_value>,

"hits":<Double_value>,

"txbytes":<Double_value>,

"rxbytes":<Double_value>,

"clientttlb":<Double_value>,

"clienttransactions":<Double_value>,

"serverttlb":<Double_value>,

"servertransactions":<Double_value>,

"description":<String_value>,

"isdefault":<Boolean_value>

}]}
```

get

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/cmppolicy/name_value<String>

Query-parameters:

attrs

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/cmppolicy/name_value<String>?**attrs=property-name1,property-name2**

Use this query parameter to specify the resource details that you want to retrieve.

view

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/cmppolicy/name_value<String>?**view=summary**

Note: By default, the retrieved results are displayed in detail view (?view=detail).

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "cmppolicy": [ {

"name":<String_value>,

"expressiontype":<String_value>,
```

```
"rule":<String_value>,

"reqaction":<String_value>,

"resaction":<String_value>,

"hits":<Double_value>,

"txbytes":<Double_value>,

"rxbytes":<Double_value>,

"clientttlb":<Double_value>,

"clienttransactions":<Double_value>,

"serverttlb":<Double_value>,

"servertransactions":<Double_value>,

"description":<String_value>,

"isdefault":<Boolean_value>

}]}
```

count

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/cmppolicy?**count=yes**

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "cmppolicy": [ { "__count": "#no"} ] }
```

cmppolicylabel

Configuration for compression policy label resource.

Properties

(click to see Operations)

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
labelname	<String>	Read-write	Name of the HTTP compression policy label. Must begin with a letter, number, or the underscore character (_). Additional characters allowed, after the first character, are the hyphen (-), period (.) pound sign (#), space ( ), at sign (@), equals (=), and colon (:). The name must be unique within the list of policy labels for compression policies. Can be renamed after the policy label is created.

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
			The following requirement applies only to the Citrix ADC CLI: If the name includes one or more spaces, enclose the name in double or single quotation marks (for example, "my cmp policylabel" or 'my cmp policylabel'). Minimum length = 1
type	<String>	Read-write	Type of packets (request packets or response) against which to match the policies bound to this policy label. Possible values = REQ, RES
newname	<String>	Read-write	New name for the compression policy label. Must begin with an ASCII alphabetic or underscore (_) character, and must contain only ASCII alphanumeric, underscore, hash (#), period (.), space, colon (:), at (@), equals (=), and

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
			hyphen (-) characters. The following requirement applies only to the Citrix ADC CLI: If the name includes one or more spaces, enclose the name in double or single quotation marks (for example, "my cmp policylabel" or 'my cmp policylabel'). Minimum length = 1
numpol	<Double>	Read-only	Number of polices bound to label.
hits	<Double>	Read-only	Number of times policy label was invoked.
priority	<Double>	Read-only	Specifies the priority of the policy.
gotopriorityexpression	<String>	Read-only	Expression specifying the priority of the next policy which will get evaluated if the current policy rule evaluates to TRUE.
labeltype	<String>	Read-only	Type of policy label invocation.

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
			Possible values = reqvserver, resvserver, policylabel
invoke_labelname	<String>	Read-only	Name of the label to invoke if the current policy evaluates to TRUE.
flowtype	<Double>	Read-only	Flowtype of the bound compression policy.
description	<String>	Read-only	Description of the policylabel.
__count	<Double>	Read-only	count parameter

Operations

(click to see Properties)

ADD| DELETE| RENAME| GET (ALL)| GET| COUNT

Some options that you can use for each operations:

- **Getting warnings in response:**NITRO allows you to get warnings in an operation by specifying the "warning" query parameter as "yes". For example, to get warnings while connecting to the Citrix ADC appliance, the URL is as follows:

  http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/login?warning=yes

  If any, the warnings are displayed in the response payload with the HTTP code "209 X-NITRO-WARNING".
- **Authenticated access for individual NITRO operations:**NITRO allows you to logon to the Citrix ADC appliance to perform individual operations. You can use this option instead of creating a NITRO session (using the login object) and then using that session to perform all operations,

To do this, you must specify the username and password in the request header of the NITRO request as follows:

X-NITRO-USER:<username>

X-NITRO-PASS:<password>

Note: In such cases, make sure that the request header DOES not include the following:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue>

`*Note:`*

Mandatory parameters are marked in redand placeholder content is marked in <green>.

add

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/cmppolicylabel

HTTP Method:POST

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Content-Type:application/json

Request Payload:

```
{"cmppolicylabel":{

<b>"labelname":<String_value>,

</b><b>"type":<String_value>

</b>}}
```

Response:

HTTP Status Code on Success: 201 Created HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error

delete

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/cmppolicylabel/labelname_value<String>

HTTP Method:DELETE

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue>

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error

rename

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/cmppolicylabel?**action=rename**

HTTP Method:POST

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Content-Type:application/json

Request Payload:

```
{"cmppolicylabel":{

<b>"labelname":<String_value>,

</b><b>"newname":<String_value>

</b>}}
```

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error

get (all)

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/cmppolicylabel

Query-parameters:

attrs

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/cmppolicylabel?**attrs=property-name1,property-name2**

Use this query parameter to specify the resource details that you want to retrieve.

filter

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/cmppolicylabel?**filter=property-name1:property-val1,property-name2:property-val2**

Use this query-parameter to get the filtered set of cmppolicylabel resources configured on Citrix ADC. Filtering can be done on any of the properties of the resource.

view

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/cmppolicylabel?**view=summary**

Note: By default, the retrieved results are displayed in detail view (?view=detail).

pagination

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/cmppolicylabel?**pagesize=#no;pageno=#no**

Use this query-parameter to get the cmppolicylabel resources in chunks.

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "cmppolicylabel": [ {

"labelname":<String_value>,

"type":<String_value>,

"numpol":<Double_value>,

"hits":<Double_value>,

"priority":<Double_value>,

"gotopriorityexpression":<String_value>,

"labeltype":<String_value>,

"invoke_labelname":<String_value>,

"flowtype":<Double_value>,

"description":<String_value>

}]}
```

get

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/cmppolicylabel/labelname_value<String>

Query-parameters:

attrs

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/cmppolicylabel/labelname_value<String>?**attrs=property-name1,property-name2**

Use this query parameter to specify the resource details that you want to retrieve.

view

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/cmppolicylabel/labelname_value<String>?**view=summary**

Note: By default, the retrieved results are displayed in detail view (?view=detail).

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "cmppolicylabel": [ {

"labelname":<String_value>,

"type":<String_value>,

"numpol":<Double_value>,

"hits":<Double_value>,

"priority":<Double_value>,

"gotopriorityexpression":<String_value>,

"labeltype":<String_value>,

"invoke_labelname":<String_value>,

"flowtype":<Double_value>,

"description":<String_value>
```

```
}]}
```

count

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/cmppolicylabel?**count=yes**

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "cmppolicylabel": [ { "__count": "#no"} ] }
```

cmppolicylabel_binding

Binding object which returns the resources bound to cmppolicylabel.

Properties

(click to see Operations)

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
labelname	<String>	Read-write	Name of the HTTP compression policy label for which to display details. Minimum length = 1

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
cmppolicylabel_cmppolicy_binding	<cmppolicylabel_cmppolicy_binding[]>	Read-only	cmppolicy that can be bound to cmppolicylabel.
cmppolicylabel_policybinding_binding	<cmppolicylabel_policybinding_binding[]>	Read-only	policybinding that can be bound to cmppolicylabel.

Operations

(click to see Properties)

GET| GET (ALL)

Some options that you can use for each operations:

- **Getting warnings in response:**NITRO allows you to get warnings in an operation by specifying the "warning" query parameter as "yes". For example, to get warnings while connecting to the Citrix ADC appliance, the URL is as follows:

  http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/login?warning=yes

  If any, the warnings are displayed in the response payload with the HTTP code "209 X-NITRO-WARNING".

- **Authenticated access for individual NITRO operations:**NITRO allows you to logon to the Citrix ADC appliance to perform individual operations. You can use this option instead of creating a NITRO session (using the login object) and then using that session to perform all operations,

  To do this, you must specify the username and password in the request header of the NITRO request as follows:

  X-NITRO-USER:<username>

  X-NITRO-PASS:<password>

  Note: In such cases, make sure that the request header DOES not include the following:

  Cookie:NITRO_AUTH_TOKEN=<tokenvalue>

`*Note:`*

Mandatory parameters are marked in redand placeholder content is marked in <green>.

get

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/cmppolicylabel_binding/labelname_value<String>

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "cmppolicylabel_binding": [ {

"labelname":<String_value>,

"cmppolicylabel_policybinding_binding":<cmppolicylabel_policybinding_binding[]_value>,

"cmppolicylabel_cmppolicy_binding":<cmppolicylabel_cmppolicy_binding[]_value>

}]}
```

get (all)

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/cmppolicylabel_binding

Query-parameters:

bulkbindings

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/cmppolicylabel_binding?**bulkbindings=yes**

NITRO allows you to fetch bindings in bulk.

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "cmppolicylabel_binding": [ {

"labelname":<String_value>,

"cmppolicylabel_policybinding_binding":<cmppolicylabel_policybinding_bindi
ng[]_value>,

"cmppolicylabel_cmppolicy_binding":<cmppolicylabel_cmppolicy_binding[]_val
ue>

}]}
```

cmppolicylabel_cmppolicy_binding

Binding object showing the cmppolicy that can be bound to cmppolicylabel.

Properties

(click to see Operations)

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
priority	<Double>	Read-write	Specifies the priority of the policy.
policyname	<String>	Read-write	The compression policy name.
labelname	<String>	Read-write	Name of the HTTP compression policy label to which to bind the policy. Minimum length = 1
invoke_labelname	<String>	Read-write	Name of the label to invoke if the current policy evaluates to TRUE.
gotopriorityexpression	<String>	Read-write	Expression specifying the priority of the next policy which will get evaluated if the current policy rule evaluates to TRUE.
invoke	<Boolean>	Read-write	Invoke policies bound to a virtual server or a user-defined policy label. After the invoked policies are evaluated, the flow returns to the policy with the next higher priority number in the original label.
labeltype	<String>	Read-write	Type of policy label invocation.

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
			Possible values = reqvserver, resvserver, policylabel
__count	<Double>	Read-write	count parameter

Operations

(click to see Properties)

ADD:| DELETE:| GET| GET (ALL)| COUNT

Some options that you can use for each operations:

- **Getting warnings in response:**NITRO allows you to get warnings in an operation by specifying the "warning" query parameter as "yes". For example, to get warnings while connecting to the Citrix ADC appliance, the URL is as follows:

  http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/login?warning=yes

  If any, the warnings are displayed in the response payload with the HTTP code "209 X-NITRO-WARNING".

- **Authenticated access for individual NITRO operations:**NITRO allows you to logon to the Citrix ADC appliance to perform individual operations. You can use this option instead of creating a NITRO session (using the login object) and then using that session to perform all operations,

  To do this, you must specify the username and password in the request header of the NITRO request as follows:

  X-NITRO-USER:<username>

  X-NITRO-PASS:<password>

  Note: In such cases, make sure that the request header DOES not include the following:

  Cookie:NITRO_AUTH_TOKEN=<tokenvalue>

`*Note:`*

Mandatory parameters are marked in redand placeholder content is marked in <green>.

add

URL:http://<Citrix-ADC-IP-address(NSIP)/nitro/v1/config/cmppolicylabel_cmppolicy_binding

HTTP Method:PUT

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Content-Type:application/json

Request Payload:

```
{

"cmppolicylabel_cmppolicy_binding":{

<b>"labelname":<String_value>,

</b><b>"policyname":<String_value>,

</b><b>"priority":<Double_value>,

</b>"gotopriorityexpression":<String_value>,

"invoke":<Boolean_value>,

"labeltype":<String_value>,

"invoke_labelname":<String_value>

}}
```

Response:

HTTP Status Code on Success: 201 Created HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error

delete

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/cmppolicylabel_cmppolicy_binding/labelname_value<String>

Query-parameters:

args

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/cmppolicylabel_cmppolicy_binding/labelname_value<String>?**args=policyname:<String_value>,priority:<Double_value>**

HTTP Method:DELETE

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue>

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error

get

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/cmppolicylabel_cmppolicy_binding/labelname_value<String>

Query-parameters:

filter

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/cmppolicylabel_cmppolicy_binding/labelname_value<String>?**filter=property-name1:property-value1,property-name2:property-value2**

Use this query-parameter to get the filtered set of cmppolicylabel_cmppolicy_binding resources configured on Citrix ADC. Filtering can be done on any of the properties of the resource.

pagination

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/cmppolicylabel_cmppolicy_binding/labelname_value<String>?**pagesize=#no;pageno=#no**

Use this query-parameter to get the cmppolicylabel_cmppolicy_binding resources in chunks.

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "cmppolicylabel_cmppolicy_binding": [ {

"priority":<Double_value>,

"policyname":<String_value>,

"labelname":<String_value>,

"invoke_labelname":<String_value>,

"gotopriorityexpression":<String_value>,

"invoke":<Boolean_value>,

"labeltype":<String_value>

}]}
```

get (all)

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/cmppolicylabel_cmppolicy_binding

Query-parameters:

bulkbindings

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/cmppolicylabel_cmppolicy_binding?**bulkbindings=yes**

NITRO allows you to fetch bindings in bulk.

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "cmppolicylabel_cmppolicy_binding": [ {

"priority":<Double_value>,

"policyname":<String_value>,

"labelname":<String_value>,

"invoke_labelname":<String_value>,

"gotopriorityexpression":<String_value>,

"invoke":<Boolean_value>,

"labeltype":<String_value>

}]}
```

count

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/cmppolicylabel_cmppolicy_binding/labelname_value<String>?**count=yes**

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{"cmppolicylabel_cmppolicy_binding": [ { "__count": "#no"} ] }
```

cmppolicylabel_policybinding_binding

Binding object showing the policybinding that can be bound to cmppolicylabel.

Properties

(click to see Operations)

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
priority	<Double>	Read-write	Specifies the priority of the policy.
policyname	<String>	Read-write	The compression policy name.
labelname	<String>	Read-write	Name of the HTTP compression policy

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
			label to which to bind the policy. Minimum length = 1
invoke_labelname	<String>	Read-write	Name of the label to invoke if the current policy evaluates to TRUE.
gotopriorityexpression	<String>	Read-write	Expression specifying the priority of the next policy which will get evaluated if the current policy rule evaluates to TRUE.
invoke	<Boolean>	Read-write	Invoke policies bound to a virtual server or a user-defined policy label. After the invoked policies are evaluated, the flow returns to the policy with the next higher priority number in the original label.
labeltype	<String>	Read-write	Type of policy label invocation. Possible values = reqvserver, resvserver, policylabel
__count	<Double>	Read-write	count parameter

Operations

(click to see Properties)

GET| GET (ALL)| COUNT

Some options that you can use for each operations:

- **Getting warnings in response:**NITRO allows you to get warnings in an operation by specifying the "warning" query parameter as "yes". For example, to get warnings while connecting to the Citrix ADC appliance, the URL is as follows:

  http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/login?warning=yes

  If any, the warnings are displayed in the response payload with the HTTP code "209 X-NITRO-WARNING".

- **Authenticated access for individual NITRO operations:**NITRO allows you to logon to the Citrix ADC appliance to perform individual operations. You can use this option instead of creating a NITRO session (using the login object) and then using that session to perform all operations,

  To do this, you must specify the username and password in the request header of the NITRO request as follows:

  X-NITRO-USER:<username>

  X-NITRO-PASS:<password>

  Note: In such cases, make sure that the request header DOES not include the following:

  Cookie:NITRO_AUTH_TOKEN=<tokenvalue>

`*Note:`*

Mandatory parameters are marked in redand placeholder content is marked in <green>.

get

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/cmppolicylabel_policybinding_binding/labelname_value<String>

Query-parameters:

filter

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/cmppolicylabel_policybinding_binding/labelname_value<String>?**filter=property-name1:property-value1,property-name2:property-value2**

Use this query-parameter to get the filtered set of cmppolicylabel_policybinding_binding resources configured on Citrix ADC. Filtering can be done on any of the properties of the resource.

pagination

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/cmppolicylabel_policybinding_binding/labelname_value<String>?**pagesize=#no;pageno=#no**

Use this query-parameter to get the cmppolicylabel_policybinding_binding resources in chunks.

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "cmppolicylabel_policybinding_binding": [ {

"priority":<Double_value>,

"policyname":<String_value>,

"labelname":<String_value>,

"invoke_labelname":<String_value>,

"gotopriorityexpression":<String_value>,

"invoke":<Boolean_value>,

"labeltype":<String_value>
```

```
}]}
```

get (all)

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/cmppolicylabel_policybinding_binding

Query-parameters:

bulkbindings

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/cmppolicylabel_policybinding_binding?**bulkbindings=yes**

NITRO allows you to fetch bindings in bulk.

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "cmppolicylabel_policybinding_binding": [ {

"priority":<Double_value>,

"policyname":<String_value>,

"labelname":<String_value>,

"invoke_labelname":<String_value>,
```

```
"gotopriorityexpression":<String_value>,

"invoke":<Boolean_value>,

"labeltype":<String_value>

}]}
```

count

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/cmppolicylabel_policybinding_binding/labelname_value<String>?**count=yes**

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{"cmppolicylabel_policybinding_binding": [ { "__count": "#no"} ] }
```

cmppolicy_binding

Binding object which returns the resources bound to cmppolicy.

Properties

(click to see Operations)

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
name	<String>	Read-write	Name of the HTTP compression policy for which to display details. Minimum length = 1
cmppolicy_csvserver_binding	<cmppolicy_csvserver_binding[]>	Read-only	csvserver that can be bound to cmppolicy.
cmppolicy_cmppolicylabel_binding	<cmppolicy_cmppolicylabel_binding[]>	Read-only	cmppolicylabel that can be bound to cmppolicy.
cmppolicy_lbvserver_binding	<cmppolicy_lbvserver_binding[]>	Read-only	lbvserver that can be bound to cmppolicy.
cmppolicy_cmpglobal_binding	<cmppolicy_cmpglobal_binding[]>	Read-only	cmpglobal that can be bound to cmppolicy.
cmppolicy_crvserver_binding	<cmppolicy_crvserver_binding[]>	Read-only	crvserver that can be bound to cmppolicy.

Operations

(click to see Properties)

GET| GET (ALL)

Some options that you can use for each operations:

- **Getting warnings in response:**NITRO allows you to get warnings in an operation by specifying the "warning" query parameter as "yes". For example, to get warnings while connecting to the Citrix ADC appliance, the URL is as follows:

  http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/login?warning=yes

  If any, the warnings are displayed in the response payload with the HTTP code "209 X-NITRO-WARNING".

- **Authenticated access for individual NITRO operations:**NITRO allows you to logon to the Citrix ADC appliance to perform individual operations. You can use this option instead of creating a NITRO session (using the login object) and then using that session to perform all operations,

  To do this, you must specify the username and password in the request header of the NITRO request as follows:

  X-NITRO-USER:<username>

  X-NITRO-PASS:<password>

  Note: In such cases, make sure that the request header DOES not include the following:

  Cookie:NITRO_AUTH_TOKEN=<tokenvalue>

`*Note:`*

Mandatory parameters are marked in redand placeholder content is marked in <green>.

get

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/cmppolicy_binding/name_value<String>

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "cmppolicy_binding": [ {

"name":<String_value>,
```

```
"cmppolicy_crvserver_binding":<cmppolicy_crvserver_binding[]_value>,

"cmppolicy_cmpglobal_binding":<cmppolicy_cmpglobal_binding[]_value>,

"cmppolicy_csvserver_binding":<cmppolicy_csvserver_binding[]_value>,

"cmppolicy_cmppolicylabel_binding":<cmppolicy_cmppolicylabel_binding[]_val
ue>,

"cmppolicy_lbvserver_binding":<cmppolicy_lbvserver_binding[]_value>

}]}
```

get (all)

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/cmppolicy_binding

Query-parameters:

bulkbindings

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/cmppolicy_binding?**bulkbindings=yes**

NITRO allows you to fetch bindings in bulk.

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "cmppolicy_binding": [ {

"name":<String_value>,

"cmppolicy_crvserver_binding":<cmppolicy_crvserver_binding[]_value>,

"cmppolicy_cmpglobal_binding":<cmppolicy_cmpglobal_binding[]_value>,

"cmppolicy_csvserver_binding":<cmppolicy_csvserver_binding[]_value>,

"cmppolicy_cmppolicylabel_binding":<cmppolicy_cmppolicylabel_binding[]_val
ue>,

"cmppolicy_lbvserver_binding":<cmppolicy_lbvserver_binding[]_value>

}]}
```

cmppolicy_cmpglobal_binding

Binding object showing the cmpglobal that can be bound to cmppolicy.

Properties

(click to see Operations)

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
boundto	<String>	Read-write	The name of the entity to which the policy is bound.
name	<String>	Read-write	Name of the HTTP compression policy for which to display details. Minimum length = 1
priority	<Double>	Read-only	.

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
labelname	<String>	Read-only	Name of the label to invoke if the current policy rule evaluates to TRUE.
gotopriorityexpression	<String>	Read-only	Expression specifying the priority of the next policy which will get evaluated if the current policy rule evaluates to TRUE.
labeltype	<String>	Read-only	Type of policy label invocation. Possible values = reqvserver, resvserver, policylabel
activepolicy	<Double>	Read-only	.
__count	<Double>	Read-only	count parameter

Operations

(click to see Properties)

GET| GET (ALL)| COUNT

Some options that you can use for each operations:

- **Getting warnings in response:**NITRO allows you to get warnings in an operation by specifying the "warning" query parameter as "yes". For example, to get warnings while connecting to the Citrix ADC appliance, the URL is as follows:

  http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/login?warning=yes

  If any, the warnings are displayed in the response payload with the HTTP code "209 X-NITRO-WARNING".

- **Authenticated access for individual NITRO operations:**NITRO allows you to logon to the Citrix ADC appliance to perform individual operations. You can use this option instead of creating a NITRO session (using the login object) and then using that session to perform all operations,

  To do this, you must specify the username and password in the request header of the NITRO request as follows:

  X-NITRO-USER:<username>

  X-NITRO-PASS:<password>

  Note: In such cases, make sure that the request header DOES not include the following:

  Cookie:NITRO_AUTH_TOKEN=<tokenvalue>

`*Note:`*

Mandatory parameters are marked in redand placeholder content is marked in <green>.

get

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/cmppolicy_cmpglobal_binding/name_value<String>

Query-parameters:

filter

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/cmppolicy_cmpglobal_binding/name_value<String>?**filter=property-name1:property-value1,property-name2:property-value2**

Use this query-parameter to get the filtered set of cmppolicy_cmpglobal_binding resources configured on Citrix ADC. Filtering can be done on any of the properties of the resource.

pagination

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/cmppolicy_cmpglobal_binding/name_value<String>?**pagesize=#no;pageno=#no**

Use this query-parameter to get the cmppolicy_cmpglobal_binding resources in chunks.

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "cmppolicy_cmpglobal_binding": [ {

"boundto":<String_value>,

"name":<String_value>,

"priority":<Double_value>,

"labelname":<String_value>,

"gotopriorityexpression":<String_value>,

"labeltype":<String_value>,

"activepolicy":<Double_value>

}]}
```

get (all)

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/cmppolicy_cmpglobal_binding

Query-parameters:

bulkbindings

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/cmppolicy_cmpglobal_binding?**bulkbindings=yes**

NITRO allows you to fetch bindings in bulk.

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "cmppolicy_cmpglobal_binding": [ {

"boundto":<String_value>,

"name":<String_value>,

"priority":<Double_value>,

"labelname":<String_value>,

"gotopriorityexpression":<String_value>,

"labeltype":<String_value>,

"activepolicy":<Double_value>

}]}
```

count

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/cmppolicy_cmpglobal_binding/name_value<String>?**count=yes**

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{"cmppolicy_cmpglobal_binding": [ { "__count": "#no"} ] }
```

cmppolicy_cmppolicylabel_binding

Binding object showing the cmppolicylabel that can be bound to cmppolicy.

Properties

(click to see Operations)

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
boundto	<String>	Read-write	The name of the entity to which the policy is bound.
name	<String>	Read-write	Name of the HTTP compression policy for which to display details. Minimum length = 1
priority	<Double>	Read-only	.

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
labelname	<String>	Read-only	Name of the label to invoke if the current policy rule evaluates to TRUE.
gotopriorityexpression	<String>	Read-only	Expression specifying the priority of the next policy which will get evaluated if the current policy rule evaluates to TRUE.
labeltype	<String>	Read-only	Type of policy label invocation. Possible values = reqvserver, resvserver, policylabel
activepolicy	<Double>	Read-only	.
__count	<Double>	Read-only	count parameter

Operations

(click to see Properties)

GET| GET (ALL)| COUNT

Some options that you can use for each operations:

- **Getting warnings in response:**NITRO allows you to get warnings in an operation by specifying the "warning" query parameter as "yes". For example, to get warnings while connecting to the Citrix ADC appliance, the URL is as follows:

  http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/login?warning=yes

  If any, the warnings are displayed in the response payload with the HTTP code "209 X-NITRO-WARNING".

- **Authenticated access for individual NITRO operations:**NITRO allows you to logon to the Citrix ADC appliance to perform individual operations. You can use this option instead of creating a NITRO session (using the login object) and then using that session to perform all operations,

  To do this, you must specify the username and password in the request header of the NITRO request as follows:

  X-NITRO-USER:<username>

  X-NITRO-PASS:<password>

  Note: In such cases, make sure that the request header DOES not include the following:

  Cookie:NITRO_AUTH_TOKEN=<tokenvalue>

`*Note:`*

Mandatory parameters are marked in redand placeholder content is marked in <green>.

get

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/cmppolicy_cmppolicylabel_binding/name_value<String>

Query-parameters:

filter

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/cmppolicy_cmppolicylabel_binding/name_value<String>?**filter=property-name1:property-value1,property-name2:property-value2**

Use this query-parameter to get the filtered set of cmppolicy_cmppolicylabel_binding resources configured on Citrix ADC. Filtering can be done on any of the properties of the resource.

pagination

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/cmppolicy_cmppolicylabel_binding/name_value<String>?**pagesize=#no;pageno=#no**

Use this query-parameter to get the cmppolicy_cmppolicylabel_binding resources in chunks.

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "cmppolicy_cmppolicylabel_binding": [ {

"boundto":<String_value>,

"name":<String_value>,

"priority":<Double_value>,

"labelname":<String_value>,

"gotopriorityexpression":<String_value>,

"labeltype":<String_value>,

"activepolicy":<Double_value>

}]}
```

get (all)

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/cmppolicy_cmppolicylabel_binding

Query-parameters:

bulkbindings

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/cmppolicy_cmppolicylabel_binding?**bulkbindings=yes**

NITRO allows you to fetch bindings in bulk.

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "cmppolicy_cmppolicylabel_binding": [ {

"boundto":<String_value>,

"name":<String_value>,

"priority":<Double_value>,

"labelname":<String_value>,

"gotopriorityexpression":<String_value>,

"labeltype":<String_value>,

"activepolicy":<Double_value>

}]}
```

count

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/cmppolicy_cmppolicylabel_binding/name_value<String>?**count=yes**

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{"cmppolicy_cmppolicylabel_binding": [ { "__count": "#no"} ] }
```

cmppolicy_crvserver_binding

Binding object showing the crvserver that can be bound to cmppolicy.

Properties

(click to see Operations)

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
boundto	<String>	Read-write	The name of the entity to which the policy is bound.
name	<String>	Read-write	Name of the HTTP compression policy for which to display details. Minimum length = 1
priority	<Double>	Read-only	.

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
activepolicy	<Double>	Read-only	.
__count	<Double>	Read-only	count parameter

Operations

(click to see Properties)

GET| GET (ALL)| COUNT

Some options that you can use for each operations:

- **Getting warnings in response:**NITRO allows you to get warnings in an operation by specifying the "warning" query parameter as "yes". For example, to get warnings while connecting to the Citrix ADC appliance, the URL is as follows:

  http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/login?warning=yes

  If any, the warnings are displayed in the response payload with the HTTP code "209 X-NITRO-WARNING".
- **Authenticated access for individual NITRO operations:**NITRO allows you to logon to the Citrix ADC appliance to perform individual operations. You can use this option instead of creating a NITRO session (using the login object) and then using that session to perform all operations,

  To do this, you must specify the username and password in the request header of the NITRO request as follows:

  X-NITRO-USER:<username>

  X-NITRO-PASS:<password>

  Note: In such cases, make sure that the request header DOES not include the following:

  Cookie:NITRO_AUTH_TOKEN=<tokenvalue>

`*Note:`*

Mandatory parameters are marked in redand placeholder content is marked in <green>.

get

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/cmppolicy_crvserver_binding/name_value<String>

Query-parameters:

filter

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/cmppolicy_crvserver_binding/name_value<String>?**filter=property-name1:property-value1,property-name2:property-value2**

Use this query-parameter to get the filtered set of cmppolicy_crvserver_binding resources configured on Citrix ADC. Filtering can be done on any of the properties of the resource.

pagination

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/cmppolicy_crvserver_binding/name_value<String>?**pagesize=#no;pageno=#no**

Use this query-parameter to get the cmppolicy_crvserver_binding resources in chunks.

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "cmppolicy_crvserver_binding": [ {

"boundto":<String_value>,

"name":<String_value>,
```

```
"priority":<Double_value>,

"activepolicy":<Double_value>

}]}
```

get (all)

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/cmppolicy_crvserver_binding

Query-parameters:

bulkbindings

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/cmppolicy_crvserver_binding?**bulkbindings=yes**

NITRO allows you to fetch bindings in bulk.

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "cmppolicy_crvserver_binding": [ {

"boundto":<String_value>,

"name":<String_value>,
```

```
"priority":<Double_value>,

"activepolicy":<Double_value>

}]}
```

count

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/cmppolicy_crvserver_binding/name_value<String>?**count=yes**

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{"cmppolicy_crvserver_binding": [ { "__count": "#no"} ] }
```

cmppolicy_csvserver_binding

Binding object showing the csvserver that can be bound to cmppolicy.

Properties

(click to see Operations)

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
boundto	<String>	Read-write	The name of the entity to which the policy is bound.
name	<String>	Read-write	Name of the HTTP compression policy for which to display details. Minimum length = 1
priority	<Double>	Read-only	.
labelname	<String>	Read-only	Name of the label to invoke if the current policy rule evaluates to TRUE.
gotopriorityexpression	<String>	Read-only	Expression specifying the priority of the next policy which will get evaluated if the current policy rule evaluates to TRUE.
labeltype	<String>	Read-only	Type of policy label invocation. Possible values = reqvserver, resvserver, policylabel
activepolicy	<Double>	Read-only	.
__count	<Double>	Read-only	count parameter

Operations

(click to see Properties)

GET| GET (ALL)| COUNT

Some options that you can use for each operations:

- **Getting warnings in response:**NITRO allows you to get warnings in an operation by specifying the "warning" query parameter as "yes". For example, to get warnings while connecting to the Citrix ADC appliance, the URL is as follows:

  http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/login?warning=yes

  If any, the warnings are displayed in the response payload with the HTTP code "209 X-NITRO-WARNING".

- **Authenticated access for individual NITRO operations:**NITRO allows you to logon to the Citrix ADC appliance to perform individual operations. You can use this option instead of creating a NITRO session (using the login object) and then using that session to perform all operations,

  To do this, you must specify the username and password in the request header of the NITRO request as follows:

  X-NITRO-USER:<username>

  X-NITRO-PASS:<password>

  Note: In such cases, make sure that the request header DOES not include the following:

  Cookie:NITRO_AUTH_TOKEN=<tokenvalue>

`*Note:`*

Mandatory parameters are marked in redand placeholder content is marked in <green>.

get

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/cmppolicy_csvserver_binding/name_value<String>

Query-parameters:

filter

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/cmppolicy_csvserver_binding/name_value<String>?**filter=property-name1:property-value1,property-name2:property-value2**

Use this query-parameter to get the filtered set of cmppolicy_csvserver_binding resources configured on Citrix ADC. Filtering can be done on any of the properties of the resource.

pagination

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/cmppolicy_csvserver_binding/name_value<String>?**pagesize=#no;pageno=#no**

Use this query-parameter to get the cmppolicy_csvserver_binding resources in chunks.

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "cmppolicy_csvserver_binding": [ {

"boundto":<String_value>,

"name":<String_value>,

"priority":<Double_value>,

"labelname":<String_value>,

"gotopriorityexpression":<String_value>,

"labeltype":<String_value>,

"activepolicy":<Double_value>

}]}
```

get (all)

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/cmppolicy_csvserver_binding

Query-parameters:

bulkbindings

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/cmppolicy_csvserver_binding?**bulkbindings=yes**

NITRO allows you to fetch bindings in bulk.

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "cmppolicy_csvserver_binding": [ {

"boundto":<String_value>,

"name":<String_value>,

"priority":<Double_value>,

"labelname":<String_value>,

"gotopriorityexpression":<String_value>,

"labeltype":<String_value>,
```

```
"activepolicy":<Double_value>

}]}
```

count

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/cmppolicy_csvserver_binding/name_value<String>?**count=yes**

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{"cmppolicy_csvserver_binding": [ { "__count": "#no"} ] }
```

cmppolicy_lbvserver_binding

Binding object showing the lbvserver that can be bound to cmppolicy.

Properties

(click to see Operations)

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
boundto	<String>	Read-write	The name of the entity to which the policy is bound.
name	<String>	Read-write	Name of the HTTP compression policy for which to display details. Minimum length = 1
priority	<Double>	Read-only	.
labelname	<String>	Read-only	Name of the label to invoke if the current policy rule evaluates to TRUE.
gotopriorityexpression	<String>	Read-only	Expression specifying the priority of the next policy which will get evaluated if the current policy rule evaluates to TRUE.
labeltype	<String>	Read-only	Type of policy label invocation. Possible values = reqvserver, resvserver, policylabel
activepolicy	<Double>	Read-only	.
__count	<Double>	Read-only	count parameter

Operations

(click to see Properties)

GET| GET (ALL)| COUNT

Some options that you can use for each operations:

- **Getting warnings in response:**NITRO allows you to get warnings in an operation by specifying the "warning" query parameter as "yes". For example, to get warnings while connecting to the Citrix ADC appliance, the URL is as follows:

  http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/login?warning=yes

  If any, the warnings are displayed in the response payload with the HTTP code "209 X-NITRO-WARNING".

- **Authenticated access for individual NITRO operations:**NITRO allows you to logon to the Citrix ADC appliance to perform individual operations. You can use this option instead of creating a NITRO session (using the login object) and then using that session to perform all operations,

  To do this, you must specify the username and password in the request header of the NITRO request as follows:

  X-NITRO-USER:<username>

  X-NITRO-PASS:<password>

  Note: In such cases, make sure that the request header DOES not include the following:

  Cookie:NITRO_AUTH_TOKEN=<tokenvalue>

`*Note:`*

Mandatory parameters are marked in redand placeholder content is marked in <green>.

get

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/cmppolicy_lbvserver_binding/name_value<String>

Query-parameters:

filter

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/cmppolicy_lbvserver_binding/name_value<String>?**filter=property-name1:property-value1,property-name2:property-value2**

Use this query-parameter to get the filtered set of cmppolicy_lbvserver_binding resources configured on Citrix ADC. Filtering can be done on any of the properties of the resource.

pagination

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/cmppolicy_lbvserver_binding/name_value<String>?**pagesize=#no;pageno=#no**

Use this query-parameter to get the cmppolicy_lbvserver_binding resources in chunks.

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "cmppolicy_lbvserver_binding": [ {

"boundto":<String_value>,

"name":<String_value>,

"priority":<Double_value>,

"labelname":<String_value>,

"gotopriorityexpression":<String_value>,

"labeltype":<String_value>,

"activepolicy":<Double_value>

}]}
```

get (all)

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/cmppolicy_lbvserver_binding

Query-parameters:

bulkbindings

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/cmppolicy_lbvserver_binding?**bulkbindings=yes**

NITRO allows you to fetch bindings in bulk.

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "cmppolicy_lbvserver_binding": [ {

"boundto":<String_value>,

"name":<String_value>,

"priority":<Double_value>,

"labelname":<String_value>,

"gotopriorityexpression":<String_value>,

"labeltype":<String_value>,
```

```
"activepolicy":<Double_value>

}]}
```

count

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/cmppolicy_lbvserver_binding/name_value<String>?**count=yes**

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{"cmppolicy_lbvserver_binding": [ { "__count": "#no"} ] }
```

Content Switching

Content Switching

Content Switching configuration. Content Switching feature that enables you to direct traffic to servers on the basis of content.

NAME	DESCRIPTION
csaction	Configuration for Content Switching action

NAME	DESCRIPTION
csparameter	Configuration for CS parameter
cspolicy	Configuration for content-switching policy
cspolicy_binding	Binding object showing the resources that can be bound to cspolicy
cspolicy_crvserver_binding	Binding object showing the crvserver that can be bound to cspolicy
cspolicy_cspolicylabel_binding	Binding object showing the cspolicylabel that can be bound to cspolicy
cspolicy_csvserver_binding	Binding object showing the csvserver that can be bound to cspolicy
cspolicylabel	Configuration for CS policy label
cspolicylabel_binding	Binding object showing the resources that can be bound to cspolicylabel
cspolicylabel_cspolicy_binding	Binding object showing the cspolicy that can be bound to cspolicylabel
csvserver	Configuration for CS virtual server
csvserver_appflowpolicy_binding	Binding object showing the appflowpolicy that can be bound to csvserver
csvserver_appfwpolicy_binding	Binding object showing the appfwpolicy that can be bound to csvserver
csvserver_appqoepolicy_binding	Binding object showing the appqoepolicy that can be bound to csvserver
csvserver_auditnslogpolicy_binding	Binding object showing the auditnslogpolicy that can be bound to csvserver
csvserver_auditsyslogpolicy_binding	Binding object showing the auditsyslogpolicy that can be bound to csvserver

NAME	DESCRIPTION
csvserver_authorizationpolicy_binding	Binding object showing the authorizationpolicy that can be bound to csvserver
csvserver_binding	Binding object showing the resources that can be bound to csvserver
csvserver_cachepolicy_binding	Binding object showing the cachepolicy that can be bound to csvserver
csvserver_cmppolicy_binding	Binding object showing the cmppolicy that can be bound to csvserver
csvserver_contentinspectionpolicy_binding	Binding object showing the contentinspectionpolicy that can be bound to csvserver
csvserver_cspolicy_binding	Binding object showing the cspolicy that can be bound to csvserver
csvserver_domain_binding	Binding object showing the domain that can be bound to csvserver
csvserver_feopolicy_binding	Binding object showing the feopolicy that can be bound to csvserver
csvserver_filterpolicy_binding	Binding object showing the filterpolicy that can be bound to csvserver
csvserver_gslbvserver_binding	Binding object showing the gslbvserver that can be bound to csvserver
csvserver_lbvserver_binding	Binding object showing the lbvserver that can be bound to csvserver
csvserver_responderpolicy_binding	Binding object showing the responderpolicy that can be bound to csvserver
csvserver_rewritepolicy_binding	Binding object showing the rewritepolicy that can be bound to csvserver
csvserver_spilloverpolicy_binding	Binding object showing the spilloverpolicy that can be bound to csvserver

NAME	DESCRIPTION
csvserver_tmtrafficpolicy_binding	Binding object showing the tmtrafficpolicy that can be bound to csvserver
csvserver_transformpolicy_binding	Binding object showing the transformpolicy that can be bound to csvserver
csvserver_vpnvserver_binding	Binding object showing the vpnvserver that can be bound to csvserver

csaction

Configuration for Content Switching action resource.

Properties

(click to see Operations)

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
name	<String>	Read-write	Name for the content switching action. Must begin with an ASCII alphanumeric or underscore (_) character, and must contain only ASCII alphanumeric, underscore, hash (#), period (.), space, colon (:), at sign (@), equal sign (=), and hyphen (-) characters. Can be changed after the content switching action is created. The following requirement applies

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
			only to the Citrix ADC CLI: If the name includes one or more spaces, enclose the name in double or single quotation marks (for example, "my action" or 'my action').
targetlbvserver	<String>	Read-write	Name of the load balancing virtual server to which the content is switched.
targetvserver	<String>	Read-write	Name of the VPN virtual server to which the content is switched.
targetvserverexpr	<String>	Read-write	Information about this content switching action. Maximum length = 1499
comment	<String>	Read-write	Comments associated with this cs action.
newname	<String>	Read-write	New name for the content switching action. Must begin with an ASCII alphanumeric or underscore (_) character, and must contain only ASCII

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
			alphanumeric, underscore, hash (#), period (.), space, colon (:), at sign (@), equal sign (=), and hyphen (-) characters. The following requirement applies only to the Citrix ADC CLI: If the name includes one or more spaces, enclose the name in double or single quotation marks (for example, "my name" or 'my name'). Minimum length = 1
hits	<Double>	Read-only	The number of times the action has been taken.
referencecount	<Double>	Read-only	The number of references to the action.
undefhits	<Double>	Read-only	The number of times the action resulted in UNDEF.
__count	<Double>	Read-only	count parameter

Operations

(click to see Properties)

ADD| DELETE| UPDATE| UNSET| RENAME| GET (ALL)| GET| COUNT

Some options that you can use for each operations:

- **Getting warnings in response:**NITRO allows you to get warnings in an operation by specifying the "warning" query parameter as "yes". For example, to get warnings while connecting to the Citrix ADC appliance, the URL is as follows:

  http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/login?warning=yes

  If any, the warnings are displayed in the response payload with the HTTP code "209 X-NITRO-WARNING".

- **Authenticated access for individual NITRO operations:**NITRO allows you to logon to the Citrix ADC appliance to perform individual operations. You can use this option instead of creating a NITRO session (using the login object) and then using that session to perform all operations,

  To do this, you must specify the username and password in the request header of the NITRO request as follows:

  X-NITRO-USER:<username>

  X-NITRO-PASS:<password>

  Note: In such cases, make sure that the request header DOES not include the following:

  Cookie:NITRO_AUTH_TOKEN=<tokenvalue>

`*Note:`*

Mandatory parameters are marked in redand placeholder content is marked in <green>.

add

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/csaction

HTTP Method:POST

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Content-Type:application/json

Request Payload:

```
{"csaction":{

<b>"name":<String_value>,

</b>"targetlbvserver":<String_value>,

"targetvserver":<String_value>,

"targetvserverexpr":<String_value>,

"comment":<String_value>

}}
```

Response:

HTTP Status Code on Success: 201 Created HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error

delete

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/csaction/name_value<String>

HTTP Method:DELETE

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue>

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error

update

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/csaction

HTTP Method:PUT

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Content-Type:application/json

Request Payload:

```
{"csaction":{

<b>"name":<String_value>,

</b>"targetlbvserver":<String_value>,

"targetvserver":<String_value>,

"targetvserverexpr":<String_value>,

"comment":<String_value>

}}
```

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error

unset

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/csaction?**action=unset**

HTTP Method:POST

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Content-Type:application/json

Request Payload:

```
{"csaction":{
```

```
<b>"name":<String_value>,

</b>"comment":true

}}
```

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error

rename

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/csaction?**action=rename**

HTTP Method:POST

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Content-Type:application/json

Request Payload:

```
{"csaction":{

<b>"name":<String_value>,

</b><b>"newname":<String_value>

</b>}}
```

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error

get (all)

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/csaction

Query-parameters:

attrs

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/csaction?**attrs=property-name1,property-name2**

Use this query parameter to specify the resource details that you want to retrieve.

filter

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/csaction?**filter=property-name1:property-val1,property-name2:property-val2**

Use this query-parameter to get the filtered set of csaction resources configured on Citrix ADC. Filtering can be done on any of the properties of the resource.

view

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/csaction?**view=summary**

Note: By default, the retrieved results are displayed in detail view (?view=detail).

pagination

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/csaction?**pagesize=#no;pageno=#no**

Use this query-parameter to get the csaction resources in chunks.

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "csaction": [ {

"name":<String_value>,

"targetlbvserver":<String_value>,

"targetvserver":<String_value>,

"targetvserverexpr":<String_value>,

"hits":<Double_value>,

"referencecount":<Double_value>,

"undefhits":<Double_value>,

"comment":<String_value>

}]}
```

get

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/csaction/name_value<String>

Query-parameters:

attrs

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/csaction/name_value<String>?**attrs=property-name1,property-name2**

Use this query parameter to specify the resource details that you want to retrieve.

view

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/csaction/name_value<String>?**view=summary**

Note: By default, the retrieved results are displayed in detail view (?view=detail).

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "csaction": [ {

"name":<String_value>,

"targetlbvserver":<String_value>,

"targetvserver":<String_value>,

"targetvserverexpr":<String_value>,

"hits":<Double_value>,

"referencecount":<Double_value>,

"undefhits":<Double_value>,

"comment":<String_value>

}]}
```

count

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/csaction?**count=yes**

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "csaction": [ { "__count": "#no"} ] }
```

csparameter

Configuration for CS parameter resource.

Properties

(click to see Operations)

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
stateupdate	<String>	Read-write	Specifies whether the virtual server checks the attached load balancing server for state information. Default value: DISABLED Possible values = ENABLED, DISABLED

Operations

(click to see Properties)

UPDATE| UNSET| GET (ALL)

Some options that you can use for each operations:

- **Getting warnings in response:**NITRO allows you to get warnings in an operation by specifying the "warning" query parameter as "yes". For example, to get warnings while connecting to the Citrix ADC appliance, the URL is as follows:

  http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/login?warning=yes

  If any, the warnings are displayed in the response payload with the HTTP code "209 X-NITRO-WARNING".

- **Authenticated access for individual NITRO operations:**NITRO allows you to logon to the Citrix ADC appliance to perform individual operations. You can use this option instead of creating a NITRO session (using the login object) and then using that session to perform all operations,

  To do this, you must specify the username and password in the request header of the NITRO request as follows:

  X-NITRO-USER:<username>

  X-NITRO-PASS:<password>

  Note: In such cases, make sure that the request header DOES not include the following:

  Cookie:NITRO_AUTH_TOKEN=<tokenvalue>

`*Note:`*

Mandatory parameters are marked in redand placeholder content is marked in <green>.

update

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/csparameter

HTTP Method:PUT

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Content-Type:application/json

Request Payload:

```
{"csparameter":{

"stateupdate":<String_value>

}}
```

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error

unset

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/csparameter?**action=unset**

HTTP Method:POST

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Content-Type:application/json

Request Payload:

```
{"csparameter":{

"stateupdate":true

}}
```

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error

get (all)

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/csparameter

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "csparameter": [ {

"stateupdate":<String_value>

}]}
```

cspolicy

Configuration for content-switching policy resource.

Properties

(click to see Operations)

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
policyname	<String>	Read-write	Name for the content switching policy. Must begin with an ASCII alphanumeric or underscore (_) character, and must

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
			contain only ASCII alphanumeric, underscore, hash (#), period (.), space, colon (:), at sign (@), equal sign (=), and hyphen (-) characters. Cannot be changed after a policy is created. The following requirement applies only to the Citrix ADC CLI: If the name includes one or more spaces, enclose the name in double or single quotation marks (for example, "my policy" or 'my policy'). Minimum length = 1
url	<String>	Read-write	URL string that is matched with the URL of a request. Can contain a wildcard character. Specify the string value in the following format: [[prefix] [*]] [.suffix]. Minimum length = 1 Maximum length = 208

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
rule	<String>	Read-write	Expression, or name of a named expression, against which traffic is evaluated. The following requirements apply only to the Citrix ADC CLI: * If the expression includes one or more spaces, enclose the entire expression in double quotation marks. * If the expression itself includes double quotation marks, escape the quotations by using the character. * Alternatively, you can use single quotation marks to enclose the rule, in which case you do not have to escape the double quotation marks.
domain	<String>	Read-write	The domain name. The string value can range to 63 characters.

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
			Minimum length = 1
action	<String>	Read-write	Content switching action that names the target load balancing virtual server to which the traffic is switched.
logaction	<String>	Read-write	The log action associated with the content switching policy.
newname	<String>	Read-write	The new name of the content switching policy. Minimum length = 1
vstype	<Double>	Read-only	Virtual server type.
hits	<Double>	Read-only	Total number of hits.
bindhits	<Double>	Read-only	Total number of hits.
labelname	<String>	Read-only	Name of the label invoked.
labeltype	<String>	Read-only	The invocation type. Possible values = reqvserver, resvserver, policylabel
priority	<Double>	Read-only	priority of bound policy.
activepolicy	<Boolean>	Read-only	Indicates whether policy is bound or not.

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
cspolicytype	<String>	Read-only	Indicates whether policy is PI or not. (used only during display). Possible values = Classic Policy, Advanced Policy
__count	<Double>	Read-only	count parameter

Operations

(click to see Properties)

ADD| DELETE| UPDATE| UNSET| RENAME| GET (ALL)| GET| COUNT

Some options that you can use for each operations:

- **Getting warnings in response:**NITRO allows you to get warnings in an operation by specifying the "warning" query parameter as "yes". For example, to get warnings while connecting to the Citrix ADC appliance, the URL is as follows:

  http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/login?warning=yes

  If any, the warnings are displayed in the response payload with the HTTP code "209 X-NITRO-WARNING".

- **Authenticated access for individual NITRO operations:**NITRO allows you to logon to the Citrix ADC appliance to perform individual operations. You can use this option instead of creating a NITRO session (using the login object) and then using that session to perform all operations,

  To do this, you must specify the username and password in the request header of the NITRO request as follows:

  X-NITRO-USER:<username>

  X-NITRO-PASS:<password>

  Note: In such cases, make sure that the request header DOES not include the following:

  Cookie:NITRO_AUTH_TOKEN=<tokenvalue>

`*Note:` *

Mandatory parameters are marked in redand placeholder content is marked in <green>.

add

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/cspolicy

HTTP Method:POST

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Content-Type:application/json

Request Payload:

```
{"cspolicy":{

<b>"policyname":<String_value>,

</b>"url":<String_value>,

"rule":<String_value>,

"domain":<String_value>,

"action":<String_value>,

"logaction":<String_value>

}}
```

Response:

HTTP Status Code on Success: 201 Created HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error

delete

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/cspolicy/policyname_value<String>

HTTP Method:DELETE

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue>

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error

update

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/cspolicy

HTTP Method:PUT

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Content-Type:application/json

Request Payload:

```
{"cspolicy":{

<b>"policyname":<String_value>,

</b>"url":<String_value>,

"rule":<String_value>,

"domain":<String_value>,

"action":<String_value>,

"logaction":<String_value>

}}
```

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error

unset

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/cspolicy?**action=unset**

HTTP Method:POST

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Content-Type:application/json

Request Payload:

```
{"cspolicy":{

<b>"policyname":<String_value>,

</b>"logaction":true,

"url":true,

"rule":true,

"domain":true,

"action":true

}}
```

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error

rename

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/cspolicy?**action=rename**

HTTP Method:POST

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Content-Type:application/json

Request Payload:

```
{"cspolicy":{

<b>"policyname":<String_value>,

</b><b>"newname":<String_value>

</b>}}
```

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error

get (all)

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/cspolicy

Query-parameters:

attrs

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/cspolicy?**attrs=property-name1,property-name2**

Use this query parameter to specify the resource details that you want to retrieve.

filter

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/cspolicy?**filter=property-name1:property-val1,property-name2:property-val2**

Use this query-parameter to get the filtered set of cspolicy resources configured on Citrix ADC. Filtering can be done on any of the properties of the resource.

view

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/cspolicy?**view=summary**

Note: By default, the retrieved results are displayed in detail view (?view=detail).

pagination

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/cspolicy?**pagesize=#no;pageno=#no**

Use this query-parameter to get the cspolicy resources in chunks.

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "cspolicy": [ {

"policyname":<String_value>,

"url":<String_value>,

"rule":<String_value>,

"domain":<String_value>,

"action":<String_value>,
```

```
"vstype":<Double_value>,

"hits":<Double_value>,

"bindhits":<Double_value>,

"labelname":<String_value>,

"labeltype":<String_value>,

"target":<String_value>,

"priority":<Double_value>,

"activepolicy":<Boolean_value>,

"cspolicytype":<String_value>,

"logaction":<String_value>

}]}
```

get

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/cspolicy/policyname_value<String>

Query-parameters:

attrs

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/cspolicy/policyname_value<String>?**attrs=property-name1,property-name2**

Use this query parameter to specify the resource details that you want to retrieve.

view

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/cspolicy/policyname_value<String>?**view=summary**

Note: By default, the retrieved results are displayed in detail view (?view=detail).

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "cspolicy": [ {

"policyname":<String_value>,

"url":<String_value>,

"rule":<String_value>,

"domain":<String_value>,

"action":<String_value>,

"vstype":<Double_value>,

"hits":<Double_value>,

"bindhits":<Double_value>,

"labelname":<String_value>,

"labeltype":<String_value>,

"target":<String_value>,

"priority":<Double_value>,

"activepolicy":<Boolean_value>,
```

```
"cspolicytype":<String_value>,

"logaction":<String_value>

}]}
```

count

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/cspolicy?**count=yes**

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "cspolicy": [ { "__count": "#no"} ] }
```

cspolicylabel

Configuration for CS policy label resource.

Properties

(click to see Operations)

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
labelname	<String>	Read-write	Name for the policy label. Must begin with an ASCII alphanumeric or underscore (_) character, and must contain only ASCII alphanumeric, underscore, hash (#), period (.), space, colon (:), at sign (@), equal sign (=), and hyphen (-) characters. The label name must be unique within the list of policy labels for content switching. The following requirement applies only to the Citrix ADC CLI: If the name includes one or more spaces, enclose the name in double or single quotation marks (for example, "my policylabel" or 'my policylabel').
cspolicylabeltype	<String>	Read-write	Protocol supported by the policy label. All policies bound to the policy label must either match the

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
			specified protocol or be a subtype of that protocol. Available settings function as follows: * HTTP - Supports policies that process HTTP traffic. Used to access unencrypted Web sites. (The default.) * SSL - Supports policies that process HTTPS/ SSL encrypted traffic. Used to access encrypted Web sites. * TCP - Supports policies that process any type of TCP traffic, including HTTP. * SSL_TCP - Supports policies that process SSL-encrypted TCP traffic, including SSL. * UDP - Supports policies that process any type of

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
			UDP-based traffic, including DNS. * DNS - Supports policies that process DNS traffic. * ANY - Supports all types of policies except HTTP, SSL, and TCP. * SIP_UDP - Supports policies that process UDP based Session Initiation Protocol (SIP) traffic. SIP initiates, manages, and terminates multimedia communications sessions, and has emerged as the standard for Internet telephony (VoIP). * RTSP - Supports policies that process Real Time Streaming Protocol (RTSP) traffic. RTSP provides delivery of multimedia and other streaming data, such as audio,

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
			video, and other types of streamed media. * RADIUS - Supports policies that process Remote Authentication Dial In User Service (RADIUS) traffic. RADIUS supports combined authentication, authorization, and auditing services for network management. * MYSQL - Supports policies that process MYSQL traffic. * MSSQL - Supports policies that process Microsoft SQL traffic. Possible values = HTTP, TCP, RTSP, SSL, SSL_TCP, UDP, DNS, SIP_UDP, SIP_TCP, ANY, RADIUS, RDP, MYSQL, MSSQL, ORACLE,

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
			DIAMETER, SSL_DIAMETER, FTP, DNS_TCP, SMPP
newname	<String>	Read-write	The new name of the content switching policylabel. Minimum length = 1
numpol	<Double>	Read-only	number of polices bound to label.
hits	<Double>	Read-only	Number of times policy label was invoked.
policyname	<String>	Read-only	Name of the content switching policy.
priority	<Double>	Read-only	Specifies the priority of the policy.
targetvserver	<String>	Read-only	Name of the virtual server to which to forward requests that match the policy.
gotopriorityexpression	<String>	Read-only	Expression specifying the priority of the next policy which will get evaluated if the current policy rule evaluates to TRUE.
labeltype	<String>	Read-only	Type of policy label invocation. Possible values = policylabel

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
invoke_labelname	<String>	Read-only	Name of the label to invoke if the current policy rule evaluates to TRUE.
__count	<Double>	Read-only	count parameter

Operations

(click to see Properties)

ADD| DELETE| RENAME| GET (ALL)| GET| COUNT

Some options that you can use for each operations:

- **Getting warnings in response:**NITRO allows you to get warnings in an operation by specifying the "warning" query parameter as "yes". For example, to get warnings while connecting to the Citrix ADC appliance, the URL is as follows:

  http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/login?warning=yes

  If any, the warnings are displayed in the response payload with the HTTP code "209 X-NITRO-WARNING".

- **Authenticated access for individual NITRO operations:**NITRO allows you to logon to the Citrix ADC appliance to perform individual operations. You can use this option instead of creating a NITRO session (using the login object) and then using that session to perform all operations,

  To do this, you must specify the username and password in the request header of the NITRO request as follows:

  X-NITRO-USER:<username>

  X-NITRO-PASS:<password>

  Note: In such cases, make sure that the request header DOES not include the following:

  Cookie:NITRO_AUTH_TOKEN=<tokenvalue>

`*Note:` *

Mandatory parameters are marked in redand placeholder content is marked in <green>.

add

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/cspolicylabel

HTTP Method:POST

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Content-Type:application/json

Request Payload:

```
{"cspolicylabel":{

<b>"labelname":<String_value>,

</b><b>"cspolicylabeltype":<String_value>

</b>}}
```

Response:

HTTP Status Code on Success: 201 Created HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error

delete

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/cspolicylabel/labelname_value<String>

HTTP Method:DELETE

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue>

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error

rename

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/cspolicylabel?**action=rename**

HTTP Method:POST

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Content-Type:application/json

Request Payload:

```
{"cspolicylabel":{

<b>"labelname":<String_value>,

</b><b>"newname":<String_value>

</b>}}
```

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error

get (all)

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/cspolicylabel

Query-parameters:

attrs

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/cspolicylabel?**attrs=property-name1,property-name2**

Use this query parameter to specify the resource details that you want to retrieve.

filter

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/cspolicylabel?**filter=property-name1:property-val1,property-name2:property-val2**

Use this query-parameter to get the filtered set of cspolicylabel resources configured on Citrix ADC. Filtering can be done on any of the properties of the resource.

view

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/cspolicylabel?**view=summary**

Note: By default, the retrieved results are displayed in detail view (?view=detail).

pagination

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/cspolicylabel?**pagesize=#no;pageno=#no**

Use this query-parameter to get the cspolicylabel resources in chunks.

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "cspolicylabel": [ {

"labelname":<String_value>,

"cspolicylabeltype":<String_value>,

"numpol":<Double_value>,

"hits":<Double_value>,
```

```
"policyname":<String_value>,

"priority":<Double_value>,

"targetvserver":<String_value>,

"gotopriorityexpression":<String_value>,

"labeltype":<String_value>,

"invoke_labelname":<String_value>

}]}
```

get

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/cspolicylabel/labelname_value<String>

Query-parameters:

attrs

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/cspolicylabel/labelname_value<String>?**attrs=property-name1,property-name2**

Use this query parameter to specify the resource details that you want to retrieve.

view

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/cspolicylabel/labelname_value<String>?**view=summary**

Note: By default, the retrieved results are displayed in detail view (?view=detail).

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "cspolicylabel": [ {

"labelname":<String_value>,

"cspolicylabeltype":<String_value>,

"numpol":<Double_value>,

"hits":<Double_value>,

"policyname":<String_value>,

"priority":<Double_value>,

"targetvserver":<String_value>,

"gotopriorityexpression":<String_value>,

"labeltype":<String_value>,

"invoke_labelname":<String_value>

}]}
```

count

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/cspolicylabel?**count=yes**

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "cspolicylabel": [ { "__count": "#no"} ] }
```

cspolicylabel_binding

Binding object which returns the resources bound to cspolicylabel.

Properties

(click to see Operations)

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
labelname	<String>	Read-write	Name of the content switching policy label to display.
cspolicylabel_cspolicy_binding	<cspolicylabel_cspolicy_binding[]>	Read-only	cspolicy that can be bound to cspolicylabel.

Operations

(click to see Properties)

GET| GET (ALL)

Some options that you can use for each operations:

- **Getting warnings in response:**NITRO allows you to get warnings in an operation by specifying the "warning" query parameter as "yes". For example, to get warnings while connecting to the Citrix ADC appliance, the URL is as follows:

  http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/login?warning=yes

  If any, the warnings are displayed in the response payload with the HTTP code "209 X-NITRO-WARNING".

- **Authenticated access for individual NITRO operations:**NITRO allows you to logon to the Citrix ADC appliance to perform individual operations. You can use this option instead of creating a NITRO session (using the login object) and then using that session to perform all operations,

  To do this, you must specify the username and password in the request header of the NITRO request as follows:

  X-NITRO-USER:<username>

  X-NITRO-PASS:<password>

  Note: In such cases, make sure that the request header DOES not include the following:

  Cookie:NITRO_AUTH_TOKEN=<tokenvalue>

`*Note:`*

Mandatory parameters are marked in redand placeholder content is marked in <green>.

get

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/cspolicylabel_binding/labelname_value<String>

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "cspolicylabel_binding": [ {

"labelname":<String_value>,

"cspolicylabel_cspolicy_binding":<cspolicylabel_cspolicy_binding[]_value>

}]}
```

get (all)

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/cspolicylabel_binding

Query-parameters:

bulkbindings

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/cspolicylabel_binding?**bulkbindings=yes**

NITRO allows you to fetch bindings in bulk.

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "cspolicylabel_binding": [ {

"labelname":<String_value>,

"cspolicylabel_cspolicy_binding":<cspolicylabel_cspolicy_binding[]_value>

}]}
```

cspolicylabel_cspolicy_binding

Binding object showing the cspolicy that can be bound to cspolicylabel.

Properties

(click to see Operations)

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
priority	<Double>	Read-write	Specifies the priority of the policy.
policyname	<String>	Read-write	Name of the content switching policy.
labelname	<String>	Read-write	Name of the policy label to which to bind a content switching policy.
targetvserver	<String>	Read-write	Name of the virtual server to which to forward requests that match the policy.
invoke_labelname	<String>	Read-write	Name of the label to invoke if the current policy rule evaluates to TRUE.

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
gotopriorityexpression	<String>	Read-write	Expression specifying the priority of the next policy which will get evaluated if the current policy rule evaluates to TRUE.
invoke	<Boolean>	Read-write	.
labeltype	<String>	Read-write	Type of policy label invocation. Possible values = policylabel
__count	<Double>	Read-write	count parameter

Operations

(click to see Properties)

ADD:| DELETE:| GET| GET (ALL)| COUNT

Some options that you can use for each operations:

- **Getting warnings in response:**NITRO allows you to get warnings in an operation by specifying the "warning" query parameter as "yes". For example, to get warnings while connecting to the Citrix ADC appliance, the URL is as follows:

  http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/login?warning=yes

  If any, the warnings are displayed in the response payload with the HTTP code "209 X-NITRO-WARNING".
- **Authenticated access for individual NITRO operations:**NITRO allows you to logon to the Citrix ADC appliance to perform individual operations. You can use this option instead of creating a NITRO session (using the login object) and then using that session to perform all operations,

  To do this, you must specify the username and password in the request header of the NITRO request as follows:

X-NITRO-USER:<username>

X-NITRO-PASS:<password>

Note: In such cases, make sure that the request header DOES not include the following:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue>

`*Note:`*

Mandatory parameters are marked in redand placeholder content is marked in <green>.

add

URL:http://<Citrix-ADC-IP-address(NSIP)/nitro/v1/config/cspolicylabel_cspolicy_binding

HTTP Method:PUT

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Content-Type:application/json

Request Payload:

```
{

"cspolicylabel_cspolicy_binding":{

<b>"labelname":<String_value>,

</b><b>"policyname":<String_value>,

</b><b>"priority":<Double_value>,

</b>"targetvserver":<String_value>,

"gotopriorityexpression":<String_value>,

"invoke":<Boolean_value>,

"labeltype":<String_value>,

"invoke_labelname":<String_value>
```

```
}}
```

Response:

HTTP Status Code on Success: 201 Created HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error

delete

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/cspolicylabel_cspolicy_binding/labelname_value<String>

Query-parameters:

args

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/cspolicylabel_cspolicy_binding/labelname_value<String>?**args=policyname:<String_value>**

HTTP Method:DELETE

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue>

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error

get

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/cspolicylabel_cspolicy_binding/labelname_value<String>

Query-parameters:

filter

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/cspolicylabel_cspolicy_binding/labelname_value<String>?**filter=property-name1:property-value1,property-name2:property-value2**

Use this query-parameter to get the filtered set of cspolicylabel_cspolicy_binding resources configured on Citrix ADC. Filtering can be done on any of the properties of the resource.

pagination

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/cspolicylabel_cspolicy_binding/labelname_value<String>?**pagesize=#no;pageno=#no**

Use this query-parameter to get the cspolicylabel_cspolicy_binding resources in chunks.

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "cspolicylabel_cspolicy_binding": [ {

"priority":<Double_value>,

"policyname":<String_value>,

"labelname":<String_value>,

"targetvserver":<String_value>,

"invoke_labelname":<String_value>,
```

```
"gotopriorityexpression":<String_value>,

"invoke":<Boolean_value>,

"labeltype":<String_value>

}]}
```

get (all)

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/cspolicylabel_cspolicy_binding

Query-parameters:

bulkbindings

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/cspolicylabel_cspolicy_binding?**bulkbindings=yes**

NITRO allows you to fetch bindings in bulk.

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "cspolicylabel_cspolicy_binding": [ {

"priority":<Double_value>,
```

```
"policyname":<String_value>,

"labelname":<String_value>,

"targetvserver":<String_value>,

"invoke_labelname":<String_value>,

"gotopriorityexpression":<String_value>,

"invoke":<Boolean_value>,

"labeltype":<String_value>

}]}
```

count

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/cspolicylabel_cspolicy_binding/labelname_value<String>?**count=yes**

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{"cspolicylabel_cspolicy_binding": [ { "__count": "#no"} ] }
```

cspolicy_binding

Binding object which returns the resources bound to cspolicy.

Properties

(click to see Operations)

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
policyname	<String>	Read-write	Name of the content switching policy to display. If this parameter is omitted, details of all the policies are displayed. Minimum length = 1
cspolicy_cspolicylabel_binding	<cspolicy_cspolicylabel_binding[]>	Read-only	cspolicylabel that can be bound to cspolicy.
cspolicy_csvserver_binding	<cspolicy_csvserver_binding[]>	Read-only	csvserver that can be bound to cspolicy.
cspolicy_crvserver_binding	<cspolicy_crvserver_binding[]>	Read-only	crvserver that can be bound to cspolicy.

Operations

(click to see Properties)

GET| GET (ALL)

Some options that you can use for each operations:

- **Getting warnings in response:**NITRO allows you to get warnings in an operation by specifying the "warning" query parameter as "yes". For example, to get warnings while connecting to the Citrix ADC appliance, the URL is as follows:

  http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/login?warning=yes

  If any, the warnings are displayed in the response payload with the HTTP code "209 X-NITRO-WARNING".

- **Authenticated access for individual NITRO operations:**NITRO allows you to logon to the Citrix ADC appliance to perform individual operations. You can use this option instead of creating a NITRO session (using the login object) and then using that session to perform all operations,

  To do this, you must specify the username and password in the request header of the NITRO request as follows:

  X-NITRO-USER:<username>

  X-NITRO-PASS:<password>

  Note: In such cases, make sure that the request header DOES not include the following:

  Cookie:NITRO_AUTH_TOKEN=<tokenvalue>

`*Note:`*

Mandatory parameters are marked in redand placeholder content is marked in <green>.

get

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/cspolicy_binding/policyname_value<String>

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "cspolicy_binding": [ {
```

```
"policyname":<String_value>,

"cspolicy_cspolicylabel_binding":<cspolicy_cspolicylabel_binding[]_value>,

"cspolicy_crvserver_binding":<cspolicy_crvserver_binding[]_value>,

"cspolicy_csvserver_binding":<cspolicy_csvserver_binding[]_value>

}]}
```

get (all)

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/cspolicy_binding

Query-parameters:

bulkbindings

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/cspolicy_binding?**bulkbindings=yes**

NITRO allows you to fetch bindings in bulk.

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "cspolicy_binding": [ {

"policyname":<String_value>,
```

```
"cspolicy_cspolicylabel_binding":<cspolicy_cspolicylabel_binding[]_value>,

"cspolicy_crvserver_binding":<cspolicy_crvserver_binding[]_value>,

"cspolicy_csvserver_binding":<cspolicy_csvserver_binding[]_value>

}]}
```

cspolicy_crvserver_binding

Binding object showing the crvserver that can be bound to cspolicy.

Properties

(click to see Operations)

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
policyname	<String>	Read-write	Name of the content switching policy to display. If this parameter is omitted, details of all the policies are displayed. Minimum length = 1
domain	<String>	Read-write	The domain name. The string value can range to 63 characters. Minimum length = 1
priority	<Double>	Read-only	priority of bound policy.
hits	<Double>	Read-only	Total number of hits.
url	<String>	Read-only	URL string that is matched with the URL of a request.

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
			Can contain a wildcard character. Specify the string value in the following format: [[prefix] [*]] [.suffix]. Minimum length = 1 Maximum length = 208
labeltype	<String>	Read-only	The invocation type. Possible values = reqvserver, resvserver, policylabel
labelname	<String>	Read-only	Name of the label invoked.
__count	<Double>	Read-only	count parameter

Operations

(click to see Properties)

GET| GET (ALL)| COUNT

Some options that you can use for each operations:

- **Getting warnings in response:**NITRO allows you to get warnings in an operation by specifying the "warning" query parameter as "yes". For example, to get warnings while connecting to the Citrix ADC appliance, the URL is as follows:

  http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/login?warning=yes

  If any, the warnings are displayed in the response payload with the HTTP code "209 X-NITRO-WARNING".
- **Authenticated access for individual NITRO operations:**NITRO allows you to logon to the Citrix ADC appliance to perform individual operations. You can use this option instead of

creating a NITRO session (using the login object) and then using that session to perform all operations,

To do this, you must specify the username and password in the request header of the NITRO request as follows:

X-NITRO-USER:<username>

X-NITRO-PASS:<password>

Note: In such cases, make sure that the request header DOES not include the following:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue>

`*Note:`*

Mandatory parameters are marked in redand placeholder content is marked in <green>.

get

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/cspolicy_crvserver_binding/policyname_value<String>

Query-parameters:

filter

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/cspolicy_crvserver_binding/policyname_value<String>?**filter=property-name1:property-value1,property-name2:property-value2**

Use this query-parameter to get the filtered set of cspolicy_crvserver_binding resources configured on Citrix ADC. Filtering can be done on any of the properties of the resource.

pagination

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/cspolicy_crvserver_binding/policyname_value<String>?**pagesize=#no;pageno=#no**

Use this query-parameter to get the cspolicy_crvserver_binding resources in chunks.

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "cspolicy_crvserver_binding": [ {

"policyname":<String_value>,

"domain":<String_value>,

"priority":<Double_value>,

"hits":<Double_value>,

"url":<String_value>,

"labeltype":<String_value>,

"labelname":<String_value>

}]}
```

get (all)

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/cspolicy_crvserver_binding

Query-parameters:

bulkbindings

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/cspolicy_crvserver_binding?**bulkbindings=yes**

NITRO allows you to fetch bindings in bulk.

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "cspolicy_crvserver_binding": [ {

"policyname":<String_value>,

"domain":<String_value>,

"priority":<Double_value>,

"hits":<Double_value>,

"url":<String_value>,

"labeltype":<String_value>,

"labelname":<String_value>

}]}
```

count

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/cspolicy_crvserver_binding/policyname_value<String>?**count=yes**

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{"cspolicy_crvserver_binding": [ { "__count": "#no"} ] }
```

cspolicy_cspolicylabel_binding

Binding object showing the cspolicylabel that can be bound to cspolicy.

Properties

(click to see Operations)

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
policyname	<String>	Read-write	Name of the content switching policy to display. If this parameter is omitted, details of all the policies are displayed. Minimum length = 1
domain	<String>	Read-write	The domain name. The string value can range to 63 characters. Minimum length = 1

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
priority	<Double>	Read-only	priority of bound policy.
hits	<Double>	Read-only	Total number of hits.
url	<String>	Read-only	URL string that is matched with the URL of a request. Can contain a wildcard character. Specify the string value in the following format: [[prefix] [*]] [.suffix]. Minimum length = 1 Maximum length = 208
labeltype	<String>	Read-only	The invocation type. Possible values = reqvserver, resvserver, policylabel
labelname	<String>	Read-only	Name of the label invoked.
__count	<Double>	Read-only	count parameter

Operations

(click to see Properties)

GET| GET (ALL)| COUNT

Some options that you can use for each operations:

- **Getting warnings in response:**NITRO allows you to get warnings in an operation by specifying the "warning" query parameter as "yes". For example, to get warnings while connecting to the Citrix ADC appliance, the URL is as follows:

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/login?warning=yes

If any, the warnings are displayed in the response payload with the HTTP code "209 X-NITRO-WARNING".

- **Authenticated access for individual NITRO operations:**NITRO allows you to logon to the Citrix ADC appliance to perform individual operations. You can use this option instead of creating a NITRO session (using the login object) and then using that session to perform all operations,

  To do this, you must specify the username and password in the request header of the NITRO request as follows:

  X-NITRO-USER:<username>

  X-NITRO-PASS:<password>

  Note: In such cases, make sure that the request header DOES not include the following:

  Cookie:NITRO_AUTH_TOKEN=<tokenvalue>

`*Note:`*

Mandatory parameters are marked in redand placeholder content is marked in <green>.

get

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/cspolicy_cspolicylabel_binding/policyname_value<String>

Query-parameters:

filter

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/cspolicy_cspolicylabel_binding/policyname_value<String>?**filter=property-name1:property-value1,property-name2:property-value2**

Use this query-parameter to get the filtered set of cspolicy_cspolicylabel_binding resources configured on Citrix ADC. Filtering can be done on any of the properties of the resource.

pagination

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/cspolicy_cspolicylabel_binding/policyname_value<String>?**pagesize=#no;pageno=#no**

Use this query-parameter to get the cspolicy_cspolicylabel_binding resources in chunks.

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "cspolicy_cspolicylabel_binding": [ {

"policyname":<String_value>,

"domain":<String_value>,

"priority":<Double_value>,

"hits":<Double_value>,

"url":<String_value>,

"labeltype":<String_value>,

"labelname":<String_value>

}]}
```

get (all)

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/cspolicy_cspolicylabel_binding

Query-parameters:

bulkbindings

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/cspolicy_cspolicylabel_binding?**bulkbindings=yes**

NITRO allows you to fetch bindings in bulk.

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "cspolicy_cspolicylabel_binding": [ {

"policyname":<String_value>,

"domain":<String_value>,

"priority":<Double_value>,

"hits":<Double_value>,

"url":<String_value>,

"labeltype":<String_value>,

"labelname":<String_value>

}]}
```

count

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/cspolicy_cspolicylabel_binding/policyname_value<String>?**count=yes**

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{"cspolicy_cspolicylabel_binding": [ { "__count": "#no"} ] }
```

cspolicy_csvserver_binding

Binding object showing the csvserver that can be bound to cspolicy.

Properties

(click to see Operations)

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
policyname	<String>	Read-write	Name of the content switching policy to display. If this parameter is omitted, details of all the policies are displayed. Minimum length = 1

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
domain	<String>	Read-write	The domain name. The string value can range to 63 characters. Minimum length = 1
priority	<Double>	Read-only	priority of bound policy.
hits	<Double>	Read-only	Total number of hits.
url	<String>	Read-only	URL string that is matched with the URL of a request. Can contain a wildcard character. Specify the string value in the following format: [[prefix] [*]] [.suffix]. Minimum length = 1 Maximum length = 208
pipolicyhits	<Double>	Read-only	bind hits for PI CS Policy.
labeltype	<String>	Read-only	The invocation type. Possible values = reqvserver, resvserver, policylabel
labelname	<String>	Read-only	Name of the label invoked.
pihits	<Double>	Read-only	Total number of hits.

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
action	<String>	Read-only	Content switching action that names the target load balancing virtual server to which the traffic is switched.
__count	<Double>	Read-only	count parameter

Operations

(click to see Properties)

GET| GET (ALL)| COUNT

Some options that you can use for each operations:

- **Getting warnings in response:**NITRO allows you to get warnings in an operation by specifying the "warning" query parameter as "yes". For example, to get warnings while connecting to the Citrix ADC appliance, the URL is as follows:

  http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/login?warning=yes

  If any, the warnings are displayed in the response payload with the HTTP code "209 X-NITRO-WARNING".

- **Authenticated access for individual NITRO operations:**NITRO allows you to logon to the Citrix ADC appliance to perform individual operations. You can use this option instead of creating a NITRO session (using the login object) and then using that session to perform all operations,

  To do this, you must specify the username and password in the request header of the NITRO request as follows:

  X-NITRO-USER:<username>

  X-NITRO-PASS:<password>

  Note: In such cases, make sure that the request header DOES not include the following:

  Cookie:NITRO_AUTH_TOKEN=<tokenvalue>

`*Note:` *

Mandatory parameters are marked in redand placeholder content is marked in <green>.

get

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/cspolicy_csvserver_binding/policyname_value<String>

Query-parameters:

filter

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/cspolicy_csvserver_binding/policyname_value<String>?**filter=property-name1:property-value1,property-name2:property-value2**

Use this query-parameter to get the filtered set of cspolicy_csvserver_binding resources configured on Citrix ADC. Filtering can be done on any of the properties of the resource.

pagination

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/cspolicy_csvserver_binding/policyname_value<String>?**pagesize=#no;pageno=#no**

Use this query-parameter to get the cspolicy_csvserver_binding resources in chunks.

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "cspolicy_csvserver_binding": [ {
```

```
"policyname":<String_value>,

"domain":<String_value>,

"priority":<Double_value>,

"hits":<Double_value>,

"url":<String_value>,

"pipolicyhits":<Double_value>,

"labeltype":<String_value>,

"labelname":<String_value>,

"pihits":<Double_value>,

"action":<String_value>

}]}
```

get (all)

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/cspolicy_csvserver_binding

Query-parameters:

bulkbindings

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/cspolicy_csvserver_binding?**bulkbindings=yes**

NITRO allows you to fetch bindings in bulk.

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "cspolicy_csvserver_binding": [ {

"policyname":<String_value>,

"domain":<String_value>,

"priority":<Double_value>,

"hits":<Double_value>,

"url":<String_value>,

"pipolicyhits":<Double_value>,

"labeltype":<String_value>,

"labelname":<String_value>,

"pihits":<Double_value>,

"action":<String_value>

}]}
```

count

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/cspolicy_csvserver_binding/policyname_value<String>?**count=yes**

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{"cspolicy_csvserver_binding": [ { "__count": "#no"} ] }
```

csvserver

Configuration for CS virtual server resource.

Properties

(click to see Operations)

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
name	<String>	Read-write	Name for the content switching virtual server. Must begin with an ASCII alphanumeric or underscore (_) character, and must contain only ASCII alphanumeric, underscore, hash (#), period (.), space, colon (:), at sign (@), equal sign (=), and hyphen (-) characters.

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
			Cannot be changed after the CS virtual server is created. The following requirement applies only to the Citrix ADC CLI: If the name includes one or more spaces, enclose the name in double or single quotation marks (for example, my server or my server). Minimum length = 1
td	<Double>	Read-write	Integer value that uniquely identifies the traffic domain in which you want to configure the entity. If you do not specify an ID, the entity becomes part of the default traffic domain, which has an ID of 0. Minimum value = 0 Maximum value = 4094
servicetype	<String>	Read-write	Protocol used by the virtual server.

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
			Possible values = HTTP, SSL, TCP, FTP, RTSP, SSL_TCP, UDP, DNS, SIP_UDP, SIP_TCP, SIP_SSL, ANY, RADIUS, RDP, MYSQL, MSSQL, DIAMETER, SSL_DIAMETER, DNS_TCP, ORACLE, SMPP, PROXY
ipv46	<String>	Read-write	IP address of the content switching virtual server. Minimum length = 1
targettype	<String>	Read-write	Virtual server target type. Possible values = GSLB
dnsrecordtype	<String>	Read-write	. Default value: NSGSLB_IPV4 Possible values = A, AAAA, CNAME, NAPTR
persistenceid	<Double>	Read-write	. Minimum value = 0 Maximum value = 65535

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
ippattern	<String>	Read-write	IP address pattern, in dotted decimal notation, for identifying packets to be accepted by the virtual server. The IP Mask parameter specifies which part of the destination IP address is matched against the pattern. Mutually exclusive with the IP Address parameter. For example, if the IP pattern assigned to the virtual server is 198.51.100.0 and the IP mask is 255.255.240.0 (a forward mask), the first 20 bits in the destination IP addresses are matched with the first 20 bits in the pattern. The virtual server accepts requests with IP addresses that range from 198.51.96.1 to 198.51.111.254. You can also use a pattern such as 0.0.2.2 and a mask

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
			such as 0.0.255.255 (a reverse mask). If a destination IP address matches more than one IP pattern, the pattern with the longest match is selected, and the associated virtual server processes the request. For example, if the virtual servers, vs1 and vs2, have the same IP pattern, 0.0.100.128, but different IP masks of 0.0.255.255 and 0.0.224.255, a destination IP address of 198.51.100.128 has the longest match with the IP pattern of vs1. If a destination IP address matches two or more virtual servers to the same extent, the request is processed by the virtual server whose port number matches the port

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
			number in the request.
ipmask	<String>	Read-write	IP mask, in dotted decimal notation, for the IP Pattern parameter. Can have leading or trailing non-zero octets (for example, 255.255.240.0 or 0.0.255.255). Accordingly, the mask specifies whether the first n bits or the last n bits of the destination IP address in a client request are to be matched with the corresponding bits in the IP pattern. The former is called a forward mask. The latter is called a reverse mask.
range	<Double>	Read-write	Number of consecutive IP addresses, starting with the address specified by the IP Address parameter, to include in a range of addresses assigned to this virtual server. Default value: 1 Minimum value = 1

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
			Maximum value = 254
port	<Integer>	Read-write	Port number for content switching virtual server. Minimum value = 1 Range 1 - 65535 * in CLI is represented as 65535 in NITRO API
ipset	<String>	Read-write	The list of IPv4/IPv6 addresses bound to ipset would form a part of listening service on the current cs vserver. Minimum length = 1
state	<String>	Read-write	Initial state of the load balancing virtual server. Default value: ENABLED Possible values = ENABLED, DISABLED
stateupdate	<String>	Read-write	Enable state updates for a specific content switching virtual server. By default, the Content Switching virtual server is always UP,

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
			regardless of the state of the Load Balancing virtual servers bound to it. This parameter interacts with the global setting as follows: Global Level | Vserver Level | Result ENABLED ENABLED ENABLED ENABLED DISABLED ENABLED DISABLED ENABLED ENABLED DISABLED DISABLED DISABLED If you want to enable state updates for only some content switching virtual servers, be sure to disable the state update parameter.

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
			Default value: DISABLED Possible values = ENABLED, DISABLED, UPDATEONBACKENDUPDATE
cacheable	<String>	Read-write	Use this option to specify whether a virtual server, used for load balancing or content switching, routes requests to the cache redirection virtual server before sending it to the configured servers. Default value: NO Possible values = YES, NO
redirecturl	<String>	Read-write	URL to which traffic is redirected if the virtual server becomes unavailable. The service type of the virtual server should be either HTTP or SSL. Caution: Make sure that the domain in the URL does not match the domain specified for a content switching

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
			policy. If it does, requests are continuously redirected to the unavailable virtual server. Minimum length = 1
clttimeout	<Double>	Read-write	Idle time, in seconds, after which the client connection is terminated. The default values are: 180 seconds for HTTP/SSL-based services. 9000 seconds for other TCP-based services. 120 seconds for DNS-based services. 120 seconds for other UDP-based services. Minimum value = 0 Maximum value = 31536000
precedence	<String>	Read-write	Type of precedence to use for both RULE-based and URL-based policies on the content

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
			switching virtual server. With the default (RULE) setting, incoming requests are evaluated against the rule-based content switching policies. If none of the rules match, the URL in the request is evaluated against the URL-based content switching policies. Default value: RULE Possible values = RULE, URL
casesensitive	<String>	Read-write	Consider case in URLs (for policies that use URLs instead of RULES). For example, with the ON setting, the URLs /a/1.html and /A/1.HTML are treated differently and can have different targets (set by content switching policies). With the OFF setting, /a/1.html and /A/1.HTML are switched to the same target. Default value: ON

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
			Possible values = ON, OFF
somethod	<String>	Read-write	Type of spillover used to divert traffic to the backup virtual server when the primary virtual server reaches the spillover threshold. Connection spillover is based on the number of connections. Bandwidth spillover is based on the total Kbps of incoming and outgoing traffic. Possible values = CONNECTION, DYNAMICCONNECTION, BANDWIDTH, HEALTH, NONE
sopersistence	<String>	Read-write	Maintain source-IP based persistence on primary and backup virtual servers. Default value: DISABLED Possible values = ENABLED, DISABLED

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
sopersistencetimeout	<Double>	Read-write	Time-out value, in minutes, for spillover persistence. Default value: 2 Minimum value = 2 Maximum value = 1440
sothreshold	<Double>	Read-write	Depending on the spillover method, the maximum number of connections or the maximum total bandwidth (Kbps) that a virtual server can handle before spillover occurs. Minimum value = 1 Maximum value = 4294967287
sobackupaction	<String>	Read-write	Action to be performed if spillover is to take effect, but no backup chain to spillover is usable or exists. Possible values = DROP, ACCEPT, REDIRECT
redirectportrewrite	<String>	Read-write	State of port rewrite while performing HTTP redirect. Default value: DISABLED

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
			Possible values = ENABLED, DISABLED
downstateflush	<String>	Read-write	Flush all active transactions associated with a virtual server whose state transitions from UP to DOWN. Do not enable this option for applications that must complete their transactions. Default value: ENABLED Possible values = ENABLED, DISABLED
backupvserver	<String>	Read-write	Name of the backup virtual server that you are configuring. Must begin with an ASCII alphanumeric or underscore (_) character, and must contain only ASCII alphanumeric, underscore, hash (#), period (.), space, colon (:), at sign (@), equal sign (=), and hyphen (-) characters. Can be changed after the backup virtual server is created. You can assign a different

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
			backup virtual server or rename the existing virtual server. The following requirement applies only to the Citrix ADC CLI: If the name includes one or more spaces, enclose the name in double or single quotation marks. Minimum length = 1
disableprimaryondown	<String>	Read-write	Continue forwarding the traffic to backup virtual server even after the primary server comes UP from the DOWN state. Default value: DISABLED Possible values = ENABLED, DISABLED
insertvserveripport	<String>	Read-write	Insert the virtual server's VIP address and port number in the request header. Available values function as follows:

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
			VIPADDR - Header contains the vserver's IP address and port number without any translation. OFF - The virtual IP and port header insertion option is disabled. V6TOV4MAPPING - Header contains the mapped IPv4 address corresponding to the IPv6 address of the vserver and the port number. An IPv6 address can be mapped to a user-specified IPv4 address using the set ns ip6 command. Possible values = OFF, VIPADDR, V6TOV4MAPPING
vipheader	<String>	Read-write	Name of virtual server IP and port header, for use with the VServer IP Port Insertion parameter. Minimum length = 1

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
rtspnat	<String>	Read-write	Enable network address translation (NAT) for real-time streaming protocol (RTSP) connections. Default value: OFF Possible values = ON, OFF
authenticationhost	<String>	Read-write	FQDN of the authentication virtual server. The service type of the virtual server should be either HTTP or SSL. Minimum length = 3 Maximum length = 252
authentication	<String>	Read-write	Authenticate users who request a connection to the content switching virtual server. Default value: OFF Possible values = ON, OFF
listenpolicy	<String>	Read-write	String specifying the listen policy for the content switching virtual server. Can be either the name of an existing expression or an in-line expression.

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
			Default value: "NONE"
listenpriority	<Double>	Read-write	Integer specifying the priority of the listen policy. A higher number specifies a lower priority. If a request matches the listen policies of more than one virtual server the virtual server whose listen policy has the highest priority (the lowest priority number) accepts the request. Default value: 101 Minimum value = 0 Maximum value = 100
authn401	<String>	Read-write	Enable HTTP 401-response based authentication. Default value: OFF Possible values = ON, OFF
authnvsname	<String>	Read-write	Name of authentication virtual server that authenticates the incoming user requests to this content switching virtual server. .

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
			Minimum length = 1 Maximum length = 252
push	<String>	Read-write	Process traffic with the push virtual server that is bound to this content switching virtual server (specified by the Push VServer parameter). The service type of the push virtual server should be either HTTP or SSL. Default value: DISABLED Possible values = ENABLED, DISABLED
pushvserver	<String>	Read-write	Name of the load balancing virtual server, of type PUSH or SSL_PUSH, to which the server pushes updates received on the client-facing load balancing virtual server. Minimum length = 1
pushlabel	<String>	Read-write	Expression for extracting the label from the response received from server. This string can be

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
			either an existing rule name or an inline expression. The service type of the virtual server should be either HTTP or SSL. Default value: "none"
pushmulticlients	<String>	Read-write	Allow multiple Web 2.0 connections from the same client to connect to the virtual server and expect updates. Default value: NO Possible values = YES, NO
tcpprofilename	<String>	Read-write	Name of the TCP profile containing TCP configuration settings for the virtual server. Minimum length = 1 Maximum length = 127
httpprofilename	<String>	Read-write	Name of the HTTP profile containing HTTP configuration settings for the virtual server. The service type of the virtual server should be either HTTP or SSL.

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
			Minimum length = 1 Maximum length = 127
dbprofilename	<String>	Read-write	Name of the DB profile. Minimum length = 1 Maximum length = 127
oracleserverversion	<String>	Read-write	Oracle server version. Default value: 10G Possible values = 10G, 11G
comment	<String>	Read-write	Information about this virtual server.
mssqlserverversion	<String>	Read-write	The version of the MSSQL server. Default value: 2008R2 Possible values = 70, 2000, 2000SP1, 2005, 2008, 2008R2, 2012, 2014
l2conn	<String>	Read-write	Use L2 Parameters to identify a connection. Possible values = ON, OFF

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
mysqlprotocolversion	<Double>	Read-write	The protocol version returned by the mysql vserver. Default value: 10
mysqlserverversion	<String>	Read-write	The server version string returned by the mysql vserver. Minimum length = 1 Maximum length = 31
mysqlcharacterset	<Double>	Read-write	The character set returned by the mysql vserver. Default value: 8
mysqlservercapabilities	<Double>	Read-write	The server capabilities returned by the mysql vserver. Default value: 41613
appflowlog	<String>	Read-write	Enable logging appflow flow information. Default value: ENABLED Possible values = ENABLED, DISABLED
netprofile	<String>	Read-write	The name of the network profile. Minimum length = 1 Maximum length = 127

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
icmpvsrresponse	<String>	Read-write	Can be active or passive. Default value: PASSIVE Possible values = PASSIVE, ACTIVE
rhistate	<String>	Read-write	A host route is injected according to the setting on the virtual servers * If set to PASSIVE on all the virtual servers that share the IP address, the appliance always injects the hostroute. * If set to ACTIVE on all the virtual servers that share the IP address, the appliance injects even if one virtual server is UP. * If set to ACTIVE on some virtual servers and PASSIVE on the others, the appliance, injects even if one virtual server set to ACTIVE is UP.

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
			Default value: PASSIVE Possible values = PASSIVE, ACTIVE
authnprofile	<String>	Read-write	Name of the authentication profile to be used when authentication is turned on.
dnsprofilename	<String>	Read-write	Name of the DNS profile to be associated with the VServer. DNS profile properties will applied to the transactions processed by a VServer. This parameter is valid only for DNS and DNS-TCP VServers. Minimum length = 1 Maximum length = 127
persistencetype	<String>	Read-write	Type of persistence for the virtual server. Available settings function as follows: * SOURCEIP - Connections from the same client IP address belong to the same

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
			persistence session. * COOKIEINSERT - Connections that have the same HTTP Cookie, inserted by a Set-Cookie directive from a server, belong to the same persistence session. * SSLSESSION - Connections that have the same SSL Session ID belong to the same persistence session. Possible values = SOURCEIP, COOKIEINSERT, SSLSESSION, NONE
persistmask	<String>	Read-write	Persistence mask for IP based persistence types, for IPv4 virtual servers. Minimum length = 1
v6persistmasklen	<Double>	Read-write	Persistence mask for IP based persistence types, for IPv6 virtual servers.

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
			Default value: 128 Minimum value = 1 Maximum value = 128
timeout	<Double>	Read-write	Time period for which a persistence session is in effect. Default value: 2 Minimum value = 0 Maximum value = 1440
cookiename	<String>	Read-write	Use this parameter to specify the cookie name for COOKIE peristence type. It specifies the name of cookie with a maximum of 32 characters. If not specified, cookie name is internally generated.
persistencebackup	<String>	Read-write	Backup persistence type for the virtual server. Becomes operational if the primary persistence mechanism fails. Possible values = SOURCEIP, NONE
backuppersistencetimeout	<Double>	Read-write	Time period for which backup

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
			persistence is in effect. Default value: 2 Minimum value = 2 Maximum value = 1440
domainname	<String>	Read-write	Domain name for which to change the time to live (TTL) and/or backup service IP address. Minimum length = 1
ttl	<Double>	Read-write	. Minimum value = 1
backupip	<String>	Read-write	. Minimum length = 1
cookiedomain	<String>	Read-write	. Minimum length = 1
cookietimeout	<Double>	Read-write	. Minimum value = 0 Maximum value = 1440
sitedomainttl	<Double>	Read-write	. Minimum value = 1
newname	<String>	Read-write	New name for the virtual server. Must begin with an ASCII alphanumeric or underscore (_) character, and must contain only ASCII

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
			alphanumeric, underscore, hash (#), period (.), space, colon (:), at sign (@), equal sign (=), and hyphen (-) characters. The following requirement applies only to the Citrix ADC CLI: If the name includes one or more spaces, enclose the name in double or single quotation marks (for example, "my name" or 'my name'). Minimum length = 1
ip	<String>	Read-only	The IP address of the virtual server.
value	<String>	Read-only	The ssl card status for the transparent ssl cs vserver. Possible values = Certkey not bound, SSL feature disabled
ngname	<String>	Read-only	Nodegroup devno to which this csvserver belongs to.
type	<String>	Read-only	Virtual server type.

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
			Possible values = CONTENT, ADDRESS
curstate	<String>	Read-only	The state of the cs vserver. Possible values = UP, DOWN, UNKNOWN, BUSY, OUT OF SERVICE, GOING OUT OF SERVICE, DOWN WHEN GOING OUT OF SERVICE, NS_EMPTY_STR, Unknown, DISABLED
sc	<String>	Read-only	The state of SureConnect the specified virtual server. Possible values = ON, OFF
status	<Integer>	Read-only	Status.
cachetype	<String>	Read-only	Cache type. Possible values = TRANSPARENT, REVERSE, FORWARD
redirect	<String>	Read-only	Redirect URL string. Possible values = CACHE, POLICY, ORIGIN

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
homepage	<String>	Read-only	Home page.
dnsvservername	<String>	Read-only	DNS vserver name.
domain	<String>	Read-only	Domain.
policyname	<String>	Read-only	Policies bound to this vserver.
servicename	<String>	Read-only	Service name.
weight	<Double>	Read-only	Weight for this service.
cachevserver	<String>	Read-only	Cache vserver name.
targetvserver	<String>	Read-only	target vserver name.
priority	<Double>	Read-only	Priority for the policy.
url	<String>	Read-only	URL string.
gotopriorityexpression	<String>	Read-only	Expression specifying the priority of the next policy which will get evaluated if the current policy rule evaluates to TRUE.
bindpoint	<String>	Read-only	The bindpoint to which the policy is bound. Possible values = REQUEST, RESPONSE, ICA_REQUEST, OTHERTCP_REQUEST

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
invoke	<Boolean>	Read-only	Invoke flag.
labeltype	<String>	Read-only	The invocation type. Possible values = reqvserver, resvserver, policylabel
labelname	<String>	Read-only	Name of the label invoked.
gt2gb	<String>	Read-only	This argument has no effect. Default value: DISABLED Possible values = ENABLED, DISABLED
statechangetimesec	<String>	Read-only	Time when last state change happened. Seconds part.
statechangetimemsec	<Double>	Read-only	Time at which last state change happened. Milliseconds part.
tickssincelaststatechange	<Double>	Read-only	Time in 10 millisecond ticks since the last state change.
ruletype	<Double>	Read-only	Rule type.
lbvserver	<String>	Read-only	Name of the default lb vserver bound. Use this param for Default binding only. For Example: bind cs

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
			vserver cs1 -lbvserver lb1. Minimum length = 1
targetlbvserver	<String>	Read-only	target vserver name.
analyticsprofile	<String>	Read-only	Name of the analytics profile bound to the LB vserver.
nodefaultbindings	<String>	Read-only	to determine if the configuration will have default ssl CIPHER and ECC curve bindings. Default value: NO Possible values = YES, NO
version	<Integer>	Read-only	Cookie version.
__count	<Double>	Read-only	count parameter

Operations

(click to see Properties)

ADD| DELETE| UPDATE| UNSET| ENABLE| DISABLE| RENAME| GET (ALL)| GET| COUNT

Some options that you can use for each operations:

- **Getting warnings in response:**NITRO allows you to get warnings in an operation by specifying the "warning" query parameter as "yes". For example, to get warnings while connecting to the Citrix ADC appliance, the URL is as follows:

  http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/login?warning=yes

  If any, the warnings are displayed in the response payload with the HTTP code "209 X-NITRO-WARNING".

- **Authenticated access for individual NITRO operations:**NITRO allows you to logon to the Citrix ADC appliance to perform individual operations. You can use this option instead of creating a NITRO session (using the login object) and then using that session to perform all operations,

  To do this, you must specify the username and password in the request header of the NITRO request as follows:

  X-NITRO-USER:<username>

  X-NITRO-PASS:<password>

  Note: In such cases, make sure that the request header DOES not include the following:

  Cookie:NITRO_AUTH_TOKEN=<tokenvalue>

`*Note:`*

Mandatory parameters are marked in redand placeholder content is marked in <green>.

add

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/csvserver

HTTP Method:POST

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Content-Type:application/json

Request Payload:

```
{"csvserver":{

<b>"name":<String_value>,

</b>"td":<Double_value>,

<b>"servicetype":<String_value>,

</b>"ipv46":<String_value>,

"targettype":<String_value>,
```

```
"dnsrecordtype":<String_value>,

"persistenceid":<Double_value>,

"ippattern":<String_value>,

"ipmask":<String_value>,

"range":<Double_value>,

"port":<Integer_value>,

"ipset":<String_value>,

"state":<String_value>,

"stateupdate":<String_value>,

"cacheable":<String_value>,

"redirecturl":<String_value>,

"clttimeout":<Double_value>,

"precedence":<String_value>,

"casesensitive":<String_value>,

"somethod":<String_value>,

"sopersistence":<String_value>,

"sopersistencetimeout":<Double_value>,

"sothreshold":<Double_value>,

"sobackupaction":<String_value>,

"redirectportrewrite":<String_value>,

"downstateflush":<String_value>,

"backupvserver":<String_value>,
```

```
"disableprimaryondown":<String_value>,

"insertvserveripport":<String_value>,

"vipheader":<String_value>,

"rtspnat":<String_value>,

"authenticationhost":<String_value>,

"authentication":<String_value>,

"listenpolicy":<String_value>,

"listenpriority":<Double_value>,

"authn401":<String_value>,

"authnvsname":<String_value>,

"push":<String_value>,

"pushvserver":<String_value>,

"pushlabel":<String_value>,

"pushmulticlients":<String_value>,

"tcpprofilename":<String_value>,

"httpprofilename":<String_value>,

"dbprofilename":<String_value>,

"oracleserverversion":<String_value>,

"comment":<String_value>,

"mssqlserverversion":<String_value>,

"l2conn":<String_value>,

"mysqlprotocolversion":<Double_value>,
```

```
"mysqlserverversion":<String_value>,

"mysqlcharacterset":<Double_value>,

"mysqlservercapabilities":<Double_value>,

"appflowlog":<String_value>,

"netprofile":<String_value>,

"icmpvsrresponse":<String_value>,

"rhistate":<String_value>,

"authnprofile":<String_value>,

"dnsprofilename":<String_value>,

"persistencetype":<String_value>,

"persistmask":<String_value>,

"v6persistmasklen":<Double_value>,

"timeout":<Double_value>,

"cookiename":<String_value>,

"persistencebackup":<String_value>,

"backuppersistencetimeout":<Double_value>

}}
```

Response:

HTTP Status Code on Success: 201 Created HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error

delete

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/csvserver/name_value<String>

HTTP Method:DELETE

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue>

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error

update

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/csvserver

HTTP Method:PUT

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Content-Type:application/json

Request Payload:

```
{"csvserver":{

<b>"name":<String_value>,

</b>"ipv46":<String_value>,

"ipset":<String_value>,

"ippattern":<String_value>,

"ipmask":<String_value>,

"stateupdate":<String_value>,

"precedence":<String_value>,
```

```
"casesensitive":<String_value>,

"backupvserver":<String_value>,

"redirecturl":<String_value>,

"cacheable":<String_value>,

"clttimeout":<Double_value>,

"somethod":<String_value>,

"sopersistence":<String_value>,

"sopersistencetimeout":<Double_value>,

"sothreshold":<Double_value>,

"sobackupaction":<String_value>,

"redirectportrewrite":<String_value>,

"downstateflush":<String_value>,

"disableprimaryondown":<String_value>,

"insertvserveripport":<String_value>,

"vipheader":<String_value>,

"rtspnat":<String_value>,

"authenticationhost":<String_value>,

"authentication":<String_value>,

"listenpolicy":<String_value>,

"listenpriority":<Double_value>,

"authn401":<String_value>,

"authnvsname":<String_value>,
```

```
"push":<String_value>,

"pushvserver":<String_value>,

"pushlabel":<String_value>,

"pushmulticlients":<String_value>,

"tcpprofilename":<String_value>,

"httpprofilename":<String_value>,

"dbprofilename":<String_value>,

"comment":<String_value>,

"l2conn":<String_value>,

"mssqlserverversion":<String_value>,

"mysqlprotocolversion":<Double_value>,

"oracleserverversion":<String_value>,

"mysqlserverversion":<String_value>,

"mysqlcharacterset":<Double_value>,

"mysqlservercapabilities":<Double_value>,

"appflowlog":<String_value>,

"netprofile":<String_value>,

"authnprofile":<String_value>,

"icmpvsrresponse":<String_value>,

"rhistate":<String_value>,

"dnsprofilename":<String_value>,

"dnsrecordtype":<String_value>,
```

```
"persistenceid":<Double_value>,

"domainname":<String_value>,

"ttl":<Double_value>,

"backupip":<String_value>,

"cookiedomain":<String_value>,

"cookietimeout":<Double_value>,

"sitedomainttl":<Double_value>,

"persistencetype":<String_value>,

"persistmask":<String_value>,

"v6persistmasklen":<Double_value>,

"timeout":<Double_value>,

"cookiename":<String_value>,

"persistencebackup":<String_value>,

"backuppersistencetimeout":<Double_value>

}}
```

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error

unset

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/csvserver?**action=unset**

HTTP Method:POST

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Content-Type:application/json

Request Payload:

```
{"csvserver":{

<b>"name":<String_value>,

</b>"casesensitive":true,

"backupvserver":true,

"clttimeout":true,

"redirecturl":true,

"authn401":true,

"authentication":true,

"authenticationhost":true,

"authnvsname":true,

"pushvserver":true,

"pushlabel":true,

"tcpprofilename":true,

"httpprofilename":true,

"dbprofilename":true,

"l2conn":true,

"mysqlprotocolversion":true,

"mysqlserverversion":true,
```

```
"mysqlcharacterset":true,

"mysqlservercapabilities":true,

"appflowlog":true,

"netprofile":true,

"icmpvsrresponse":true,

"authnprofile":true,

"sothreshold":true,

"dnsprofilename":true,

"ipset":true,

"stateupdate":true,

"precedence":true,

"cacheable":true,

"somethod":true,

"sopersistence":true,

"sopersistencetimeout":true,

"sobackupaction":true,

"redirectportrewrite":true,

"downstateflush":true,

"disableprimaryondown":true,

"insertvserveripport":true,

"vipheader":true,

"rtspnat":true,
```

```
"listenpolicy":true,

"listenpriority":true,

"push":true,

"pushmulticlients":true,

"comment":true,

"mssqlserverversion":true,

"oracleserverversion":true,

"rhistate":true,

"dnsrecordtype":true,

"persistenceid":true,

"persistencetype":true,

"persistmask":true,

"v6persistmasklen":true,

"timeout":true,

"cookiename":true,

"persistencebackup":true,

"backuppersistencetimeout":true

}}
```


Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error

enable

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/csvserver?**action=enable**

HTTP Method:POST

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Content-Type:application/json

Request Payload:

```
{"csvserver":{

<b>"name":<String_value>

</b>}}
```

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error

disable

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/csvserver?**action=disable**

HTTP Method:POST

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Content-Type:application/json

Request Payload:

```
{"csvserver":{

<b>"name":<String_value>
```

```
</b>}}
```

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error

rename

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/csvserver?**action=rename**

HTTP Method:POST

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Content-Type:application/json

Request Payload:

```
{"csvserver":{

<b>"name":<String_value>,

</b><b>"newname":<String_value>

</b>}}
```

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error

get (all)

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/csvserver

Query-parameters:

attrs

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/csvserver?**attrs=property-name1,property-name2**

Use this query parameter to specify the resource details that you want to retrieve.

filter

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/csvserver?**filter=property-name1:property-val1,property-name2:property-val2**

Use this query-parameter to get the filtered set of csvserver resources configured on Citrix ADC. Filtering can be done on any of the properties of the resource.

view

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/csvserver?**view=summary**

Note: By default, the retrieved results are displayed in detail view (?view=detail).

pagination

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/csvserver?**pagesize=#no;pageno=#no**

Use this query-parameter to get the csvserver resources in chunks.

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "csvserver": [ {
"name":<String_value>,
"insertvserveripport":<String_value>,
"vipheader":<String_value>,
"ip":<String_value>,
"td":<Double_value>,
"ipv46":<String_value>,
"ippattern":<String_value>,
"ipmask":<String_value>,
"value":<String_value>,
"port":<Integer_value>,
"ipset":<String_value>,
"range":<Double_value>,
"servicetype":<String_value>,
"ngname":<String_value>,
"type":<String_value>,
"curstate":<String_value>,
"sc":<String_value>,
"stateupdate":<String_value>,
"status":<Integer_value>,
"cachetype":<String_value>,
```

```
"redirect":<String_value>,

"precedence":<String_value>,

"redirecturl":<String_value>,

"authentication":<String_value>,

"authn401":<String_value>,

"authnvsname":<String_value>,

"casesensitive":<String_value>,

"homepage":<String_value>,

"dnsvservername":<String_value>,

"domain":<String_value>,

"policyname":<String_value>,

"servicename":<String_value>,

"weight":<Double_value>,

"cachevserver":<String_value>,

"targetvserver":<String_value>,

"backupvserver":<String_value>,

"priority":<Double_value>,

"clttimeout":<Double_value>,

"listenpolicy":<String_value>,

"listenpriority":<Double_value>,

"somethod":<String_value>,

"sopersistence":<String_value>,
```

```
"sopersistencetimeout":<Double_value>,

"sothreshold":<Double_value>,

"sobackupaction":<String_value>,

"cacheable":<String_value>,

"url":<String_value>,

"gotopriorityexpression":<String_value>,

"redirectportrewrite":<String_value>,

"downstateflush":<String_value>,

"disableprimaryondown":<String_value>,

"bindpoint":<String_value>,

"invoke":<Boolean_value>,

"labeltype":<String_value>,

"labelname":<String_value>,

"gt2gb":<String_value>,

"statechangetimesec":<String_value>,

"statechangetimemsec":<Double_value>,

"tickssincelaststatechange":<Double_value>,

"rtspnat":<String_value>,

"authenticationhost":<String_value>,

"push":<String_value>,

"pushvserver":<String_value>,

"pushlabel":<String_value>,
```

```
"pushmulticlients":<String_value>,

"tcpprofilename":<String_value>,

"httpprofilename":<String_value>,

"dbprofilename":<String_value>,

"comment":<String_value>,

"oracleserverversion":<String_value>,

"mssqlserverversion":<String_value>,

"l2conn":<String_value>,

"ruletype":<Double_value>,

"mysqlprotocolversion":<Double_value>,

"mysqlserverversion":<String_value>,

"mysqlcharacterset":<Double_value>,

"mysqlservercapabilities":<Double_value>,

"appflowlog":<String_value>,

"netprofile":<String_value>,

"icmpvsrresponse":<String_value>,

"rhistate":<String_value>,

"lbvserver":<String_value>,

"targetlbvserver":<String_value>,

"authnprofile":<String_value>,

"dnsprofilename":<String_value>,

"targettype":<String_value>,
```

```
"domainname":<String_value>,

"ttl":<Double_value>,

"backupip":<String_value>,

"cookiedomain":<String_value>,

"cookietimeout":<Double_value>,

"sitedomainttl":<Double_value>,

"dnsrecordtype":<String_value>,

"persistenceid":<Double_value>,

"analyticsprofile":<String_value>,

"nodefaultbindings":<String_value>,

"persistencetype":<String_value>,

"persistmask":<String_value>,

"v6persistmasklen":<Double_value>,

"timeout":<Double_value>,

"cookiename":<String_value>,

"version":<Integer_value>,

"persistencebackup":<String_value>,

"backuppersistencetimeout":<Double_value>

}]}
```

get

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/csvserver/name_value<String>

Query-parameters:

attrs

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/csvserver/name_value<String>?**attrs=property-name1,property-name2**

Use this query parameter to specify the resource details that you want to retrieve.

view

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/csvserver/name_value<String>?**view=summary**

Note: By default, the retrieved results are displayed in detail view (?view=detail).

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "csvserver": [ {

"name":<String_value>,

"insertvserveripport":<String_value>,

"vipheader":<String_value>,

"ip":<String_value>,

"td":<Double_value>,
```

```
"ipv46":<String_value>,

"ippattern":<String_value>,

"ipmask":<String_value>,

"value":<String_value>,

"port":<Integer_value>,

"ipset":<String_value>,

"range":<Double_value>,

"servicetype":<String_value>,

"ngname":<String_value>,

"type":<String_value>,

"curstate":<String_value>,

"sc":<String_value>,

"stateupdate":<String_value>,

"status":<Integer_value>,

"cachetype":<String_value>,

"redirect":<String_value>,

"precedence":<String_value>,

"redirecturl":<String_value>,

"authentication":<String_value>,

"authn401":<String_value>,

"authnvsname":<String_value>,

"casesensitive":<String_value>,
```

```
"homepage":<String_value>,

"dnsvservername":<String_value>,

"domain":<String_value>,

"policyname":<String_value>,

"servicename":<String_value>,

"weight":<Double_value>,

"cachevserver":<String_value>,

"targetvserver":<String_value>,

"backupvserver":<String_value>,

"priority":<Double_value>,

"clttimeout":<Double_value>,

"listenpolicy":<String_value>,

"listenpriority":<Double_value>,

"somethod":<String_value>,

"sopersistence":<String_value>,

"sopersistencetimeout":<Double_value>,

"sothreshold":<Double_value>,

"sobackupaction":<String_value>,

"cacheable":<String_value>,

"url":<String_value>,

"gotopriorityexpression":<String_value>,

"redirectportrewrite":<String_value>,
```

```
"downstateflush":<String_value>,

"disableprimaryondown":<String_value>,

"bindpoint":<String_value>,

"invoke":<Boolean_value>,

"labeltype":<String_value>,

"labelname":<String_value>,

"gt2gb":<String_value>,

"statechangetimesec":<String_value>,

"statechangetimemsec":<Double_value>,

"tickssincelaststatechange":<Double_value>,

"rtspnat":<String_value>,

"authenticationhost":<String_value>,

"push":<String_value>,

"pushvserver":<String_value>,

"pushlabel":<String_value>,

"pushmulticlients":<String_value>,

"tcpprofilename":<String_value>,

"httpprofilename":<String_value>,

"dbprofilename":<String_value>,

"comment":<String_value>,

"oracleserverversion":<String_value>,

"mssqlserverversion":<String_value>,
```

```
"l2conn":<String_value>,

"ruletype":<Double_value>,

"mysqlprotocolversion":<Double_value>,

"mysqlserverversion":<String_value>,

"mysqlcharacterset":<Double_value>,

"mysqlservercapabilities":<Double_value>,

"appflowlog":<String_value>,

"netprofile":<String_value>,

"icmpvsrresponse":<String_value>,

"rhistate":<String_value>,

"lbvserver":<String_value>,

"targetlbvserver":<String_value>,

"authnprofile":<String_value>,

"dnsprofilename":<String_value>,

"targettype":<String_value>,

"domainname":<String_value>,

"ttl":<Double_value>,

"backupip":<String_value>,

"cookiedomain":<String_value>,

"cookietimeout":<Double_value>,

"sitedomainttl":<Double_value>,

"dnsrecordtype":<String_value>,
```

```
"persistenceid":<Double_value>,

"analyticsprofile":<String_value>,

"nodefaultbindings":<String_value>,

"persistencetype":<String_value>,

"persistmask":<String_value>,

"v6persistmasklen":<Double_value>,

"timeout":<Double_value>,

"cookiename":<String_value>,

"version":<Integer_value>,

"persistencebackup":<String_value>,

"backuppersistencetimeout":<Double_value>

}]}
```

count

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/csvserver?**count=yes**

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "csvserver": [ { "__count": "#no"} ] }
```

csvserver_appflowpolicy_binding

Binding object showing the appflowpolicy that can be bound to csvserver.

Properties

(click to see Operations)

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
priority	<Double>	Read-write	Priority for the policy.
bindpoint	<String>	Read-write	For a rewrite policy, the bind point to which to bind the policy. Note: This parameter applies only to rewrite policies, because content switching policies are evaluated only at request time. Possible values = REQUEST, RESPONSE, ICA_REQUEST, OTHERTCP_REQUEST
policyname	<String>	Read-write	Policies bound to this vserver.
labelname	<String>	Read-write	Name of the label invoked.

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
name	<String>	Read-write	Name of the content switching virtual server to which the content switching policy applies. Minimum length = 1
targetlbvserver	<String>	Read-write	Name of the Load Balancing virtual server to which the content is switched, if policy rule is evaluated to be TRUE. Example: bind cs vs cs1 -policyname pol1 -priority 101 -targetLBVserver lb1 Note: Use this parameter only in case of Content Switching policy bind operations to a CS vserver. Minimum length = 1
gotopriorityexpression	<String>	Read-write	Expression specifying the priority of the next policy which will get evaluated if the current policy rule evaluates to TRUE.
invoke	<Boolean>	Read-write	Invoke flag.
labeltype	<String>	Read-write	The invocation type. Possible values = reqvserver,

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
			resvserver, policylabel
__count	<Double>	Read-write	count parameter

Operations

(click to see Properties)

ADD:| DELETE:| GET| GET (ALL)| COUNT

Some options that you can use for each operations:

- **Getting warnings in response:**NITRO allows you to get warnings in an operation by specifying the "warning" query parameter as "yes". For example, to get warnings while connecting to the Citrix ADC appliance, the URL is as follows:

  http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/login?warning=yes

  If any, the warnings are displayed in the response payload with the HTTP code "209 X-NITRO-WARNING".

- **Authenticated access for individual NITRO operations:**NITRO allows you to logon to the Citrix ADC appliance to perform individual operations. You can use this option instead of creating a NITRO session (using the login object) and then using that session to perform all operations,

  To do this, you must specify the username and password in the request header of the NITRO request as follows:

  X-NITRO-USER:<username>

  X-NITRO-PASS:<password>

  Note: In such cases, make sure that the request header DOES not include the following:

  Cookie:NITRO_AUTH_TOKEN=<tokenvalue>

`*Note:`*

Mandatory parameters are marked in redand placeholder content is marked in <green>.

add

URL:http://<Citrix-ADC-IP-address(NSIP)/nitro/v1/config/csvserver_appflowpolicy_binding

HTTP Method:PUT

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Content-Type:application/json

Request Payload:

```
{

"csvserver_appflowpolicy_binding":{

<b>"name":<String_value>,

</b>"policyname":<String_value>,

"targetlbvserver":<String_value>,

"priority":<Double_value>,

"gotopriorityexpression":<String_value>,

"bindpoint":<String_value>,

"invoke":<Boolean_value>,

"labeltype":<String_value>,

"labelname":<String_value>

}}
```

Response:

HTTP Status Code on Success: 201 Created HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error

delete

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/csvserver_appflowpolicy_binding/name_value<String>

Query-parameters:

args

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/csvserver_appflowpolicy_binding/name_value<String>?**args=policyname:<String_value>,bindpoint:<String_value>,priority:<Double_value>**

HTTP Method:DELETE

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue>

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error

get

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/csvserver_appflowpolicy_binding/name_value<String>

Query-parameters:

filter

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/csvserver_appflowpolicy_binding/name_value<String>?**filter=property-name1:property-value1,property-name2:property-value2**

Use this query-parameter to get the filtered set of csvserver_appflowpolicy_binding resources configured on Citrix ADC. Filtering can be done on any of the properties of the resource.

pagination

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/csvserver_appflowpolicy_binding/name_value<String>?**pagesize=#no;pageno=#no**

Use this query-parameter to get the csvserver_appflowpolicy_binding resources in chunks.

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "csvserver_appflowpolicy_binding": [ {

"priority":<Double_value>,

"bindpoint":<String_value>,

"policyname":<String_value>,

"labelname":<String_value>,

"name":<String_value>,

"targetlbvserver":<String_value>,

"gotopriorityexpression":<String_value>,

"invoke":<Boolean_value>,

"labeltype":<String_value>

}]}
```

get (all)

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/csvserver_appflowpolicy_binding

Query-parameters:

bulkbindings

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/csvserver_appflowpolicy_binding?**bulkbindings=yes**

NITRO allows you to fetch bindings in bulk.

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "csvserver_appflowpolicy_binding": [ {

"priority":<Double_value>,

"bindpoint":<String_value>,

"policyname":<String_value>,

"labelname":<String_value>,

"name":<String_value>,

"targetlbvserver":<String_value>,

"gotopriorityexpression":<String_value>,

"invoke":<Boolean_value>,
```

```
"labeltype":<String_value>

}]}
```

count

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/csvserver_appflowpolicy_binding/name_value<String>?**count=yes**

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{"csvserver_appflowpolicy_binding": [ { "__count": "#no"} ] }
```

csvserver_appfwpolicy_binding

Binding object showing the appfwpolicy that can be bound to csvserver.

Properties

(click to see Operations)

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
priority	<Double>	Read-write	Priority for the policy.

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
bindpoint	<String>	Read-write	The bindpoint to which the policy is bound. Possible values = REQUEST, RESPONSE, ICA_REQUEST, OTHERTCP_REQUEST
policyname	<String>	Read-write	Policies bound to this vserver.
labelname	<String>	Read-write	Name of the label invoked.
name	<String>	Read-write	Name of the content switching virtual server to which the content switching policy applies. Minimum length = 1
targetlbvserver	<String>	Read-write	Name of the Load Balancing virtual server to which the content is switched, if policy rule is evaluated to be TRUE. Example: bind cs vs cs1 -policyname pol1 -priority 101 -targetLBVserver lb1 Note: Use this parameter only in case of Content Switching policy bind operations to a CS vserver.

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
			Minimum length = 1
gotopriorityexpression	<String>	Read-write	Expression specifying the priority of the next policy which will get evaluated if the current policy rule evaluates to TRUE.
invoke	<Boolean>	Read-write	Invoke flag.
labeltype	<String>	Read-write	The invocation type. Possible values = reqvserver, resvserver, policylabel
sc	<String>	Read-only	The state of SureConnect the specified virtual server. Possible values = ON, OFF
__count	<Double>	Read-only	count parameter

Operations

(click to see Properties)

ADD:| DELETE:| GET| GET (ALL)| COUNT

Some options that you can use for each operations:

- **Getting warnings in response:**NITRO allows you to get warnings in an operation by specifying the "warning" query parameter as "yes". For example, to get warnings while connecting to the Citrix ADC appliance, the URL is as follows:

  http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/login?warning=yes

If any, the warnings are displayed in the response payload with the HTTP code "209 X-NITRO-WARNING".

- **Authenticated access for individual NITRO operations:**NITRO allows you to logon to the Citrix ADC appliance to perform individual operations. You can use this option instead of creating a NITRO session (using the login object) and then using that session to perform all operations,

  To do this, you must specify the username and password in the request header of the NITRO request as follows:

  X-NITRO-USER:<username>

  X-NITRO-PASS:<password>

  Note: In such cases, make sure that the request header DOES not include the following:

  Cookie:NITRO_AUTH_TOKEN=<tokenvalue>

`*Note:`*

Mandatory parameters are marked in redand placeholder content is marked in <green>.

add

URL:http://<Citrix-ADC-IP-address(NSIP)/nitro/v1/config/csvserver_appfwpolicy_binding

HTTP Method:PUT

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Content-Type:application/json

Request Payload:

```
{

"csvserver_appfwpolicy_binding":{

<b>"name":<String_value>,

</b>"policyname":<String_value>,
```

```
"targetlbvserver":<String_value>,

"priority":<Double_value>,

"gotopriorityexpression":<String_value>,

"bindpoint":<String_value>,

"invoke":<Boolean_value>,

"labeltype":<String_value>,

"labelname":<String_value>

}}
```

Response:

HTTP Status Code on Success: 201 Created HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error

delete

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/csvserver_appfwpolicy_binding/name_value<String>

Query-parameters:

args

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/csvserver_appfwpolicy_binding/name_value<String>?**args=policyname:<String_value>,bindpoint:<String_value>,priority:<Double_value>**

HTTP Method:DELETE

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue>

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error

get

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/csvserver_appfwpolicy_binding/name_value<String>

Query-parameters:

filter

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/csvserver_appfwpolicy_binding/name_value<String>?**filter=property-name1:property-value1,property-name2:property-value2**

Use this query-parameter to get the filtered set of csvserver_appfwpolicy_binding resources configured on Citrix ADC. Filtering can be done on any of the properties of the resource.

pagination

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/csvserver_appfwpolicy_binding/name_value<String>?**pagesize=#no;pageno=#no**

Use this query-parameter to get the csvserver_appfwpolicy_binding resources in chunks.

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "csvserver_appfwpolicy_binding": [ {
```

```
"priority":<Double_value>,

"bindpoint":<String_value>,

"policyname":<String_value>,

"labelname":<String_value>,

"name":<String_value>,

"targetlbvserver":<String_value>,

"gotopriorityexpression":<String_value>,

"invoke":<Boolean_value>,

"labeltype":<String_value>,

"sc":<String_value>

}]}
```

get (all)

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/csvserver_appfwpolicy_binding

Query-parameters:

bulkbindings

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/csvserver_appfwpolicy_binding?**bulkbindings=yes**

NITRO allows you to fetch bindings in bulk.

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "csvserver_appfwpolicy_binding": [ {

"priority":<Double_value>,

"bindpoint":<String_value>,

"policyname":<String_value>,

"labelname":<String_value>,

"name":<String_value>,

"targetlbvserver":<String_value>,

"gotopriorityexpression":<String_value>,

"invoke":<Boolean_value>,

"labeltype":<String_value>,

"sc":<String_value>

}]}
```

count

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/csvserver_appfwpolicy_binding/name_value<String>?**count=yes**

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{"csvserver_appfwpolicy_binding": [ { "__count": "#no"} ] }
```

csvserver_appqoepolicy_binding

Binding object showing the appqoepolicy that can be bound to csvserver.

Properties

(click to see Operations)

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
priority	<Double>	Read-write	Priority for the policy.
bindpoint	<String>	Read-write	The bindpoint to which the policy is bound. Possible values = REQUEST, RESPONSE, ICA_REQUEST, OTHERTCP_REQUEST
policyname	<String>	Read-write	Policies bound to this vserver.

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
labelname	<String>	Read-write	Name of the label invoked.
name	<String>	Read-write	Name of the content switching virtual server to which the content switching policy applies. Minimum length = 1
targetlbvserver	<String>	Read-write	Name of the Load Balancing virtual server to which the content is switched, if policy rule is evaluated to be TRUE. Example: bind cs vs cs1 -policyname pol1 -priority 101 -targetLBVserver lb1 Note: Use this parameter only in case of Content Switching policy bind operations to a CS vserver. Minimum length = 1
gotopriorityexpression	<String>	Read-write	Expression specifying the priority of the next policy which will get evaluated if the current policy rule evaluates to TRUE.
invoke	<Boolean>	Read-write	Invoke flag.
labeltype	<String>	Read-write	The invocation type.

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
			Possible values = reqvserver, resvserver, policylabel
__count	<Double>	Read-write	count parameter

Operations

(click to see Properties)

ADD:| DELETE:| GET| GET (ALL)| COUNT

Some options that you can use for each operations:

- **Getting warnings in response:**NITRO allows you to get warnings in an operation by specifying the "warning" query parameter as "yes". For example, to get warnings while connecting to the Citrix ADC appliance, the URL is as follows:

  http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/login?warning=yes

  If any, the warnings are displayed in the response payload with the HTTP code "209 X-NITRO-WARNING".

- **Authenticated access for individual NITRO operations:**NITRO allows you to logon to the Citrix ADC appliance to perform individual operations. You can use this option instead of creating a NITRO session (using the login object) and then using that session to perform all operations,

  To do this, you must specify the username and password in the request header of the NITRO request as follows:

  X-NITRO-USER:<username>

  X-NITRO-PASS:<password>

  Note: In such cases, make sure that the request header DOES not include the following:

  Cookie:NITRO_AUTH_TOKEN=<tokenvalue>

`*Note:`*

Mandatory parameters are marked in redand placeholder content is marked in <green>.

add

URL:http://<Citrix-ADC-IP-address(NSIP)/nitro/v1/config/csvserver_appqoepolicy_binding

HTTP Method:PUT

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Content-Type:application/json

Request Payload:

```
{

"csvserver_appqoepolicy_binding":{

<b>"name":<String_value>,

</b>"policyname":<String_value>,

"targetlbvserver":<String_value>,

"priority":<Double_value>,

"gotopriorityexpression":<String_value>,

"bindpoint":<String_value>,

"invoke":<Boolean_value>,

"labeltype":<String_value>,

"labelname":<String_value>

}}
```

Response:

HTTP Status Code on Success: 201 Created HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error

delete

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/csvserver_appqoepolicy_binding/name_value<String>

Query-parameters:

args

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/csvserver_appqoepolicy_binding/name_value<String>?**args=policyname:<String_value>,bindpoint:<String_value>,priority:<Double_value>**

HTTP Method:DELETE

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue>

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error

get

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/csvserver_appqoepolicy_binding/name_value<String>

Query-parameters:

filter

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/csvserver_appqoepolicy_binding/name_value<String>?**filter=property-name1:property-value1,property-name2:property-value2**

Use this query-parameter to get the filtered set of csvserver_appqoepolicy_binding resources configured on Citrix ADC. Filtering can be done on any of the properties of the resource.

pagination

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/csvserver_appqoepolicy_binding/name_value<String>?**pagesize=#no;pageno=#no**

Use this query-parameter to get the csvserver_appqoepolicy_binding resources in chunks.

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "csvserver_appqoepolicy_binding": [ {

"priority":<Double_value>,

"bindpoint":<String_value>,

"policyname":<String_value>,

"labelname":<String_value>,

"name":<String_value>,

"targetlbvserver":<String_value>,

"gotopriorityexpression":<String_value>,

"invoke":<Boolean_value>,

"labeltype":<String_value>

}]}
```

get (all)

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/csvserver_appqoepolicy_binding

Query-parameters:

bulkbindings

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/csvserver_appqoepolicy_binding?**bulkbindings=yes**

NITRO allows you to fetch bindings in bulk.

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "csvserver_appqoepolicy_binding": [ {

"priority":<Double_value>,

"bindpoint":<String_value>,

"policyname":<String_value>,

"labelname":<String_value>,

"name":<String_value>,

"targetlbvserver":<String_value>,

"gotopriorityexpression":<String_value>,

"invoke":<Boolean_value>,
```

```
"labeltype":<String_value>

}]}
```

count

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/csvserver_appqoepolicy_binding/name_value<String>?**count=yes**

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{"csvserver_appqoepolicy_binding": [ { "__count": "#no"} ] }
```

csvserver_auditnslogpolicy_binding

Binding object showing the auditnslogpolicy that can be bound to csvserver.

Properties

(click to see Operations)

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
priority	<Double>	Read-write	Priority for the policy.

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
bindpoint	<String>	Read-write	For a rewrite policy, the bind point to which to bind the policy. Note: This parameter applies only to rewrite policies, because content switching policies are evaluated only at request time. Possible values = REQUEST, RESPONSE, ICA_REQUEST, OTHERTCP_REQUEST
policyname	<String>	Read-write	Policies bound to this vserver.
labelname	<String>	Read-write	Name of the label to be invoked.
name	<String>	Read-write	Name of the content switching virtual server to which the content switching policy applies. Minimum length = 1
gotopriorityexpression	<String>	Read-write	Expression or other value specifying the next policy to be evaluated if the current policy evaluates to TRUE. Specify one of the following values: * NEXT - Evaluate the

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
			policy with the next higher priority number. * END - End policy evaluation. * USE_INVOCATION_RESULT - Applicable if this policy invokes another policy label. If the final goto in the invoked policy label has a value of END, the evaluation stops. If the final goto is anything other than END, the current policy label performs a NEXT. * An expression that evaluates to a number. If you specify an expression, the number to which it evaluates determines the next policy to evaluate, as follows: * If the expression evaluates to a higher numbered priority, the policy with that priority is evaluated next. * If the expression evaluates to the priority of the current policy, the policy with the next higher numbered priority is evaluated next. * If the expression evaluates

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
			to a priority number that is numerically higher than the highest numbered priority, policy evaluation ends. An UNDEF event is triggered if: * The expression is invalid. * The expression evaluates to a priority number that is numerically lower than the current policy's priority. * The expression evaluates to a priority number that is between the current policy's priority number (say, 30) and the highest priority number (say, 100), but does not match any configured priority number (for example, the expression evaluates to the number 85). This example assumes that the priority number increments by 10 for every successive policy, and therefore a priority number of 85 does not exist in the policy label.
targetlbvserver	<String>	Read-write	Name of the Load Balancing virtual

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
			server to which the content is switched, if policy rule is evaluated to be TRUE. Example: bind cs vs cs1 -policyname pol1 -priority 101 -targetLBVserver lb1 Note: Use this parameter only in case of Content Switching policy bind operations to a CS vserver. Minimum length = 1
invoke	<Boolean>	Read-write	Invoke a policy label if this policy's rule evaluates to TRUE (valid only for default-syntax policies such as application firewall, transform, integrated cache, rewrite, responder, and content switching).
labeltype	<String>	Read-write	Type of label to be invoked. Possible values = reqvserver, resvserver, policylabel
sc	<String>	Read-only	The state of SureConnect the specified virtual server.

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
			Possible values = ON, OFF
__count	<Double>	Read-only	count parameter

Operations

(click to see Properties)

ADD:| DELETE:| GET| GET (ALL)| COUNT

Some options that you can use for each operations:

- **Getting warnings in response:**NITRO allows you to get warnings in an operation by specifying the "warning" query parameter as "yes". For example, to get warnings while connecting to the Citrix ADC appliance, the URL is as follows:

  http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/login?warning=yes

  If any, the warnings are displayed in the response payload with the HTTP code "209 X-NITRO-WARNING".

- **Authenticated access for individual NITRO operations:**NITRO allows you to logon to the Citrix ADC appliance to perform individual operations. You can use this option instead of creating a NITRO session (using the login object) and then using that session to perform all operations,

  To do this, you must specify the username and password in the request header of the NITRO request as follows:

  X-NITRO-USER:<username>

  X-NITRO-PASS:<password>

  Note: In such cases, make sure that the request header DOES not include the following:

  Cookie:NITRO_AUTH_TOKEN=<tokenvalue>

`*Note:`*

Mandatory parameters are marked in redand placeholder content is marked in <green>.

add

URL:http://<Citrix-ADC-IP-address(NSIP)/nitro/v1/config/csvserver_auditnslogpolicy_binding

HTTP Method:PUT

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Content-Type:application/json

Request Payload:

```
{

"csvserver_auditnslogpolicy_binding":{

<b>"name":<String_value>,

</b>"policyname":<String_value>,

"targetlbvserver":<String_value>,

"priority":<Double_value>,

"gotopriorityexpression":<String_value>,

"bindpoint":<String_value>,

"invoke":<Boolean_value>,

"labeltype":<String_value>,

"labelname":<String_value>

}}
```

Response:

HTTP Status Code on Success: 201 Created HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error

delete

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/csvserver_auditnslogpolicy_binding/name_value<String>

Query-parameters:

args

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/csvserver_auditnslogpolicy_binding/name_value<String>?**args=policyname:<String_value>,bindpoint:<String_value>,priority:<Double_value>**

HTTP Method:DELETE

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue>

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error

get

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/csvserver_auditnslogpolicy_binding/name_value<String>

Query-parameters:

filter

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/csvserver_auditnslogpolicy_binding/name_value<String>?**filter=property-name1:property-value1,property-name2:property-value2**

Use this query-parameter to get the filtered set of csvserver_auditnslogpolicy_binding resources configured on Citrix ADC. Filtering can be done on any of the properties of the resource.

pagination

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/csvserver_auditnslogpolicy_binding/name_value<String>?**pagesize=#no;pageno=#no**

Use this query-parameter to get the csvserver_auditnslogpolicy_binding resources in chunks.

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "csvserver_auditnslogpolicy_binding": [ {

"priority":<Double_value>,

"bindpoint":<String_value>,

"policyname":<String_value>,

"labelname":<String_value>,

"name":<String_value>,

"gotopriorityexpression":<String_value>,

"targetlbvserver":<String_value>,

"invoke":<Boolean_value>,

"labeltype":<String_value>,

"sc":<String_value>

}]}
```

get (all)

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/csvserver_auditnslogpolicy_binding

Query-parameters:

bulkbindings

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/csvserver_auditnslogpolicy_binding?**bulkbindings=yes**

NITRO allows you to fetch bindings in bulk.

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "csvserver_auditnslogpolicy_binding": [ {

"priority":<Double_value>,

"bindpoint":<String_value>,

"policyname":<String_value>,

"labelname":<String_value>,

"name":<String_value>,

"gotopriorityexpression":<String_value>,
```

```
"targetlbvserver":<String_value>,

"invoke":<Boolean_value>,

"labeltype":<String_value>,

"sc":<String_value>

}]}
```

count

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/csvserver_auditnslogpolicy_binding/name_value<String>?**count=yes**

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{"csvserver_auditnslogpolicy_binding": [ { "__count": "#no"} ] }
```

csvserver_auditsyslogpolicy_binding

Binding object showing the auditsyslogpolicy that can be bound to csvserver.

Properties

(click to see Operations)

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
priority	<Double>	Read-write	Priority for the policy.
bindpoint	<String>	Read-write	For a rewrite policy, the bind point to which to bind the policy. Note: This parameter applies only to rewrite policies, because content switching policies are evaluated only at request time. Possible values = REQUEST, RESPONSE, ICA_REQUEST, OTHERTCP_REQUEST
policyname	<String>	Read-write	Policies bound to this vserver.
labelname	<String>	Read-write	Name of the label to be invoked.
name	<String>	Read-write	Name of the content switching virtual server to which the content switching policy applies. Minimum length = 1
gotopriorityexpression	<String>	Read-write	Expression or other value specifying the next policy to be evaluated if the

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
			current policy evaluates to TRUE. Specify one of the following values: * NEXT - Evaluate the policy with the next higher priority number. * END - End policy evaluation. * USE_INVOCATION_RESULT - Applicable if this policy invokes another policy label. If the final goto in the invoked policy label has a value of END, the evaluation stops. If the final goto is anything other than END, the current policy label performs a NEXT. * An expression that evaluates to a number. If you specify an expression, the number to which it evaluates determines the next policy to evaluate, as follows: * If the expression evaluates to a higher numbered priority, the policy with that priority is evaluated next. * If the expression evaluates to the priority of the current policy, the

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
			policy with the next higher numbered priority is evaluated next. * If the expression evaluates to a priority number that is numerically higher than the highest numbered priority, policy evaluation ends. An UNDEF event is triggered if: * The expression is invalid. * The expression evaluates to a priority number that is numerically lower than the current policy's priority. * The expression evaluates to a priority number that is between the current policy's priority number (say, 30) and the highest priority number (say, 100), but does not match any configured priority number (for example, the expression evaluates to the number 85). This example assumes that the priority number increments by 10 for every successive policy, and therefore a priority number of

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
			85 does not exist in the policy label.
targetlbvserver	<String>	Read-write	Name of the Load Balancing virtual server to which the content is switched, if policy rule is evaluated to be TRUE. Example: bind cs vs cs1 -policyname pol1 -priority 101 -targetLBVserver lb1 Note: Use this parameter only in case of Content Switching policy bind operations to a CS vserver. Minimum length = 1
invoke	<Boolean>	Read-write	Invoke a policy label if this policy's rule evaluates to TRUE (valid only for default-syntax policies such as application firewall, transform, integrated cache, rewrite, responder, and content switching).
labeltype	<String>	Read-write	Type of label to be invoked. Possible values = reqvserver, resvserver, policylabel

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
sc	<String>	Read-only	The state of SureConnect the specified virtual server. Possible values = ON, OFF
__count	<Double>	Read-only	count parameter

Operations

(click to see Properties)

ADD:| DELETE:| GET| GET (ALL)| COUNT

Some options that you can use for each operations:

- **Getting warnings in response:**NITRO allows you to get warnings in an operation by specifying the "warning" query parameter as "yes". For example, to get warnings while connecting to the Citrix ADC appliance, the URL is as follows:

  http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/login?warning=yes

  If any, the warnings are displayed in the response payload with the HTTP code "209 X-NITRO-WARNING".

- **Authenticated access for individual NITRO operations:**NITRO allows you to logon to the Citrix ADC appliance to perform individual operations. You can use this option instead of creating a NITRO session (using the login object) and then using that session to perform all operations,

  To do this, you must specify the username and password in the request header of the NITRO request as follows:

  X-NITRO-USER:<username>

  X-NITRO-PASS:<password>

  Note: In such cases, make sure that the request header DOES not include the following:

  Cookie:NITRO_AUTH_TOKEN=<tokenvalue>

`*Note:`*

Mandatory parameters are marked in redand placeholder content is marked in <green>.

add

URL:http://<Citrix-ADC-IP-address(NSIP)/nitro/v1/config/csvserver_auditsyslogpolicy_binding

HTTP Method:PUT

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Content-Type:application/json

Request Payload:

```
{

"csvserver_auditsyslogpolicy_binding":{

<b>"name":<String_value>,

</b>"policyname":<String_value>,

"targetlbvserver":<String_value>,

"priority":<Double_value>,

"gotopriorityexpression":<String_value>,

"bindpoint":<String_value>,

"invoke":<Boolean_value>,

"labeltype":<String_value>,

"labelname":<String_value>

}}
```

Response:

HTTP Status Code on Success: 201 Created HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error

delete

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/csvserver_auditsyslogpolicy_binding/name_value<String>

Query-parameters:

args

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/csvserver_auditsyslogpolicy_binding/name_value<String>?**args=policyname:<String_value>,bindpoint:<String_value>,priority:<Double_value>**

HTTP Method:DELETE

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue>

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error

get

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/csvserver_auditsyslogpolicy_binding/name_value<String>

Query-parameters:

filter

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/csvserver_auditsyslogpolicy_binding/name_value<String>?**filter=property-name1:property-value1,property-name2:property-value2**

Use this query-parameter to get the filtered set of csvserver_auditsyslogpolicy_binding resources configured on Citrix ADC. Filtering can be done on any of the properties of the resource.

pagination

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/csvserver_auditsyslogpolicy_binding/name_value<String>?**pagesize=#no;pageno=#no**

Use this query-parameter to get the csvserver_auditsyslogpolicy_binding resources in chunks.

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "csvserver_auditsyslogpolicy_binding": [ {

"priority":<Double_value>,

"bindpoint":<String_value>,

"policyname":<String_value>,

"labelname":<String_value>,

"name":<String_value>,

"gotopriorityexpression":<String_value>,

"targetlbvserver":<String_value>,

"invoke":<Boolean_value>,

"labeltype":<String_value>,

"sc":<String_value>
```

```
}]}
```

get (all)

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/csvserver_auditsyslogpolicy_binding

Query-parameters:

bulkbindings

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/csvserver_auditsyslogpolicy_binding?**bulkbindings=yes**

NITRO allows you to fetch bindings in bulk.

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "csvserver_auditsyslogpolicy_binding": [ {

"priority":<Double_value>,

"bindpoint":<String_value>,

"policyname":<String_value>,

"labelname":<String_value>,
```

```
"name":<String_value>,

"gotopriorityexpression":<String_value>,

"targetlbvserver":<String_value>,

"invoke":<Boolean_value>,

"labeltype":<String_value>,

"sc":<String_value>

}]}
```

count

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/csvserver_auditsyslogpolicy_binding/name_value<String>?**count=yes**

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{"csvserver_auditsyslogpolicy_binding": [ { "__count": "#no"} ] }
```

csvserver_authorizationpolicy_binding

Binding object showing the authorizationpolicy that can be bound to csvserver.

Properties

(click to see Operations)

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
priority	<Double>	Read-write	Priority for the policy.
bindpoint	<String>	Read-write	The bindpoint to which the policy is bound. Possible values = REQUEST, RESPONSE, ICA_REQUEST, OTHERTCP_REQUEST
policyname	<String>	Read-write	Policies bound to this vserver.
labelname	<String>	Read-write	Name of the label invoked.
name	<String>	Read-write	Name of the content switching virtual server to which the content switching policy applies. Minimum length = 1
targetlbvserver	<String>	Read-write	Name of the Load Balancing virtual server to which the content is switched, if policy rule is evaluated to be TRUE. Example: bind cs vs cs1

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
			-policyname pol1 -priority 101 -targetLBVserver lb1 Note: Use this parameter only in case of Content Switching policy bind operations to a CS vserver. Minimum length = 1
gotopriorityexpression	<String>	Read-write	Expression specifying the priority of the next policy which will get evaluated if the current policy rule evaluates to TRUE.
invoke	<Boolean>	Read-write	Invoke flag.
labeltype	<String>	Read-write	The invocation type. Possible values = reqvserver, resvserver, policylabel
sc	<String>	Read-only	The state of SureConnect the specified virtual server. Possible values = ON, OFF
__count	<Double>	Read-only	count parameter

Operations

(click to see Properties)

ADD:| DELETE:| GET| GET (ALL)| COUNT

Some options that you can use for each operations:

- **Getting warnings in response:**NITRO allows you to get warnings in an operation by specifying the "warning" query parameter as "yes". For example, to get warnings while connecting to the Citrix ADC appliance, the URL is as follows:

  http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/login?warning=yes

  If any, the warnings are displayed in the response payload with the HTTP code "209 X-NITRO-WARNING".

- **Authenticated access for individual NITRO operations:**NITRO allows you to logon to the Citrix ADC appliance to perform individual operations. You can use this option instead of creating a NITRO session (using the login object) and then using that session to perform all operations,

  To do this, you must specify the username and password in the request header of the NITRO request as follows:

  X-NITRO-USER:<username>

  X-NITRO-PASS:<password>

  Note: In such cases, make sure that the request header DOES not include the following:

  Cookie:NITRO_AUTH_TOKEN=<tokenvalue>

`*Note:`*

Mandatory parameters are marked in redand placeholder content is marked in <green>.

add

URL:http://<Citrix-ADC-IP-address(NSIP)/nitro/v1/config/csvserver_authorizationpolicy_binding

HTTP Method:PUT

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Content-Type:application/json

Request Payload:

```
{

"csvserver_authorizationpolicy_binding":{

<b>"name":<String_value>,

</b>"policyname":<String_value>,

"targetlbvserver":<String_value>,

"priority":<Double_value>,

"gotopriorityexpression":<String_value>,

"bindpoint":<String_value>,

"invoke":<Boolean_value>,

"labeltype":<String_value>,

"labelname":<String_value>

}}
```

Response:

HTTP Status Code on Success: 201 Created HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error

delete

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/csvserver_authorizationpolicy_binding/name_value<String>

Query-parameters:

args

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/csvserver_authorizationpolicy_binding/name_value<String>?**args=policyname:<String_value>,bindpoint:<String_value>,priority:<Double_value>**

HTTP Method:DELETE

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue>

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error

get

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/csvserver_authorizationpolicy_binding/name_value<String>

Query-parameters:

filter

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/csvserver_authorizationpolicy_binding/name_value<String>?**filter=property-name1:property-value1,property-name2:property-value2**

Use this query-parameter to get the filtered set of csvserver_authorizationpolicy_binding resources configured on Citrix ADC. Filtering can be done on any of the properties of the resource.

pagination

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/csvserver_authorizationpolicy_binding/name_value<String>?**pagesize=#no;pageno=#no**

Use this query-parameter to get the csvserver_authorizationpolicy_binding resources in chunks.

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "csvserver_authorizationpolicy_binding": [ {

"priority":<Double_value>,

"bindpoint":<String_value>,

"policyname":<String_value>,

"labelname":<String_value>,

"name":<String_value>,

"targetlbvserver":<String_value>,

"gotopriorityexpression":<String_value>,

"invoke":<Boolean_value>,

"labeltype":<String_value>,

"sc":<String_value>

}]}
```

get (all)

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/csvserver_authorizationpolicy_binding

Query-parameters:

bulkbindings

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/csvserver_authorizationpolicy_binding?**bulkbindings=yes**

NITRO allows you to fetch bindings in bulk.

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "csvserver_authorizationpolicy_binding": [ {

"priority":<Double_value>,

"bindpoint":<String_value>,

"policyname":<String_value>,

"labelname":<String_value>,

"name":<String_value>,

"targetlbvserver":<String_value>,

"gotopriorityexpression":<String_value>,

"invoke":<Boolean_value>,

"labeltype":<String_value>,

"sc":<String_value>
```

```
}]}
```

count

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/csvserver_authorizationpolicy_binding/name_value<String>?**count=yes**

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{"csvserver_authorizationpolicy_binding": [ { "__count": "#no"} ] }
```

csvserver_binding

Binding object which returns the resources bound to csvserver.

Properties

(click to see Operations)

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
name	<String>	Read-write	Name of a content switching virtual server for which to display information,

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
			including the policies bound to the virtual server. To display a list of all configured Content Switching virtual servers, do not specify a value for this parameter. Minimum length = 1
csvserver_spilloverpolicy_binding	<csvserver_spillover policy_binding[]>	Read-only	spilloverpolicy that can be bound to csvserver.
csvserver_auditnslogpolicy_binding	<csvserver_auditnslogpolicy_binding[]>	Read-only	auditnslogpolicy that can be bound to csvserver.
csvserver_domain_binding	<csvserver_domain_binding[]>	Read-only	domain that can be bound to csvserver.
csvserver_filterpolicy_binding	<csvserver_filterpolicy_binding[]>	Read-only	filterpolicy that can be bound to csvserver.
csvserver_cmppolicy_binding	<csvserver_cmppolicy_binding[]>	Read-only	cmppolicy that can be bound to csvserver.
csvserver_lbvserver_binding	<csvserver_lbvserver_binding[]>	Read-only	lbvserver that can be bound to csvserver.
csvserver_appflowpolicy_binding	<csvserver_appflowpolicy_binding[]>	Read-only	appflowpolicy that can be bound to csvserver.
csvserver_responderpolicy_binding	<csvserver_responderpolicy_binding[]>	Read-only	responderpolicy that can be bound to csvserver.

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
csvserver_transformpolicy_binding	<csvserver_transformpolicy_binding[]>	Read-only	transformpolicy that can be bound to csvserver.
csvserver_vpnvserver_binding	<csvserver_vpnvserver_binding[]>	Read-only	vpnvserver that can be bound to csvserver.
csvserver_contentinspectionpolicy_binding	<csvserver_contentinspectionpolicy_binding[]>	Read-only	contentinspectionpolicy that can be bound to csvserver.
csvserver_feopolicy_binding	<csvserver_feopolicy_binding[]>	Read-only	feopolicy that can be bound to csvserver.
csvserver_authorizationpolicy_binding	<csvserver_authorizationpolicy_binding[]>	Read-only	authorizationpolicy that can be bound to csvserver.
csvserver_cachepolicy_binding	<csvserver_cachepolicy_binding[]>	Read-only	cachepolicy that can be bound to csvserver.
csvserver_rewritepolicy_binding	<csvserver_rewritepolicy_binding[]>	Read-only	rewritepolicy that can be bound to csvserver.
csvserver_cspolicy_binding	<csvserver_cspolicy_binding[]>	Read-only	cspolicy that can be bound to csvserver.
csvserver_gslbvserver_binding	<csvserver_gslbvserver_binding[]>	Read-only	gslbvserver that can be bound to csvserver.
csvserver_appqoepolicy_binding	<csvserver_appqoepolicy_binding[]>	Read-only	appqoepolicy that can be bound to csvserver.
csvserver_tmtrafficpolicy_binding	<csvserver_tmtrafficpolicy_binding[]>	Read-only	tmtrafficpolicy that can be bound to csvserver.

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
csvserver_auditsyslogpolicy_binding	<csvserver_auditsyslogpolicy_binding[]>	Read-only	auditsyslogpolicy that can be bound to csvserver.
csvserver_appfwpolicy_binding	<csvserver_appfwpolicy_binding[]>	Read-only	appfwpolicy that can be bound to csvserver.

Operations

(click to see Properties)

GET| GET (ALL)

Some options that you can use for each operations:

- **Getting warnings in response:**NITRO allows you to get warnings in an operation by specifying the "warning" query parameter as "yes". For example, to get warnings while connecting to the Citrix ADC appliance, the URL is as follows:

  http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/login?warning=yes

  If any, the warnings are displayed in the response payload with the HTTP code "209 X-NITRO-WARNING".

- **Authenticated access for individual NITRO operations:**NITRO allows you to logon to the Citrix ADC appliance to perform individual operations. You can use this option instead of creating a NITRO session (using the login object) and then using that session to perform all operations,

  To do this, you must specify the username and password in the request header of the NITRO request as follows:

  X-NITRO-USER:<username>

  X-NITRO-PASS:<password>

  Note: In such cases, make sure that the request header DOES not include the following:

  Cookie:NITRO_AUTH_TOKEN=<tokenvalue>

`*Note:`*

Mandatory parameters are marked in redand placeholder content is marked in <green>.

get

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/csvserver_binding/name_value<String>

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "csvserver_binding": [ {

"name":<String_value>,

"csvserver_auditnslogpolicy_binding":<csvserver_auditnslogpolicy_binding[]
_value>,

"csvserver_tmtrafficpolicy_binding":<csvserver_tmtrafficpolicy_binding[]_v
alue>,

"csvserver_lbvserver_binding":<csvserver_lbvserver_binding[]_value>,

"csvserver_responderpolicy_binding":<csvserver_responderpolicy_binding[]_v
alue>,

"csvserver_gslbvserver_binding":<csvserver_gslbvserver_binding[]_value>,

"csvserver_cachepolicy_binding":<csvserver_cachepolicy_binding[]_value>,

"csvserver_filterpolicy_binding":<csvserver_filterpolicy_binding[]_value>,
```

```
"csvserver_transformpolicy_binding":<csvserver_transformpolicy_binding[]_v
alue>,

"csvserver_appflowpolicy_binding":<csvserver_appflowpolicy_binding[]_value
>,

"csvserver_appfwpolicy_binding":<csvserver_appfwpolicy_binding[]_value>,

"csvserver_authorizationpolicy_binding":<csvserver_authorizationpolicy_bin
ding[]_value>,

"csvserver_auditsyslogpolicy_binding":<csvserver_auditsyslogpolicy_binding
[]_value>,

"csvserver_cmppolicy_binding":<csvserver_cmppolicy_binding[]_value>,

"csvserver_rewritepolicy_binding":<csvserver_rewritepolicy_binding[]_value
>,

"csvserver_cspolicy_binding":<csvserver_cspolicy_binding[]_value>,

"csvserver_contentinspectionpolicy_binding":<csvserver_contentinspectionpo
licy_binding[]_value>,

"csvserver_spilloverpolicy_binding":<csvserver_spilloverpolicy_binding[]_v
alue>,

"csvserver_appqoepolicy_binding":<csvserver_appqoepolicy_binding[]_value>,

"csvserver_feopolicy_binding":<csvserver_feopolicy_binding[]_value>,

"csvserver_vpnvserver_binding":<csvserver_vpnvserver_binding[]_value>,

"csvserver_domain_binding":<csvserver_domain_binding[]_value>

}]}
```

get (all)

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/csvserver_binding

Query-parameters:

bulkbindings

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/csvserver_binding?**bulkbindings=yes**

NITRO allows you to fetch bindings in bulk.

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "csvserver_binding": [ {

"name":<String_value>,

"csvserver_auditnslogpolicy_binding":<csvserver_auditnslogpolicy_binding[]
_value>,

"csvserver_tmtrafficpolicy_binding":<csvserver_tmtrafficpolicy_binding[]_v
alue>,

"csvserver_lbvserver_binding":<csvserver_lbvserver_binding[]_value>,

"csvserver_responderpolicy_binding":<csvserver_responderpolicy_binding[]_v
alue>,

"csvserver_gslbvserver_binding":<csvserver_gslbvserver_binding[]_value>,

"csvserver_cachepolicy_binding":<csvserver_cachepolicy_binding[]_value>,

"csvserver_filterpolicy_binding":<csvserver_filterpolicy_binding[]_value>,
```

```
"csvserver_transformpolicy_binding":<csvserver_transformpolicy_binding[]_v
alue>,

"csvserver_appflowpolicy_binding":<csvserver_appflowpolicy_binding[]_value
>,

"csvserver_appfwpolicy_binding":<csvserver_appfwpolicy_binding[]_value>,

"csvserver_authorizationpolicy_binding":<csvserver_authorizationpolicy_bin
ding[]_value>,

"csvserver_auditsyslogpolicy_binding":<csvserver_auditsyslogpolicy_binding
[]_value>,

"csvserver_cmppolicy_binding":<csvserver_cmppolicy_binding[]_value>,

"csvserver_rewritepolicy_binding":<csvserver_rewritepolicy_binding[]_value
>,

"csvserver_cspolicy_binding":<csvserver_cspolicy_binding[]_value>,

"csvserver_contentinspectionpolicy_binding":<csvserver_contentinspectionpo
licy_binding[]_value>,

"csvserver_spilloverpolicy_binding":<csvserver_spilloverpolicy_binding[]_v
alue>,

"csvserver_appqoepolicy_binding":<csvserver_appqoepolicy_binding[]_value>,

"csvserver_feopolicy_binding":<csvserver_feopolicy_binding[]_value>,

"csvserver_vpnvserver_binding":<csvserver_vpnvserver_binding[]_value>,

"csvserver_domain_binding":<csvserver_domain_binding[]_value>

}]}
```

csvserver_cachepolicy_binding

Binding object showing the cachepolicy that can be bound to csvserver.

Properties

(click to see Operations)

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
priority	<Double>	Read-write	Priority for the policy.
bindpoint	<String>	Read-write	The bindpoint to which the policy is bound. Possible values = REQUEST, RESPONSE, ICA_REQUEST, OTHERTCP_REQUEST
policyname	<String>	Read-write	Policies bound to this vserver.
labelname	<String>	Read-write	Name of the label invoked.
name	<String>	Read-write	Name of the content switching virtual server to which the content switching policy applies. Minimum length = 1
targetlbvserver	<String>	Read-write	Name of the Load Balancing virtual server to which the content is switched, if policy rule is evaluated to be TRUE. Example: bind cs vs cs1 -policyname pol1 -priority 101 -targetLBVserver lb1 Note: Use this

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
			parameter only in case of Content Switching policy bind operations to a CS vserver. Minimum length = 1
gotopriorityexpression	<String>	Read-write	Expression specifying the priority of the next policy which will get evaluated if the current policy rule evaluates to TRUE.
invoke	<Boolean>	Read-write	Invoke flag.
labeltype	<String>	Read-write	The invocation type. Possible values = reqvserver, resvserver, policylabel
__count	<Double>	Read-write	count parameter

Operations

(click to see Properties)

ADD:| DELETE:| GET| GET (ALL)| COUNT

Some options that you can use for each operations:

- **Getting warnings in response:**NITRO allows you to get warnings in an operation by specifying the "warning" query parameter as "yes". For example, to get warnings while connecting to the Citrix ADC appliance, the URL is as follows:

  http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/login?warning=yes

  If any, the warnings are displayed in the response payload with the HTTP code "209 X-NITRO-WARNING".

- **Authenticated access for individual NITRO operations:**NITRO allows you to logon to the Citrix ADC appliance to perform individual operations. You can use this option instead of creating a NITRO session (using the login object) and then using that session to perform all operations,

  To do this, you must specify the username and password in the request header of the NITRO request as follows:

  X-NITRO-USER:<username>

  X-NITRO-PASS:<password>

  Note: In such cases, make sure that the request header DOES not include the following:

  Cookie:NITRO_AUTH_TOKEN=<tokenvalue>

`*Note:`*

Mandatory parameters are marked in redand placeholder content is marked in <green>.

add

URL:http://<Citrix-ADC-IP-address(NSIP)/nitro/v1/config/csvserver_cachepolicy_binding

HTTP Method:PUT

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Content-Type:application/json

Request Payload:

```
{

"csvserver_cachepolicy_binding":{

<b>"name":<String_value>,

</b>"policyname":<String_value>,

"targetlbvserver":<String_value>,

"priority":<Double_value>,
```

```
"gotopriorityexpression":<String_value>,

"bindpoint":<String_value>,

"invoke":<Boolean_value>,

"labeltype":<String_value>,

"labelname":<String_value>

}}
```

Response:

HTTP Status Code on Success: 201 Created HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error

delete

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/csvserver_cachepolicy_binding/name_value<String>

Query-parameters:

args

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/csvserver_cachepolicy_binding/name_value<String>?
args=policyname:<String_value>,bindpoint:<String_value>,priority:<Double_value>

HTTP Method:DELETE

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue>

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error

get

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/csvserver_cachepolicy_binding/name_value<String>

Query-parameters:

filter

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/csvserver_cachepolicy_binding/name_value<String>?**filter=property-name1:property-value1,property-name2:property-value2**

Use this query-parameter to get the filtered set of csvserver_cachepolicy_binding resources configured on Citrix ADC. Filtering can be done on any of the properties of the resource.

pagination

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/csvserver_cachepolicy_binding/name_value<String>?**pagesize=#no;pageno=#no**

Use this query-parameter to get the csvserver_cachepolicy_binding resources in chunks.

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "csvserver_cachepolicy_binding": [ {

"priority":<Double_value>,

"bindpoint":<String_value>,
```

```
"policyname":<String_value>,

"labelname":<String_value>,

"name":<String_value>,

"targetlbvserver":<String_value>,

"gotopriorityexpression":<String_value>,

"invoke":<Boolean_value>,

"labeltype":<String_value>

}]}
```

get (all)

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/csvserver_cachepolicy_binding

Query-parameters:

bulkbindings

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/csvserver_cachepolicy_binding?**bulkbindings=yes**

NITRO allows you to fetch bindings in bulk.

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "csvserver_cachepolicy_binding": [ {

"priority":<Double_value>,

"bindpoint":<String_value>,

"policyname":<String_value>,

"labelname":<String_value>,

"name":<String_value>,

"targetlbvserver":<String_value>,

"gotopriorityexpression":<String_value>,

"invoke":<Boolean_value>,

"labeltype":<String_value>

}]}
```

count

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/csvserver_cachepolicy_binding/name_value<String>?**count=yes**

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{"csvserver_cachepolicy_binding": [ { "__count": "#no"} ] }
```

csvserver_cmppolicy_binding

Binding object showing the cmppolicy that can be bound to csvserver.

Properties

(click to see Operations)

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
priority	<Double>	Read-write	Priority for the policy.
bindpoint	<String>	Read-write	The bindpoint to which the policy is bound. Possible values = REQUEST, RESPONSE, ICA_REQUEST, OTHERTCP_REQUEST
policyname	<String>	Read-write	Policies bound to this vserver.
labelname	<String>	Read-write	Name of the label invoked.
name	<String>	Read-write	Name of the content switching virtual server to which the content switching policy applies.

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
			Minimum length = 1
targetlbvserver	<String>	Read-write	Name of the Load Balancing virtual server to which the content is switched, if policy rule is evaluated to be TRUE. Example: bind cs vs cs1 -policyname pol1 -priority 101 -targetLBVserver lb1 Note: Use this parameter only in case of Content Switching policy bind operations to a CS vserver. Minimum length = 1
gotopriorityexpression	<String>	Read-write	Expression specifying the priority of the next policy which will get evaluated if the current policy rule evaluates to TRUE.
invoke	<Boolean>	Read-write	Invoke flag.
labeltype	<String>	Read-write	The invocation type. Possible values = reqvserver, resvserver, policylabel
sc	<String>	Read-only	The state of SureConnect the

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
			specified virtual server. Possible values = ON, OFF
__count	<Double>	Read-only	count parameter

Operations

(click to see Properties)

ADD:| DELETE:| GET| GET (ALL)| COUNT

Some options that you can use for each operations:

- **Getting warnings in response:**NITRO allows you to get warnings in an operation by specifying the "warning" query parameter as "yes". For example, to get warnings while connecting to the Citrix ADC appliance, the URL is as follows:

  http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/login?warning=yes

  If any, the warnings are displayed in the response payload with the HTTP code "209 X-NITRO-WARNING".

- **Authenticated access for individual NITRO operations:**NITRO allows you to logon to the Citrix ADC appliance to perform individual operations. You can use this option instead of creating a NITRO session (using the login object) and then using that session to perform all operations,

  To do this, you must specify the username and password in the request header of the NITRO request as follows:

  X-NITRO-USER:<username>

  X-NITRO-PASS:<password>

  Note: In such cases, make sure that the request header DOES not include the following:

  Cookie:NITRO_AUTH_TOKEN=<tokenvalue>

`*Note:` *

Mandatory parameters are marked in redand placeholder content is marked in <green>.

add

URL:http://<Citrix-ADC-IP-address(NSIP)/nitro/v1/config/csvserver_cmppolicy_binding

HTTP Method:PUT

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Content-Type:application/json

Request Payload:

```
{

"csvserver_cmppolicy_binding":{

<b>"name":<String_value>,

</b>"policyname":<String_value>,

"targetlbvserver":<String_value>,

"priority":<Double_value>,

"gotopriorityexpression":<String_value>,

"bindpoint":<String_value>,

"invoke":<Boolean_value>,

"labeltype":<String_value>,

"labelname":<String_value>

}}
```

Response:

HTTP Status Code on Success: 201 Created HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error

delete

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/csvserver_cmppolicy_binding/name_value<String>

Query-parameters:

args

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/csvserver_cmppolicy_binding/name_value<String>?**args=policyname:<String_value>,bindpoint:<String_value>,priority:<Double_value>**

HTTP Method:DELETE

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue>

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error

get

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/csvserver_cmppolicy_binding/name_value<String>

Query-parameters:

filter

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/csvserver_cmppolicy_binding/name_value<String>?**filter=property-name1:property-value1,property-name2:property-value2**

Use this query-parameter to get the filtered set of csvserver_cmppolicy_binding resources configured on Citrix ADC. Filtering can be done on any of the properties of the resource.

pagination

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/csvserver_cmppolicy_binding/name_value<String>?**pagesize=#no;pageno=#no**

Use this query-parameter to get the csvserver_cmppolicy_binding resources in chunks.

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "csvserver_cmppolicy_binding": [ {

"priority":<Double_value>,

"bindpoint":<String_value>,

"policyname":<String_value>,

"labelname":<String_value>,

"name":<String_value>,

"targetlbvserver":<String_value>,

"gotopriorityexpression":<String_value>,

"invoke":<Boolean_value>,

"labeltype":<String_value>,

"sc":<String_value>

}]}
```

get (all)

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/csvserver_cmppolicy_binding

Query-parameters:

bulkbindings

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/csvserver_cmppolicy_binding?**bulkbindings=yes**

NITRO allows you to fetch bindings in bulk.

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "csvserver_cmppolicy_binding": [ {

"priority":<Double_value>,

"bindpoint":<String_value>,

"policyname":<String_value>,

"labelname":<String_value>,

"name":<String_value>,

"targetlbvserver":<String_value>,
```

```
"gotopriorityexpression":<String_value>,

"invoke":<Boolean_value>,

"labeltype":<String_value>,

"sc":<String_value>

}]}
```

count

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/csvserver_cmppolicy_binding/name_value<String>?**count=yes**

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{"csvserver_cmppolicy_binding": [ { "__count": "#no"} ] }
```

csvserver_contentinspectionpolicy_binding

Binding object showing the contentinspectionpolicy that can be bound to csvserver.

Properties

(click to see Operations)

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
priority	<Double>	Read-write	Priority for the policy.
bindpoint	<String>	Read-write	The bindpoint to which the policy is bound. Possible values = REQUEST, RESPONSE, ICA_REQUEST, OTHERTCP_REQUEST
policyname	<String>	Read-write	Policies bound to this vserver.
labelname	<String>	Read-write	Name of the label invoked.
name	<String>	Read-write	Name of the content switching virtual server to which the content switching policy applies. Minimum length = 1
targetlbvserver	<String>	Read-write	Name of the Load Balancing virtual server to which the content is switched, if policy rule is evaluated to be TRUE. Example: bind cs vs cs1 -policyname pol1 -priority 101 -targetLBVserver lb1 Note: Use this

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
			parameter only in case of Content Switching policy bind operations to a CS vserver. Minimum length = 1
gotopriorityexpression	<String>	Read-write	Expression specifying the priority of the next policy which will get evaluated if the current policy rule evaluates to TRUE.
invoke	<Boolean>	Read-write	Invoke flag.
labeltype	<String>	Read-write	The invocation type. Possible values = reqvserver, resvserver, policylabel
__count	<Double>	Read-write	count parameter

Operations

(click to see Properties)

ADD:| DELETE:| GET| GET (ALL)| COUNT

Some options that you can use for each operations:

- **Getting warnings in response:**NITRO allows you to get warnings in an operation by specifying the "warning" query parameter as "yes". For example, to get warnings while connecting to the Citrix ADC appliance, the URL is as follows:

  http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/login?warning=yes

  If any, the warnings are displayed in the response payload with the HTTP code "209 X-NITRO-WARNING".

- **Authenticated access for individual NITRO operations:**NITRO allows you to logon to the Citrix ADC appliance to perform individual operations. You can use this option instead of creating a NITRO session (using the login object) and then using that session to perform all operations,

  To do this, you must specify the username and password in the request header of the NITRO request as follows:

  X-NITRO-USER:<username>

  X-NITRO-PASS:<password>

  Note: In such cases, make sure that the request header DOES not include the following:

  Cookie:NITRO_AUTH_TOKEN=<tokenvalue>

`*Note:`*

Mandatory parameters are marked in redand placeholder content is marked in <green>.

add

URL:http://<Citrix-ADC-IP-address(NSIP)/nitro/v1/config/csvserver_contentinspectionpolicy_binding

HTTP Method:PUT

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Content-Type:application/json

Request Payload:

```
{

"csvserver_contentinspectionpolicy_binding":{

<b>"name":<String_value>,

</b>"policyname":<String_value>,

"targetlbvserver":<String_value>,
```

```
"priority":<Double_value>,

"gotopriorityexpression":<String_value>,

"bindpoint":<String_value>,

"invoke":<Boolean_value>,

"labeltype":<String_value>,

"labelname":<String_value>

}}
```

Response:

HTTP Status Code on Success: 201 Created HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error

delete

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/csvserver_contentinspectionpolicy_binding/name_value<String>

Query-parameters:

args

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/csvserver_contentinspectionpolicy_binding/name_value<String>?**args=policyname:<String_value>,bindpoint:<String_value>,priority:<Double_value>**

HTTP Method:DELETE

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue>

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error

get

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/csvserver_contentinspectionpolicy_binding/name_value<String>

Query-parameters:

filter

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/csvserver_contentinspectionpolicy_binding/name_value<String>?**filter=property-name1:property-value1,property-name2:property-value2**

Use this query-parameter to get the filtered set of csvserver_contentinspectionpolicy_binding resources configured on Citrix ADC. Filtering can be done on any of the properties of the resource.

pagination

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/csvserver_contentinspectionpolicy_binding/name_value<String>?**pagesize=#no;pageno=#no**

Use this query-parameter to get the csvserver_contentinspectionpolicy_binding resources in chunks.

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "csvserver_contentinspectionpolicy_binding": [ {

"priority":<Double_value>,

"bindpoint":<String_value>,

"policyname":<String_value>,

"labelname":<String_value>,

"name":<String_value>,

"targetlbvserver":<String_value>,

"gotopriorityexpression":<String_value>,

"invoke":<Boolean_value>,

"labeltype":<String_value>

}]}
```

get (all)

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/csvserver_contentinspectionpolicy_binding

Query-parameters:

bulkbindings

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/csvserver_contentinspectionpolicy_binding?**bulkbindings=yes**

NITRO allows you to fetch bindings in bulk.

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "csvserver_contentinspectionpolicy_binding": [ {

"priority":<Double_value>,

"bindpoint":<String_value>,

"policyname":<String_value>,

"labelname":<String_value>,

"name":<String_value>,

"targetlbvserver":<String_value>,

"gotopriorityexpression":<String_value>,

"invoke":<Boolean_value>,

"labeltype":<String_value>

}]}
```

count

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/csvserver_contentinspectionpolicy_binding/name_value<String>?**count=yes**

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{"csvserver_contentinspectionpolicy_binding": [ { "__count": "#no"} ] }
```

csvserver_cspolicy_binding

Binding object showing the cspolicy that can be bound to csvserver.

Properties

(click to see Operations)

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
priority	<Double>	Read-write	Priority for the policy.
bindpoint	<String>	Read-write	The bindpoint to which the policy is bound. Possible values = REQUEST, RESPONSE, ICA_REQUEST, OTHERTCP_REQUEST
policyname	<String>	Read-write	Policies bound to this vserver.

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
labelname	<String>	Read-write	Name of the label invoked.
name	<String>	Read-write	Name of the content switching virtual server to which the content switching policy applies. Minimum length = 1
gotopriorityexpression	<String>	Read-write	Expression specifying the priority of the next policy which will get evaluated if the current policy rule evaluates to TRUE.
targetlbvserver	<String>	Read-write	target vserver name.
invoke	<Boolean>	Read-write	Invoke flag.
labeltype	<String>	Read-write	The invocation type. Possible values = reqvserver, resvserver, policylabel
rule	<String>	Read-only	Rule.
hits	<Double>	Read-only	Number of hits.
vserverid	<String>	Read-only	Vserver Id of vserver.
pipolicyhits	<Double>	Read-only	Number of hits.
cookieipport	<String>	Read-only	Vserver id of the lb vserver that is inserted into the set-cookie HTTP header.

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
__count	<Double>	Read-only	count parameter

Operations

(click to see Properties)

ADD:| DELETE:| GET| GET (ALL)| COUNT

Some options that you can use for each operations:

- **Getting warnings in response:**NITRO allows you to get warnings in an operation by specifying the "warning" query parameter as "yes". For example, to get warnings while connecting to the Citrix ADC appliance, the URL is as follows:

  http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/login?warning=yes

  If any, the warnings are displayed in the response payload with the HTTP code "209 X-NITRO-WARNING".
- **Authenticated access for individual NITRO operations:**NITRO allows you to logon to the Citrix ADC appliance to perform individual operations. You can use this option instead of creating a NITRO session (using the login object) and then using that session to perform all operations,

  To do this, you must specify the username and password in the request header of the NITRO request as follows:

  X-NITRO-USER:<username>

  X-NITRO-PASS:<password>

  Note: In such cases, make sure that the request header DOES not include the following:

  Cookie:NITRO_AUTH_TOKEN=<tokenvalue>

`*Note:`*

Mandatory parameters are marked in redand placeholder content is marked in <green>.

add

URL:http://<Citrix-ADC-IP-address(NSIP)/nitro/v1/config/csvserver_cspolicy_binding

HTTP Method:PUT

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Content-Type:application/json

Request Payload:

```
{

"csvserver_cspolicy_binding":{

<b>"name":<String_value>,

</b>"policyname":<String_value>,

"targetlbvserver":<String_value>,

"priority":<Double_value>,

"gotopriorityexpression":<String_value>,

"bindpoint":<String_value>,

"invoke":<Boolean_value>,

"labeltype":<String_value>,

"labelname":<String_value>

}}
```

Response:

HTTP Status Code on Success: 201 Created HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error

delete

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/csvserver_cspolicy_binding/name_value<String>

Query-parameters:

args

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/csvserver_cspolicy_binding/name_value<String>?**args=policyname:<String_value>,bindpoint:<String_value>,priority:<Double_value>**

HTTP Method:DELETE

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue>

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error

get

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/csvserver_cspolicy_binding/name_value<String>

Query-parameters:

filter

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/csvserver_cspolicy_binding/name_value<String>?**filter=property-name1:property-value1,property-name2:property-value2**

Use this query-parameter to get the filtered set of csvserver_cspolicy_binding resources configured on Citrix ADC. Filtering can be done on any of the properties of the resource.

pagination

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/csvserver_cspolicy_binding/name_value<String>?**pagesize=#no;pageno=#no**

Use this query-parameter to get the csvserver_cspolicy_binding resources in chunks.

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "csvserver_cspolicy_binding": [ {

"priority":<Double_value>,

"bindpoint":<String_value>,

"policyname":<String_value>,

"labelname":<String_value>,

"name":<String_value>,

"gotopriorityexpression":<String_value>,

"targetlbvserver":<String_value>,

"invoke":<Boolean_value>,

"labeltype":<String_value>,

"rule":<String_value>,

"hits":<Double_value>,

"vserverid":<String_value>,

"pipolicyhits":<Double_value>,

"cookieipport":<String_value>
```

```
}]}
```

get (all)

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/csvserver_cspolicy_binding

Query-parameters:

bulkbindings

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/csvserver_cspolicy_binding?**bulkbindings=yes**

NITRO allows you to fetch bindings in bulk.

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "csvserver_cspolicy_binding": [ {

"priority":<Double_value>,

"bindpoint":<String_value>,

"policyname":<String_value>,

"labelname":<String_value>,
```

```
"name":<String_value>,

"gotopriorityexpression":<String_value>,

"targetlbvserver":<String_value>,

"invoke":<Boolean_value>,

"labeltype":<String_value>,

"rule":<String_value>,

"hits":<Double_value>,

"vserverid":<String_value>,

"pipolicyhits":<Double_value>,

"cookieipport":<String_value>

}]}
```

count

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/csvserver_cspolicy_binding/name_value<String>?**count=yes**

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{"csvserver_cspolicy_binding": [ { "__count": "#no"} ] }
```

csvserver_domain_binding

Binding object showing the domain that can be bound to csvserver.

Properties

(click to see Operations)

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
backupip	<String>	Read-write	. Minimum length = 1
ttl	<Double>	Read-write	. Minimum value = 1
name	<String>	Read-write	Name of the content switching virtual server to which the content switching policy applies. Minimum length = 1
domainname	<String>	Read-write	Domain name for which to change the time to live (TTL) and/or backup service IP address. Minimum length = 1
sitedomainttl	<Double>	Read-write	. Minimum value = 1
cookiedomain	<String>	Read-write	. Minimum length = 1
cookietimeout	<Double>	Read-write	.

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
			Minimum value = 0 Maximum value = 1440
appflowlog	<String>	Read-only	Enable logging appflow flow information. Default value: ENABLED Possible values = ENABLED, DISABLED
__count	<Double>	Read-only	count parameter

Operations

(click to see Properties)

ADD:| DELETE:| GET| GET (ALL)| COUNT

Some options that you can use for each operations:

- **Getting warnings in response:**NITRO allows you to get warnings in an operation by specifying the "warning" query parameter as "yes". For example, to get warnings while connecting to the Citrix ADC appliance, the URL is as follows:

  http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/login?warning=yes

  If any, the warnings are displayed in the response payload with the HTTP code "209 X-NITRO-WARNING".

- **Authenticated access for individual NITRO operations:**NITRO allows you to logon to the Citrix ADC appliance to perform individual operations. You can use this option instead of creating a NITRO session (using the login object) and then using that session to perform all operations,

  To do this, you must specify the username and password in the request header of the NITRO request as follows:

X-NITRO-USER:<username>

X-NITRO-PASS:<password>

Note: In such cases, make sure that the request header DOES not include the following:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue>

`*Note:`*

Mandatory parameters are marked in redand placeholder content is marked in <green>.

add

URL:http://<Citrix-ADC-IP-address(NSIP)/nitro/v1/config/csvserver_domain_binding

HTTP Method:PUT

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Content-Type:application/json

Request Payload:

```
{

"csvserver_domain_binding":{

<b>"name":<String_value>,

</b>"domainname":<String_value>,

"ttl":<Double_value>,

"backupip":<String_value>,

"cookiedomain":<String_value>,

"cookietimeout":<Double_value>,

"sitedomainttl":<Double_value>

}}
```

Response:

HTTP Status Code on Success: 201 Created HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error

delete

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/csvserver_domain_binding/name_value<String>

Query-parameters:

args

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/csvserver_domain_binding/name_value<String>?**args=domainname:<String_value>**

HTTP Method:DELETE

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue>

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error

get

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/csvserver_domain_binding/name_value<String>

Query-parameters:

filter

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/csvserver_domain_binding/name_value<String>?**filter=property-name1:property-value1,property-name2:property-value2**

Use this query-parameter to get the filtered set of csvserver_domain_binding resources configured on Citrix ADC. Filtering can be done on any of the properties of the resource.

pagination

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/csvserver_domain_binding/name_value<String>?**pagesize=#no;pageno=#no**

Use this query-parameter to get the csvserver_domain_binding resources in chunks.

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "csvserver_domain_binding": [ {

"backupip":<String_value>,

"ttl":<Double_value>,

"name":<String_value>,

"domainname":<String_value>,

"sitedomainttl":<Double_value>,

"cookiedomain":<String_value>,

"cookietimeout":<Double_value>,
```

```
"appflowlog":<String_value>

}]}
```

get (all)

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/csvserver_domain_binding

Query-parameters:

bulkbindings

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/csvserver_domain_binding?**bulkbindings=yes**

NITRO allows you to fetch bindings in bulk.

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "csvserver_domain_binding": [ {

"backupip":<String_value>,

"ttl":<Double_value>,

"name":<String_value>,
```

```
"domainname":<String_value>,

"sitedomainttl":<Double_value>,

"cookiedomain":<String_value>,

"cookietimeout":<Double_value>,

"appflowlog":<String_value>

}]}
```

count

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/csvserver_domain_binding/name_value<String>?**count=yes**

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{"csvserver_domain_binding": [ { "__count": "#no"} ] }
```

csvserver_feopolicy_binding

Binding object showing the feopolicy that can be bound to csvserver.

Properties

(click to see Operations)

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
priority	<Double>	Read-write	Priority for the policy.
bindpoint	<String>	Read-write	The bindpoint to which the policy is bound. Possible values = REQUEST, RESPONSE, ICA_REQUEST, OTHERTCP_REQUEST
policyname	<String>	Read-write	Policies bound to this vserver.
labelname	<String>	Read-write	Name of the label to be invoked.
name	<String>	Read-write	Name of the content switching virtual server to which the content switching policy applies. Minimum length = 1
targetlbvserver	<String>	Read-write	Name of the Load Balancing virtual server to which the content is switched, if policy rule is evaluated to be TRUE. Example: bind cs vs cs1 -policyname pol1 -priority 101 -targetLBVserver lb1 Note: Use this

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
			parameter only in case of Content Switching policy bind operations to a CS vserver. Minimum length = 1
gotopriorityexpression	<String>	Read-write	Expression specifying the priority of the next policy which will get evaluated if the current policy rule evaluates to TRUE.
invoke	<Boolean>	Read-write	Invoke a policy label if this policy's rule evaluates to TRUE (valid only for default-syntax policies such as application firewall, transform, integrated cache, rewrite, responder, and content switching).
labeltype	<String>	Read-write	Type of label to be invoked. Possible values = reqvserver, resvserver, policylabel
__count	<Double>	Read-write	count parameter

Operations

(click to see Properties)

ADD:| DELETE:| GET| GET (ALL)| COUNT

Some options that you can use for each operations:

- **Getting warnings in response:**NITRO allows you to get warnings in an operation by specifying the "warning" query parameter as "yes". For example, to get warnings while connecting to the Citrix ADC appliance, the URL is as follows:

  http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/login?warning=yes

  If any, the warnings are displayed in the response payload with the HTTP code "209 X-NITRO-WARNING".

- **Authenticated access for individual NITRO operations:**NITRO allows you to logon to the Citrix ADC appliance to perform individual operations. You can use this option instead of creating a NITRO session (using the login object) and then using that session to perform all operations,

  To do this, you must specify the username and password in the request header of the NITRO request as follows:

  X-NITRO-USER:<username>

  X-NITRO-PASS:<password>

  Note: In such cases, make sure that the request header DOES not include the following:

  Cookie:NITRO_AUTH_TOKEN=<tokenvalue>

`*Note:`*

Mandatory parameters are marked in redand placeholder content is marked in <green>.

add

URL:http://<Citrix-ADC-IP-address(NSIP)/nitro/v1/config/csvserver_feopolicy_binding

HTTP Method:PUT

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Content-Type:application/json

Request Payload:

```
{

"csvserver_feopolicy_binding":{

<b>"name":<String_value>,

</b>"policyname":<String_value>,

"targetlbvserver":<String_value>,

"priority":<Double_value>,

"gotopriorityexpression":<String_value>,

"bindpoint":<String_value>,

"invoke":<Boolean_value>,

"labeltype":<String_value>,

"labelname":<String_value>

}}
```

Response:

HTTP Status Code on Success: 201 Created HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error

delete

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/csvserver_feopolicy_binding/name_value<String>

Query-parameters:

args

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/csvserver_feopolicy_binding/name_value<String>?**args=policyname:<String_value>,bindpoint:<String_value>,priority:<Double_value>**

HTTP Method:DELETE

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue>

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error

get

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/csvserver_feopolicy_binding/name_value<String>

Query-parameters:

filter

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/csvserver_feopolicy_binding/name_value<String>?**filter=property-name1:property-value1,property-name2:property-value2**

Use this query-parameter to get the filtered set of csvserver_feopolicy_binding resources configured on Citrix ADC. Filtering can be done on any of the properties of the resource.

pagination

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/csvserver_feopolicy_binding/name_value<String>?**pagesize=#no;pageno=#no**

Use this query-parameter to get the csvserver_feopolicy_binding resources in chunks.

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "csvserver_feopolicy_binding": [ {

"priority":<Double_value>,

"bindpoint":<String_value>,

"policyname":<String_value>,

"labelname":<String_value>,

"name":<String_value>,

"targetlbvserver":<String_value>,

"gotopriorityexpression":<String_value>,

"invoke":<Boolean_value>,

"labeltype":<String_value>

}]}
```

get (all)

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/csvserver_feopolicy_binding

Query-parameters:

bulkbindings

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/csvserver_feopolicy_binding?**bulkbindings=yes**

NITRO allows you to fetch bindings in bulk.

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "csvserver_feopolicy_binding": [ {

"priority":<Double_value>,

"bindpoint":<String_value>,

"policyname":<String_value>,

"labelname":<String_value>,

"name":<String_value>,

"targetlbvserver":<String_value>,

"gotopriorityexpression":<String_value>,

"invoke":<Boolean_value>,

"labeltype":<String_value>

}]}
```

count

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/csvserver_feopolicy_binding/name_value<String>?**count=yes**

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{"csvserver_feopolicy_binding": [ { "__count": "#no"} ] }
```

csvserver_filterpolicy_binding

Binding object showing the filterpolicy that can be bound to csvserver.

Properties

(click to see Operations)

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
priority	<Double>	Read-write	Priority for the policy.
bindpoint	<String>	Read-write	For a rewrite policy, the bind point to which to bind the policy. Note: This parameter applies only to rewrite

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
			policies, because content switching policies are evaluated only at request time. Possible values = REQUEST, RESPONSE, ICA_REQUEST, OTHERTCP_REQUEST
policyname	<String>	Read-write	Policies bound to this vserver.
labelname	<String>	Read-write	Name of the label to be invoked.
name	<String>	Read-write	Name of the content switching virtual server to which the content switching policy applies. Minimum length = 1
gotopriorityexpression	<String>	Read-write	Expression or other value specifying the next policy to be evaluated if the current policy evaluates to TRUE. Specify one of the following values: * NEXT - Evaluate the policy with the next higher priority number. * END - End policy evaluation. * USE_INVOCATION_RESULT - Applicable

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
			if this policy invokes another policy label. If the final goto in the invoked policy label has a value of END, the evaluation stops. If the final goto is anything other than END, the current policy label performs a NEXT. * An expression that evaluates to a number. If you specify an expression, the number to which it evaluates determines the next policy to evaluate, as follows: * If the expression evaluates to a higher numbered priority, the policy with that priority is evaluated next. * If the expression evaluates to the priority of the current policy, the policy with the next higher numbered priority is evaluated next. * If the expression evaluates to a priority number that is numerically higher than the highest numbered priority, policy evaluation ends. An

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
			UNDEF event is triggered if: * The expression is invalid. * The expression evaluates to a priority number that is numerically lower than the current policy's priority. * The expression evaluates to a priority number that is between the current policy's priority number (say, 30) and the highest priority number (say, 100), but does not match any configured priority number (for example, the expression evaluates to the number 85). This example assumes that the priority number increments by 10 for every successive policy, and therefore a priority number of 85 does not exist in the policy label.
targetlbvserver	<String>	Read-write	Name of the Load Balancing virtual server to which the content is switched, if policy rule is evaluated to be TRUE. Example: bind cs vs cs1

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
			-policyname pol1 -priority 101 -targetLBVserver lb1 Note: Use this parameter only in case of Content Switching policy bind operations to a CS vserver. Minimum length = 1
invoke	<Boolean>	Read-write	Invoke a policy label if this policy's rule evaluates to TRUE (valid only for default-syntax policies such as application firewall, transform, integrated cache, rewrite, responder, and content switching).
labeltype	<String>	Read-write	Type of label to be invoked. Possible values = reqvserver, resvserver, policylabel
sc	<String>	Read-only	The state of SureConnect the specified virtual server. Possible values = ON, OFF
__count	<Double>	Read-only	count parameter

Operations

(click to see Properties)

ADD:| DELETE:| GET| GET (ALL)| COUNT

Some options that you can use for each operations:

- **Getting warnings in response:**NITRO allows you to get warnings in an operation by specifying the "warning" query parameter as "yes". For example, to get warnings while connecting to the Citrix ADC appliance, the URL is as follows:

  http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/login?warning=yes

  If any, the warnings are displayed in the response payload with the HTTP code "209 X-NITRO-WARNING".

- **Authenticated access for individual NITRO operations:**NITRO allows you to logon to the Citrix ADC appliance to perform individual operations. You can use this option instead of creating a NITRO session (using the login object) and then using that session to perform all operations,

  To do this, you must specify the username and password in the request header of the NITRO request as follows:

  X-NITRO-USER:<username>

  X-NITRO-PASS:<password>

  Note: In such cases, make sure that the request header DOES not include the following:

  Cookie:NITRO_AUTH_TOKEN=<tokenvalue>

`*Note:`*

Mandatory parameters are marked in redand placeholder content is marked in <green>.

add

URL:http://<Citrix-ADC-IP-address(NSIP)/nitro/v1/config/csvserver_filterpolicy_binding

HTTP Method:PUT

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Content-Type:application/json

Request Payload:

```
{
"csvserver_filterpolicy_binding":{
<b>"name":<String_value>,
</b>"policyname":<String_value>,
"targetlbvserver":<String_value>,
"priority":<Double_value>,
"gotopriorityexpression":<String_value>,
"bindpoint":<String_value>,
"invoke":<Boolean_value>,
"labeltype":<String_value>,
"labelname":<String_value>
}}
```

Response:

HTTP Status Code on Success: 201 Created HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error

delete

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/csvserver_filterpolicy_binding/name_value<String>

Query-parameters:

args

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/csvserver_filterpolicy_binding/name_value<String>?**args=policyname:<String_value>,bindpoint:<String_value>,priority:<Double_value>**

HTTP Method:DELETE

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue>

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error

get

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/csvserver_filterpolicy_binding/name_value<String>

Query-parameters:

filter

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/csvserver_filterpolicy_binding/name_value<String>?**filter=property-name1:property-value1,property-name2:property-value2**

Use this query-parameter to get the filtered set of csvserver_filterpolicy_binding resources configured on Citrix ADC. Filtering can be done on any of the properties of the resource.

pagination

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/csvserver_filterpolicy_binding/name_value<String>?**pagesize=#no;pageno=#no**

Use this query-parameter to get the csvserver_filterpolicy_binding resources in chunks.

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "csvserver_filterpolicy_binding": [ {

"priority":<Double_value>,

"bindpoint":<String_value>,

"policyname":<String_value>,

"labelname":<String_value>,

"name":<String_value>,

"gotopriorityexpression":<String_value>,

"targetlbvserver":<String_value>,

"invoke":<Boolean_value>,

"labeltype":<String_value>,

"sc":<String_value>

}]}
```

get (all)

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/csvserver_filterpolicy_binding

Query-parameters:

bulkbindings

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/csvserver_filterpolicy_binding?**bulkbindings=yes**

NITRO allows you to fetch bindings in bulk.

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "csvserver_filterpolicy_binding": [ {

"priority":<Double_value>,

"bindpoint":<String_value>,

"policyname":<String_value>,

"labelname":<String_value>,

"name":<String_value>,

"gotopriorityexpression":<String_value>,

"targetlbvserver":<String_value>,

"invoke":<Boolean_value>,

"labeltype":<String_value>,

"sc":<String_value>
```

```
}]}
```

count

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/csvserver_filterpolicy_binding/name_value<String>?**count=yes**

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{"csvserver_filterpolicy_binding": [ { "__count": "#no"} ] }
```

csvserver_gslbvserver_binding

Binding object showing the gslbvserver that can be bound to csvserver.

Properties

(click to see Operations)

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
vserver	<String>	Read-write	Name of the default gslb or vpn vserver bound to CS vserver of type GSLB/VPN.

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
			For Example: bind cs vserver cs1 -vserver gslb1 or bind cs vserver cs1 -vserver vpn1. Minimum length = 1
name	<String>	Read-write	Name of the content switching virtual server to which the content switching policy applies. Minimum length = 1
hits	<Double>	Read-only	Number of hits.
__count	<Double>	Read-only	count parameter

Operations

(click to see Properties)

ADD:| DELETE:| GET| GET (ALL)| COUNT

Some options that you can use for each operations:

- **Getting warnings in response:**NITRO allows you to get warnings in an operation by specifying the "warning" query parameter as "yes". For example, to get warnings while connecting to the Citrix ADC appliance, the URL is as follows:

  http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/login?warning=yes

  If any, the warnings are displayed in the response payload with the HTTP code "209 X-NITRO-WARNING".

- **Authenticated access for individual NITRO operations:**NITRO allows you to logon to the Citrix ADC appliance to perform individual operations. You can use this option instead of creating a NITRO session (using the login object) and then using that session to perform all operations,

  To do this, you must specify the username and password in the request header of the NITRO request as follows:

X-NITRO-USER:<username>

X-NITRO-PASS:<password>

Note: In such cases, make sure that the request header DOES not include the following:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue>

`*Note:`*

Mandatory parameters are marked in redand placeholder content is marked in <green>.

add

URL:http://<Citrix-ADC-IP-address(NSIP)/nitro/v1/config/csvserver_gslbvserver_binding

HTTP Method:PUT

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Content-Type:application/json

Request Payload:

```
{

"csvserver_gslbvserver_binding":{

<b>"name":<String_value>,

</b>"vserver":<String_value>

}}
```

Response:

HTTP Status Code on Success: 201 Created HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error

delete

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/csvserver_gslbvserver_binding/name_value<String>

Query-parameters:

args

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/csvserver_gslbvserver_binding/name_value<String>?**args=vserver:<String_value>**

HTTP Method:DELETE

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue>

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error

get

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/csvserver_gslbvserver_binding/name_value<String>

Query-parameters:

filter

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/csvserver_gslbvserver_binding/name_value<String>?**filter=property-name1:property-value1,property-name2:property-value2**

Use this query-parameter to get the filtered set of csvserver_gslbvserver_binding resources configured on Citrix ADC. Filtering can be done on any of the properties of the resource.

pagination

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/csvserver_gslbvserver_binding/name_value<String>?**pagesize=#no;pageno=#no**

Use this query-parameter to get the csvserver_gslbvserver_binding resources in chunks.

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "csvserver_gslbvserver_binding": [ {

"vserver":<String_value>,

"name":<String_value>,

"hits":<Double_value>

}]}
```

get (all)

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/csvserver_gslbvserver_binding

Query-parameters:

bulkbindings

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/csvserver_gslbvserver_binding?**bulkbindings=yes**

NITRO allows you to fetch bindings in bulk.

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "csvserver_gslbvserver_binding": [ {

"vserver":<String_value>,

"name":<String_value>,

"hits":<Double_value>

}]}
```

count

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/csvserver_gslbvserver_binding/name_value<String>?**count=yes**

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{"csvserver_gslbvserver_binding": [ { "__count": "#no"} ] }
```

csvserver_lbvserver_binding

Binding object showing the lbvserver that can be bound to csvserver.

Properties

(click to see Operations)

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
name	<String>	Read-write	Name of the content switching virtual server to which the content switching policy applies. Minimum length = 1
targetvserver	<String>	Read-write	The virtual server name (created with the add lb vserver command) to which content will be switched. Minimum length = 1
lbvserver	<String>	Read-write	Name of the default lb vserver bound. Use this param for Default binding only. For Example: bind cs vserver cs1 -lbvserver lb1. Minimum length = 1
hits	<Double>	Read-only	Number of hits.

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
vserverid	<String>	Read-only	Vserver Id of vserver.
cookieipport	<String>	Read-only	Vserver id of the lb vserver that is inserted into the set-cookie HTTP header.
__count	<Double>	Read-only	count parameter

Operations

(click to see Properties)

ADD:| DELETE:| GET| GET (ALL)| COUNT

Some options that you can use for each operations:

- **Getting warnings in response:**NITRO allows you to get warnings in an operation by specifying the "warning" query parameter as "yes". For example, to get warnings while connecting to the Citrix ADC appliance, the URL is as follows:

  http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/login?warning=yes

  If any, the warnings are displayed in the response payload with the HTTP code "209 X-NITRO-WARNING".

- **Authenticated access for individual NITRO operations:**NITRO allows you to logon to the Citrix ADC appliance to perform individual operations. You can use this option instead of creating a NITRO session (using the login object) and then using that session to perform all operations,

  To do this, you must specify the username and password in the request header of the NITRO request as follows:

  X-NITRO-USER:<username>

  X-NITRO-PASS:<password>

  Note: In such cases, make sure that the request header DOES not include the following:

  Cookie:NITRO_AUTH_TOKEN=<tokenvalue>

`*Note:`*

Mandatory parameters are marked in redand placeholder content is marked in <green>.

add

URL:http://<Citrix-ADC-IP-address(NSIP)/nitro/v1/config/csvserver_lbvserver_binding

HTTP Method:PUT

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Content-Type:application/json

Request Payload:

```
{

"csvserver_lbvserver_binding":{

<b>"name":<String_value>,

</b>"lbvserver":<String_value>,

"targetvserver":<String_value>

}}
```

Response:

HTTP Status Code on Success: 201 Created HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error

delete

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/csvserver_lbvserver_binding/name_value<String>

Query-parameters:

args

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/csvserver_lbvserver_binding/name_value<String>?**args=lbvserver:<String_value>**

HTTP Method:DELETE

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue>

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error

get

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/csvserver_lbvserver_binding/name_value<String>

Query-parameters:

filter

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/csvserver_lbvserver_binding/name_value<String>?**filter=property-name1:property-value1,property-name2:property-value2**

Use this query-parameter to get the filtered set of csvserver_lbvserver_binding resources configured on Citrix ADC. Filtering can be done on any of the properties of the resource.

pagination

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/csvserver_lbvserver_binding/name_value<String>?**pagesize=#no;pageno=#no**

Use this query-parameter to get the csvserver_lbvserver_binding resources in chunks.

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "csvserver_lbvserver_binding": [ {

"name":<String_value>,

"targetvserver":<String_value>,

"lbvserver":<String_value>,

"hits":<Double_value>,

"vserverid":<String_value>,

"cookieipport":<String_value>

}]}
```

get (all)

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/csvserver_lbvserver_binding

Query-parameters:

bulkbindings

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/csvserver_lbvserver_binding?**bulkbindings=yes**

NITRO allows you to fetch bindings in bulk.

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "csvserver_lbvserver_binding": [ {

"name":<String_value>,

"targetvserver":<String_value>,

"lbvserver":<String_value>,

"hits":<Double_value>,

"vserverid":<String_value>,

"cookieipport":<String_value>

}]}
```

count

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/csvserver_lbvserver_binding/name_value<String>?**count=yes**

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{"csvserver_lbvserver_binding": [ { "__count": "#no"} ] }
```

csvserver_responderpolicy_binding

Binding object showing the responderpolicy that can be bound to csvserver.

Properties

(click to see Operations)

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
priority	<Double>	Read-write	Priority for the policy.
bindpoint	<String>	Read-write	For a rewrite policy, the bind point to which to bind the policy. Note: This parameter applies only to rewrite policies, because content switching policies are evaluated only at request time. Possible values = REQUEST, RESPONSE, ICA_REQUEST, OTHERTCP_REQUEST

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
policyname	<String>	Read-write	Policies bound to this vserver.
labelname	<String>	Read-write	Name of the label invoked.
name	<String>	Read-write	Name of the content switching virtual server to which the content switching policy applies. Minimum length = 1
targetlbvserver	<String>	Read-write	Name of the Load Balancing virtual server to which the content is switched, if policy rule is evaluated to be TRUE. Example: bind cs vs cs1 -policyname pol1 -priority 101 -targetLBVserver lb1 Note: Use this parameter only in case of Content Switching policy bind operations to a CS vserver. Minimum length = 1
gotopriorityexpression	<String>	Read-write	Expression specifying the priority of the next policy which will get evaluated if the current policy rule evaluates to TRUE.

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
invoke	<Boolean>	Read-write	Invoke flag.
labeltype	<String>	Read-write	The invocation type. Possible values = reqvserver, resvserver, policylabel
__count	<Double>	Read-write	count parameter

Operations

(click to see Properties)

ADD:| DELETE:| GET| GET (ALL)| COUNT

Some options that you can use for each operations:

- **Getting warnings in response:**NITRO allows you to get warnings in an operation by specifying the "warning" query parameter as "yes". For example, to get warnings while connecting to the Citrix ADC appliance, the URL is as follows:

  http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/login?warning=yes

  If any, the warnings are displayed in the response payload with the HTTP code "209 X-NITRO-WARNING".

- **Authenticated access for individual NITRO operations:**NITRO allows you to logon to the Citrix ADC appliance to perform individual operations. You can use this option instead of creating a NITRO session (using the login object) and then using that session to perform all operations,

  To do this, you must specify the username and password in the request header of the NITRO request as follows:

  X-NITRO-USER:<username>

  X-NITRO-PASS:<password>

  Note: In such cases, make sure that the request header DOES not include the following:

  Cookie:NITRO_AUTH_TOKEN=<tokenvalue>

`*Note:` *

Mandatory parameters are marked in redand placeholder content is marked in <green>.

add

URL:http://<Citrix-ADC-IP-address(NSIP)/nitro/v1/config/csvserver_responderpolicy_binding

HTTP Method:PUT

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Content-Type:application/json

Request Payload:

```
{

"csvserver_responderpolicy_binding":{

<b>"name":<String_value>,

</b>"policyname":<String_value>,

"targetlbvserver":<String_value>,

"priority":<Double_value>,

"gotopriorityexpression":<String_value>,

"bindpoint":<String_value>,

"invoke":<Boolean_value>,

"labeltype":<String_value>,

"labelname":<String_value>

}}
```

Response:

HTTP Status Code on Success: 201 Created HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error

delete

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/csvserver_responderpolicy_binding/name_value<String>

Query-parameters:

args

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/csvserver_responderpolicy_binding/name_value<String>?**args=policyname:<String_value>,bindpoint:<String_value>,priority:<Double_value>**

HTTP Method:DELETE

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue>

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error

get

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/csvserver_responderpolicy_binding/name_value<String>

Query-parameters:

filter

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/csvserver_responderpolicy_binding/name_value<String>?**filter=property-name1:property-value1,property-name2:property-value2**

Use this query-parameter to get the filtered set of csvserver_responderpolicy_binding resources configured on Citrix ADC. Filtering can be done on any of the properties of the resource.

pagination

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/csvserver_responderpolicy_binding/name_value<String>?**pagesize=#no;pageno=#no**

Use this query-parameter to get the csvserver_responderpolicy_binding resources in chunks.

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "csvserver_responderpolicy_binding": [ {

"priority":<Double_value>,

"bindpoint":<String_value>,

"policyname":<String_value>,

"labelname":<String_value>,

"name":<String_value>,

"targetlbvserver":<String_value>,

"gotopriorityexpression":<String_value>,

"invoke":<Boolean_value>,

"labeltype":<String_value>

}]}
```

get (all)

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/csvserver_responderpolicy_binding

Query-parameters:

bulkbindings

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/csvserver_responderpolicy_binding?**bulkbindings=yes**

NITRO allows you to fetch bindings in bulk.

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "csvserver_responderpolicy_binding": [ {

"priority":<Double_value>,

"bindpoint":<String_value>,

"policyname":<String_value>,

"labelname":<String_value>,

"name":<String_value>,
```

```
"targetlbvserver":<String_value>,

"gotopriorityexpression":<String_value>,

"invoke":<Boolean_value>,

"labeltype":<String_value>

}]}
```

count

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/csvserver_responderpolicy_binding/name_value<String>?**count=yes**

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{"csvserver_responderpolicy_binding": [ { "__count": "#no"} ] }
```

csvserver_rewritepolicy_binding

Binding object showing the rewritepolicy that can be bound to csvserver.

Properties

(click to see Operations)

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
priority	<Double>	Read-write	Priority for the policy.
bindpoint	<String>	Read-write	The bindpoint to which the policy is bound. Possible values = REQUEST, RESPONSE, ICA_REQUEST, OTHERTCP_REQUEST
policyname	<String>	Read-write	Policies bound to this vserver.
labelname	<String>	Read-write	Name of the label invoked.
name	<String>	Read-write	Name of the content switching virtual server to which the content switching policy applies. Minimum length = 1
targetlbvserver	<String>	Read-write	Name of the Load Balancing virtual server to which the content is switched, if policy rule is evaluated to be TRUE. Example: bind cs vs cs1 -policyname pol1 -priority 101 -targetLBVserver lb1 Note: Use this

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
			parameter only in case of Content Switching policy bind operations to a CS vserver. Minimum length = 1
gotopriorityexpression	<String>	Read-write	Expression specifying the priority of the next policy which will get evaluated if the current policy rule evaluates to TRUE.
invoke	<Boolean>	Read-write	Invoke flag.
labeltype	<String>	Read-write	The invocation type. Possible values = reqvserver, resvserver, policylabel
__count	<Double>	Read-write	count parameter

Operations

(click to see Properties)

ADD:| DELETE:| GET| GET (ALL)| COUNT

Some options that you can use for each operations:

- **Getting warnings in response:**NITRO allows you to get warnings in an operation by specifying the "warning" query parameter as "yes". For example, to get warnings while connecting to the Citrix ADC appliance, the URL is as follows:

  http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/login?warning=yes

  If any, the warnings are displayed in the response payload with the HTTP code "209 X-NITRO-WARNING".

- **Authenticated access for individual NITRO operations:**NITRO allows you to logon to the Citrix ADC appliance to perform individual operations. You can use this option instead of creating a NITRO session (using the login object) and then using that session to perform all operations,

  To do this, you must specify the username and password in the request header of the NITRO request as follows:

  X-NITRO-USER:<username>

  X-NITRO-PASS:<password>

  Note: In such cases, make sure that the request header DOES not include the following:

  Cookie:NITRO_AUTH_TOKEN=<tokenvalue>

`*Note:`*

Mandatory parameters are marked in redand placeholder content is marked in <green>.

add

URL:http://<Citrix-ADC-IP-address(NSIP)/nitro/v1/config/csvserver_rewritepolicy_binding

HTTP Method:PUT

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Content-Type:application/json

Request Payload:

```
{

"csvserver_rewritepolicy_binding":{

<b>"name":<String_value>,

</b>"policyname":<String_value>,

"targetlbvserver":<String_value>,

"priority":<Double_value>,
```

```
"gotopriorityexpression":<String_value>,

"bindpoint":<String_value>,

"invoke":<Boolean_value>,

"labeltype":<String_value>,

"labelname":<String_value>

}}
```

Response:

HTTP Status Code on Success: 201 Created HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error

delete

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/csvserver_rewritepolicy_binding/name_value<String>

Query-parameters:

args

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/csvserver_rewritepolicy_binding/name_value<String>?
args=policyname:<String_value>,bindpoint:<String_value>,priority:<Double_value>

HTTP Method:DELETE

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue>

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error

get

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/csvserver_rewritepolicy_binding/name_value<String>

Query-parameters:

filter

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/csvserver_rewritepolicy_binding/name_value<String>?**filter=property-name1:property-value1,property-name2:property-value2**

Use this query-parameter to get the filtered set of csvserver_rewritepolicy_binding resources configured on Citrix ADC. Filtering can be done on any of the properties of the resource.

pagination

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/csvserver_rewritepolicy_binding/name_value<String>?**pagesize=#no;pageno=#no**

Use this query-parameter to get the csvserver_rewritepolicy_binding resources in chunks.

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "csvserver_rewritepolicy_binding": [ {

"priority":<Double_value>,

"bindpoint":<String_value>,
```

```
"policyname":<String_value>,

"labelname":<String_value>,

"name":<String_value>,

"targetlbvserver":<String_value>,

"gotopriorityexpression":<String_value>,

"invoke":<Boolean_value>,

"labeltype":<String_value>

}]}
```

get (all)

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/csvserver_rewritepolicy_binding

Query-parameters:

bulkbindings

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/csvserver_rewritepolicy_binding?**bulkbindings=yes**

NITRO allows you to fetch bindings in bulk.

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "csvserver_rewritepolicy_binding": [ {

"priority":<Double_value>,

"bindpoint":<String_value>,

"policyname":<String_value>,

"labelname":<String_value>,

"name":<String_value>,

"targetlbvserver":<String_value>,

"gotopriorityexpression":<String_value>,

"invoke":<Boolean_value>,

"labeltype":<String_value>

}]}
```

count

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/csvserver_rewritepolicy_binding/name_value<String>?**count=yes**

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{"csvserver_rewritepolicy_binding": [ { "__count": "#no"} ] }
```

csvserver_spilloverpolicy_binding

Binding object showing the spilloverpolicy that can be bound to csvserver.

Properties

(click to see Operations)

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
priority	<Double>	Read-write	Priority for the policy.
bindpoint	<String>	Read-write	The bindpoint to which the policy is bound. Possible values = REQUEST, RESPONSE, ICA_REQUEST, OTHERTCP_REQUEST
policyname	<String>	Read-write	Policies bound to this vserver.
labelname	<String>	Read-write	Name of the label to be invoked.
name	<String>	Read-write	Name of the content switching virtual server to which the content switching policy applies.

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
			Minimum length = 1
targetlbvserver	<String>	Read-write	Name of the Load Balancing virtual server to which the content is switched, if policy rule is evaluated to be TRUE. Example: bind cs vs cs1 -policyname pol1 -priority 101 -targetLBVserver lb1 Note: Use this parameter only in case of Content Switching policy bind operations to a CS vserver. Minimum length = 1
gotopriorityexpression	<String>	Read-write	Expression specifying the priority of the next policy which will get evaluated if the current policy rule evaluates to TRUE.
invoke	<Boolean>	Read-write	Invoke a policy label if this policy's rule evaluates to TRUE (valid only for default-syntax policies such as application firewall, transform, integrated cache, rewrite, responder, and content switching).

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
labeltype	<String>	Read-write	Type of label to be invoked. Possible values = reqvserver, resvserver, policylabel
__count	<Double>	Read-write	count parameter

Operations

(click to see Properties)

ADD:| DELETE:| GET| GET (ALL)| COUNT

Some options that you can use for each operations:

- **Getting warnings in response:**NITRO allows you to get warnings in an operation by specifying the "warning" query parameter as "yes". For example, to get warnings while connecting to the Citrix ADC appliance, the URL is as follows:

  http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/login?warning=yes

  If any, the warnings are displayed in the response payload with the HTTP code "209 X-NITRO-WARNING".

- **Authenticated access for individual NITRO operations:**NITRO allows you to logon to the Citrix ADC appliance to perform individual operations. You can use this option instead of creating a NITRO session (using the login object) and then using that session to perform all operations,

  To do this, you must specify the username and password in the request header of the NITRO request as follows:

  X-NITRO-USER:<username>

  X-NITRO-PASS:<password>

  Note: In such cases, make sure that the request header DOES not include the following:

  Cookie:NITRO_AUTH_TOKEN=<tokenvalue>

`*Note:` *

Mandatory parameters are marked in redand placeholder content is marked in <green>.

add

URL:http://<Citrix-ADC-IP-address(NSIP)/nitro/v1/config/csvserver_spilloverpolicy_binding

HTTP Method:PUT

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Content-Type:application/json

Request Payload:

```
{

"csvserver_spilloverpolicy_binding":{

<b>"name":<String_value>,

</b>"policyname":<String_value>,

"targetlbvserver":<String_value>,

"priority":<Double_value>,

"gotopriorityexpression":<String_value>,

"bindpoint":<String_value>,

"invoke":<Boolean_value>,

"labeltype":<String_value>,

"labelname":<String_value>

}}
```

Response:

HTTP Status Code on Success: 201 Created HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error

delete

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/csvserver_spilloverpolicy_binding/name_value<String>

Query-parameters:

args

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/csvserver_spilloverpolicy_binding/name_value<String>?**args=policyname:<String_value>,bindpoint:<String_value>,priority:<Double_value>**

HTTP Method:DELETE

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue>

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error

get

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/csvserver_spilloverpolicy_binding/name_value<String>

Query-parameters:

filter

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/csvserver_spilloverpolicy_binding/name_value<String>?**filter=property-name1:property-value1,property-name2:property-value2**

Use this query-parameter to get the filtered set of csvserver_spilloverpolicy_binding resources configured on Citrix ADC. Filtering can be done on any of the properties of the resource.

pagination

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/csvserver_spilloverpolicy_binding/name_value<String>?**pagesize=#no;pageno=#no**

Use this query-parameter to get the csvserver_spilloverpolicy_binding resources in chunks.

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "csvserver_spilloverpolicy_binding": [ {

"priority":<Double_value>,

"bindpoint":<String_value>,

"policyname":<String_value>,

"labelname":<String_value>,

"name":<String_value>,

"targetlbvserver":<String_value>,

"gotopriorityexpression":<String_value>,

"invoke":<Boolean_value>,

"labeltype":<String_value>

}]}
```

get (all)

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/csvserver_spilloverpolicy_binding

Query-parameters:

bulkbindings

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/csvserver_spilloverpolicy_binding?**bulkbindings=yes**

NITRO allows you to fetch bindings in bulk.

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "csvserver_spilloverpolicy_binding": [ {

"priority":<Double_value>,

"bindpoint":<String_value>,

"policyname":<String_value>,

"labelname":<String_value>,

"name":<String_value>,
```

```
"targetlbvserver":<String_value>,

"gotopriorityexpression":<String_value>,

"invoke":<Boolean_value>,

"labeltype":<String_value>

}]}
```

count

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/csvserver_spilloverpolicy_binding/name_value<String>?**count=yes**

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{"csvserver_spilloverpolicy_binding": [ { "__count": "#no"} ] }
```

csvserver_tmtrafficpolicy_binding

Binding object showing the tmtrafficpolicy that can be bound to csvserver.

Properties

(click to see Operations)

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
priority	<Double>	Read-write	Priority for the policy.
bindpoint	<String>	Read-write	For a rewrite policy, the bind point to which to bind the policy. Note: This parameter applies only to rewrite policies, because content switching policies are evaluated only at request time. Possible values = REQUEST, RESPONSE, ICA_REQUEST, OTHERTCP_REQUEST
policyname	<String>	Read-write	Policies bound to this vserver.
labelname	<String>	Read-write	Name of the label to be invoked.
name	<String>	Read-write	Name of the content switching virtual server to which the content switching policy applies. Minimum length = 1
gotopriorityexpression	<String>	Read-write	Expression or other value specifying the next policy to be evaluated if the

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
			current policy evaluates to TRUE. Specify one of the following values: * NEXT - Evaluate the policy with the next higher priority number. * END - End policy evaluation. * USE_INVOCATION_RESULT - Applicable if this policy invokes another policy label. If the final goto in the invoked policy label has a value of END, the evaluation stops. If the final goto is anything other than END, the current policy label performs a NEXT. * An expression that evaluates to a number. If you specify an expression, the number to which it evaluates determines the next policy to evaluate, as follows: * If the expression evaluates to a higher numbered priority, the policy with that priority is evaluated next. * If the expression evaluates to the priority of the current policy, the

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
			policy with the next higher numbered priority is evaluated next. * If the expression evaluates to a priority number that is numerically higher than the highest numbered priority, policy evaluation ends. An UNDEF event is triggered if: * The expression is invalid. * The expression evaluates to a priority number that is numerically lower than the current policy's priority. * The expression evaluates to a priority number that is between the current policy's priority number (say, 30) and the highest priority number (say, 100), but does not match any configured priority number (for example, the expression evaluates to the number 85). This example assumes that the priority number increments by 10 for every successive policy, and therefore a priority number of

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
			85 does not exist in the policy label.
targetlbvserver	<String>	Read-write	Name of the Load Balancing virtual server to which the content is switched, if policy rule is evaluated to be TRUE. Example: bind cs vs cs1 -policyname pol1 -priority 101 -targetLBVserver lb1 Note: Use this parameter only in case of Content Switching policy bind operations to a CS vserver. Minimum length = 1
invoke	<Boolean>	Read-write	Invoke a policy label if this policy's rule evaluates to TRUE (valid only for default-syntax policies such as application firewall, transform, integrated cache, rewrite, responder, and content switching).
labeltype	<String>	Read-write	Type of label to be invoked. Possible values = reqvserver, resvserver, policylabel

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
__count	<Double>	Read-write	count parameter

Operations

(click to see Properties)

ADD:| DELETE:| GET| GET (ALL)| COUNT

Some options that you can use for each operations:

- **Getting warnings in response:**NITRO allows you to get warnings in an operation by specifying the "warning" query parameter as "yes". For example, to get warnings while connecting to the Citrix ADC appliance, the URL is as follows:

  http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/login?warning=yes

  If any, the warnings are displayed in the response payload with the HTTP code "209 X-NITRO-WARNING".
- **Authenticated access for individual NITRO operations:**NITRO allows you to logon to the Citrix ADC appliance to perform individual operations. You can use this option instead of creating a NITRO session (using the login object) and then using that session to perform all operations,

  To do this, you must specify the username and password in the request header of the NITRO request as follows:

  X-NITRO-USER:<username>

  X-NITRO-PASS:<password>

  Note: In such cases, make sure that the request header DOES not include the following:

  Cookie:NITRO_AUTH_TOKEN=<tokenvalue>

`*Note:`*

Mandatory parameters are marked in redand placeholder content is marked in <green>.

add

URL:http://<Citrix-ADC-IP-address(NSIP)/nitro/v1/config/csvserver_tmtrafficpolicy_binding

HTTP Method:PUT

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Content-Type:application/json

Request Payload:

```
{

"csvserver_tmtrafficpolicy_binding":{

<b>"name":<String_value>,

</b>"policyname":<String_value>,

"targetlbvserver":<String_value>,

"priority":<Double_value>,

"gotopriorityexpression":<String_value>,

"bindpoint":<String_value>,

"invoke":<Boolean_value>,

"labeltype":<String_value>,

"labelname":<String_value>

}}
```

Response:

HTTP Status Code on Success: 201 Created HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error

delete

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/csvserver_tmtrafficpolicy_binding/name_value<String>

Query-parameters:

args

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/csvserver_tmtrafficpolicy_binding/name_value<String>?**args=policyname:<String_value>,bindpoint:<String_value>,priority:<Double_value>**

HTTP Method:DELETE

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue>

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error

get

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/csvserver_tmtrafficpolicy_binding/name_value<String>

Query-parameters:

filter

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/csvserver_tmtrafficpolicy_binding/name_value<String>?**filter=property-name1:property-value1,property-name2:property-value2**

Use this query-parameter to get the filtered set of csvserver_tmtrafficpolicy_binding resources configured on Citrix ADC. Filtering can be done on any of the properties of the resource.

pagination

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/csvserver_tmtrafficpolicy_binding/name_value<String>?**pagesize=#no;pageno=#no**

Use this query-parameter to get the csvserver_tmtrafficpolicy_binding resources in chunks.

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "csvserver_tmtrafficpolicy_binding": [ {

"priority":<Double_value>,

"bindpoint":<String_value>,

"policyname":<String_value>,

"labelname":<String_value>,

"name":<String_value>,

"gotopriorityexpression":<String_value>,

"targetlbvserver":<String_value>,

"invoke":<Boolean_value>,

"labeltype":<String_value>

}]}
```

get (all)

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/csvserver_tmtrafficpolicy_binding

Query-parameters:

bulkbindings

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/csvserver_tmtrafficpolicy_binding?**bulkbindings=yes**

NITRO allows you to fetch bindings in bulk.

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "csvserver_tmtrafficpolicy_binding": [ {

"priority":<Double_value>,

"bindpoint":<String_value>,

"policyname":<String_value>,

"labelname":<String_value>,

"name":<String_value>,

"gotopriorityexpression":<String_value>,

"targetlbvserver":<String_value>,

"invoke":<Boolean_value>,

"labeltype":<String_value>

}]}
```

count

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/csvserver_tmtrafficpolicy_binding/name_value<String>?**count=yes**

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{"csvserver_tmtrafficpolicy_binding": [ { "__count": "#no"} ] }
```

csvserver_transformpolicy_binding

Binding object showing the transformpolicy that can be bound to csvserver.

Properties

(click to see Operations)

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
priority	<Double>	Read-write	Priority for the policy.
bindpoint	<String>	Read-write	The bindpoint to which the policy is bound.

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
			Possible values = REQUEST, RESPONSE, ICA_REQUEST, OTHERTCP_REQUEST
policyname	<String>	Read-write	Policies bound to this vserver.
labelname	<String>	Read-write	Name of the label invoked.
name	<String>	Read-write	Name of the content switching virtual server to which the content switching policy applies. Minimum length = 1
targetlbvserver	<String>	Read-write	Name of the Load Balancing virtual server to which the content is switched, if policy rule is evaluated to be TRUE. Example: bind cs vs cs1 -policyname pol1 -priority 101 -targetLBVserver lb1 Note: Use this parameter only in case of Content Switching policy bind operations to a CS vserver. Minimum length = 1

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
gotopriorityexpression	<String>	Read-write	Expression specifying the priority of the next policy which will get evaluated if the current policy rule evaluates to TRUE.
invoke	<Boolean>	Read-write	Invoke flag.
labeltype	<String>	Read-write	The invocation type. Possible values = reqvserver, resvserver, policylabel
__count	<Double>	Read-write	count parameter

Operations

(click to see Properties)

ADD:| DELETE:| GET| GET (ALL)| COUNT

Some options that you can use for each operations:

- **Getting warnings in response:**NITRO allows you to get warnings in an operation by specifying the "warning" query parameter as "yes". For example, to get warnings while connecting to the Citrix ADC appliance, the URL is as follows:

  http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/login?warning=yes

  If any, the warnings are displayed in the response payload with the HTTP code "209 X-NITRO-WARNING".
- **Authenticated access for individual NITRO operations:**NITRO allows you to logon to the Citrix ADC appliance to perform individual operations. You can use this option instead of creating a NITRO session (using the login object) and then using that session to perform all operations,

To do this, you must specify the username and password in the request header of the NITRO request as follows:

X-NITRO-USER:<username>

X-NITRO-PASS:<password>

Note: In such cases, make sure that the request header DOES not include the following:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue>

`*Note:`*

Mandatory parameters are marked in redand placeholder content is marked in <green>.

add

URL:http://<Citrix-ADC-IP-address(NSIP)/nitro/v1/config/csvserver_transformpolicy_binding

HTTP Method:PUT

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Content-Type:application/json

Request Payload:

```
{

"csvserver_transformpolicy_binding":{

<b>"name":<String_value>,

</b>"policyname":<String_value>,

"targetlbvserver":<String_value>,

"priority":<Double_value>,

"gotopriorityexpression":<String_value>,

"bindpoint":<String_value>,
```

```
"invoke":<Boolean_value>,

"labeltype":<String_value>,

"labelname":<String_value>

}}
```

Response:

HTTP Status Code on Success: 201 Created HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error

delete

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/csvserver_transformpolicy_binding/name_value<String>

Query-parameters:

args

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/csvserver_transformpolicy_binding/name_value<String>?
args=policyname:<String_value>,bindpoint:<String_value>,priority:<Double_value>

HTTP Method:DELETE

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue>

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error

get

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/csvserver_transformpolicy_binding/name_value<String>

Query-parameters:

filter

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/csvserver_transformpolicy_binding/name_value<String>?**filter=property-name1:property-value1,property-name2:property-value2**

Use this query-parameter to get the filtered set of csvserver_transformpolicy_binding resources configured on Citrix ADC. Filtering can be done on any of the properties of the resource.

pagination

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/csvserver_transformpolicy_binding/name_value<String>?**pagesize=#no;pageno=#no**

Use this query-parameter to get the csvserver_transformpolicy_binding resources in chunks.

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "csvserver_transformpolicy_binding": [ {

"priority":<Double_value>,

"bindpoint":<String_value>,

"policyname":<String_value>,

"labelname":<String_value>,
```

```
"name":<String_value>,

"targetlbvserver":<String_value>,

"gotopriorityexpression":<String_value>,

"invoke":<Boolean_value>,

"labeltype":<String_value>

}]}
```

get (all)

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/csvserver_transformpolicy_binding

Query-parameters:

bulkbindings

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/csvserver_transformpolicy_binding?**bulkbindings=yes**

NITRO allows you to fetch bindings in bulk.

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "csvserver_transformpolicy_binding": [ {

"priority":<Double_value>,

"bindpoint":<String_value>,

"policyname":<String_value>,

"labelname":<String_value>,

"name":<String_value>,

"targetlbvserver":<String_value>,

"gotopriorityexpression":<String_value>,

"invoke":<Boolean_value>,

"labeltype":<String_value>

}]}
```

count

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/csvserver_transformpolicy_binding/name_value<String>?**count=yes**

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{"csvserver_transformpolicy_binding": [ { "__count": "#no"} ] }
```

csvserver_vpnvserver_binding

Binding object showing the vpnvserver that can be bound to csvserver.

Properties

(click to see Operations)

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
vserver	<String>	Read-write	Name of the default gslb or vpn vserver bound to CS vserver of type GSLB/VPN. For Example: bind cs vserver cs1 -vserver gslb1 or bind cs vserver cs1 -vserver vpn1. Minimum length = 1
name	<String>	Read-write	Name of the content switching virtual server to which the content switching policy applies. Minimum length = 1
hits	<Double>	Read-only	Number of hits.
__count	<Double>	Read-only	count parameter

Operations

(click to see Properties)

ADD:| DELETE:| GET| GET (ALL)| COUNT

Some options that you can use for each operations:

- **Getting warnings in response:**NITRO allows you to get warnings in an operation by specifying the "warning" query parameter as "yes". For example, to get warnings while connecting to the Citrix ADC appliance, the URL is as follows:

  http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/login?warning=yes

  If any, the warnings are displayed in the response payload with the HTTP code "209 X-NITRO-WARNING".

- **Authenticated access for individual NITRO operations:**NITRO allows you to logon to the Citrix ADC appliance to perform individual operations. You can use this option instead of creating a NITRO session (using the login object) and then using that session to perform all operations,

  To do this, you must specify the username and password in the request header of the NITRO request as follows:

  X-NITRO-USER:<username>

  X-NITRO-PASS:<password>

  Note: In such cases, make sure that the request header DOES not include the following:

  Cookie:NITRO_AUTH_TOKEN=<tokenvalue>

`*Note:`*

Mandatory parameters are marked in redand placeholder content is marked in <green>.

add

URL:http://<Citrix-ADC-IP-address(NSIP)/nitro/v1/config/csvserver_vpnvserver_binding

HTTP Method:PUT

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Content-Type:application/json

Request Payload:

```
{

"csvserver_vpnvserver_binding":{

<b>"name":<String_value>,

</b>"vserver":<String_value>

}}
```

Response:

HTTP Status Code on Success: 201 Created HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error

delete

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/csvserver_vpnvserver_binding/name_value<String>

Query-parameters:

args

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/csvserver_vpnvserver_binding/name_value<String>?**args=vserver:<String_value>**

HTTP Method:DELETE

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue>

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error

get

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/csvserver_vpnvserver_binding/name_value<String>

Query-parameters:

filter

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/csvserver_vpnvserver_binding/name_value<String>?**filter=property-name1:property-value1,property-name2:property-value2**

Use this query-parameter to get the filtered set of csvserver_vpnvserver_binding resources configured on Citrix ADC. Filtering can be done on any of the properties of the resource.

pagination

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/csvserver_vpnvserver_binding/name_value<String>?**pagesize=#no;pageno=#no**

Use this query-parameter to get the csvserver_vpnvserver_binding resources in chunks.

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "csvserver_vpnvserver_binding": [ {

"vserver":<String_value>,

"name":<String_value>,
```

```
"hits":<Double_value>

}]}
```

get (all)

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/csvserver_vpnvserver_binding

Query-parameters:

bulkbindings

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/csvserver_vpnvserver_binding?**bulkbindings=yes**

NITRO allows you to fetch bindings in bulk.

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "csvserver_vpnvserver_binding": [ {

"vserver":<String_value>,

"name":<String_value>,

"hits":<Double_value>
```

```
}]}
```

count

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/csvserver_vpnvserver_binding/name_value<String>?**count=yes**

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{"csvserver_vpnvserver_binding": [ { "__count": "#no"} ] }
```

DB

Db

All the commands associated with database user

NAME	DESCRIPTION
dbdbprofile	Configuration for DB profile
dbuser	Configuration for DB user

dbdbprofile

Configuration for DB profile resource.

Properties

(click to see Operations)

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
name	<String>	Read-write	Name for the database profile. Must begin with an ASCII alphanumeric or underscore (_) character, and must contain only ASCII alphanumeric, underscore, hash (#), period (.), space, colon (:), at sign (@), equal sign (=), and hyphen (-) characters. Cannot be changed after the profile is created. CLI Users: If the name includes one or more spaces, enclose the name in double or single quotation marks (for example, "my profile" or 'my profile'). . Minimum length = 1 Maximum length = 127

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
interpretquery	<String>	Read-write	If ENABLED, inspect the query and update the connection information, if required. If DISABLED, forward the query to the server. Default value: YES Possible values = YES, NO
stickiness	<String>	Read-write	If the queries are related to each other, forward to the same backend server. Default value: NO Possible values = YES, NO
kcdaccount	<String>	Read-write	Name of the KCD account that is used for Windows authentication. Minimum length = 1 Maximum length = 127
conmultiplex	<String>	Read-write	Use the same server-side connection for multiple client-side requests. Default is enabled. Default value: ENABLED

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
			Possible values = ENABLED, DISABLED
enablecachingconmuxoff	<String>	Read-write	Enable caching when connection multiplexing is OFF. Default value: DISABLED Possible values = ENABLED, DISABLED
refcnt	<Double>	Read-only	Profile Reference Count.
__count	<Double>	Read-only	count parameter

Operations

(click to see Properties)

ADD| DELETE| UPDATE| UNSET| GET (ALL)| GET| COUNT

Some options that you can use for each operations:

- **Getting warnings in response:**NITRO allows you to get warnings in an operation by specifying the "warning" query parameter as "yes". For example, to get warnings while connecting to the Citrix ADC appliance, the URL is as follows:

  http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/login?warning=yes

  If any, the warnings are displayed in the response payload with the HTTP code "209 X-NITRO-WARNING".

- **Authenticated access for individual NITRO operations:**NITRO allows you to logon to the Citrix ADC appliance to perform individual operations. You can use this option instead of creating a NITRO session (using the login object) and then using that session to perform all operations,

To do this, you must specify the username and password in the request header of the NITRO request as follows:

X-NITRO-USER:<username>

X-NITRO-PASS:<password>

Note: In such cases, make sure that the request header DOES not include the following:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue>

`*Note:`*

Mandatory parameters are marked in redand placeholder content is marked in <green>.

add

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/dbdbprofile

HTTP Method:POST

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Content-Type:application/json

Request Payload:

```
{"dbdbprofile":{

<b>"name":<String_value>,

</b>"interpretquery":<String_value>,

"stickiness":<String_value>,

"kcdaccount":<String_value>,

"conmultiplex":<String_value>,

"enablecachingconmuxoff":<String_value>

}}
```

Response:

HTTP Status Code on Success: 201 Created HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error

delete

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/dbdbprofile/name_value<String>

HTTP Method:DELETE

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue>

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error

update

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/dbdbprofile

HTTP Method:PUT

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Content-Type:application/json

Request Payload:

```
{"dbdbprofile":{

<b>"name":<String_value>,

</b>"interpretquery":<String_value>,

"stickiness":<String_value>,
```

```
"kcdaccount":<String_value>,

"conmultiplex":<String_value>,

"enablecachingconmuxoff":<String_value>

}}
```

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error

unset

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/dbdbprofile?**action=unset**

HTTP Method:POST

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Content-Type:application/json

Request Payload:

```
{"dbdbprofile":{

<b>"name":<String_value>,

</b>"interpretquery":true,

"stickiness":true,

"kcdaccount":true,

"conmultiplex":true,

"enablecachingconmuxoff":true
```

```
}}
```

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error

get (all)

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/dbdbprofile

Query-parameters:

attrs

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/dbdbprofile?**attrs=property-name1,property-name2**

Use this query parameter to specify the resource details that you want to retrieve.

filter

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/dbdbprofile?**filter=property-name1:property-val1,property-name2:property-val2**

Use this query-parameter to get the filtered set of dbdbprofile resources configured on Citrix ADC. Filtering can be done on any of the properties of the resource.

view

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/dbdbprofile?**view=summary**

Note: By default, the retrieved results are displayed in detail view (?view=detail).

pagination

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/dbdbprofile?**pagesize=#no;pageno=#no**

Use this query-parameter to get the dbdbprofile resources in chunks.

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "dbdbprofile": [ {

"name":<String_value>,

"interpretquery":<String_value>,

"stickiness":<String_value>,

"kcdaccount":<String_value>,

"conmultiplex":<String_value>,

"refcnt":<Double_value>,

"enablecachingconmuxoff":<String_value>

}]}
```

get

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/dbdbprofile/name_value<String>

Query-parameters:

attrs

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/dbdbprofile/name_value<String>?**attrs=property-name1,property-name2**

Use this query parameter to specify the resource details that you want to retrieve.

view

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/dbdbprofile/name_value<String>?**view=summary**

Note: By default, the retrieved results are displayed in detail view (?view=detail).

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "dbdbprofile": [ {

"name":<String_value>,

"interpretquery":<String_value>,

"stickiness":<String_value>,

"kcdaccount":<String_value>,

"conmultiplex":<String_value>,

"refcnt":<Double_value>,

"enablecachingconmuxoff":<String_value>

}]}
```

count

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/dbdbprofile?**count=yes**

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "dbdbprofile": [ { "__count": "#no"} ] }
```

dbuser

Configuration for DB user resource.

Properties

(click to see Operations)

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
username	<String>	Read-write	Name of the database user. Must be the same as the user name specified in the database. Minimum length = 1

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
password	<String>	Read-write	Password for logging on to the database. Must be the same as the password specified in the database. Minimum length = 1
loggedin	<Boolean>	Read-write	Display the names of all database users currently logged on to the Citrix ADC.
__count	<Double>	Read-only	count parameter

Operations

(click to see Properties)

ADD| DELETE| UPDATE| GET (ALL)| GET| COUNT

Some options that you can use for each operations:

- **Getting warnings in response:**NITRO allows you to get warnings in an operation by specifying the "warning" query parameter as "yes". For example, to get warnings while connecting to the Citrix ADC appliance, the URL is as follows:

  http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/login?warning=yes

  If any, the warnings are displayed in the response payload with the HTTP code "209 X-NITRO-WARNING".

- **Authenticated access for individual NITRO operations:**NITRO allows you to logon to the Citrix ADC appliance to perform individual operations. You can use this option instead of creating a NITRO session (using the login object) and then using that session to perform all operations,

  To do this, you must specify the username and password in the request header of the NITRO request as follows:

  X-NITRO-USER:<username>

X-NITRO-PASS:<password>

Note: In such cases, make sure that the request header DOES not include the following:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue>

`*Note:`*

Mandatory parameters are marked in redand placeholder content is marked in <green>.

add

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/dbuser

HTTP Method:POST

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Content-Type:application/json

Request Payload:

```
{"dbuser":{

<b>"username":<String_value>,

</b>"password":<String_value>

}}
```

Response:

HTTP Status Code on Success: 201 Created HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error

delete

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/dbuser/username_value<String>

HTTP Method:DELETE

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue>

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error

update

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/dbuser

HTTP Method:PUT

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Content-Type:application/json

Request Payload:

```
{"dbuser":{

<b>"username":<String_value>,

</b><b>"password":<String_value>

</b>}}
```

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error

get (all)

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/dbuser

Query-parameters:

args

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/dbuser?**args=username:<String_value>,loggedin:<Boolean_value>**

Use this query-parameter to get dbuser resources based on additional properties.

attrs

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/dbuser?**attrs=property-name1,property-name2**

Use this query parameter to specify the resource details that you want to retrieve.

filter

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/dbuser?**filter=property-name1:property-val1,property-name2:property-val2**

Use this query-parameter to get the filtered set of dbuser resources configured on Citrix ADC. Filtering can be done on any of the properties of the resource.

view

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/dbuser?**view=summary**

Note: By default, the retrieved results are displayed in detail view (?view=detail).

pagination

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/dbuser?**pagesize=#no;pageno=#no**

Use this query-parameter to get the dbuser resources in chunks.

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "dbuser": [ {

username:<String_value>,loggedin:<Boolean_value>

}]}
```

get

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/dbuser/username_value<String>

Query-parameters:

attrs

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/dbuser/username_value<String>?**attrs=property-name1,property-name2**

Use this query parameter to specify the resource details that you want to retrieve.

view

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/dbuser/username_value<String>?**view=summary**

Note: By default, the retrieved results are displayed in detail view (?view=detail).

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "dbuser": [ {

username:<String_value>,loggedin:<Boolean_value>

}]}
```

count

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/dbuser?**count=yes**

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "dbuser": [ { "__count": "#no"} ] }
```

Domain Name Service

Domain Name Service

Domain Name Service(DNS) configuration.

NAME	DESCRIPTION
dnsaaaarec	Configuration for IPv6 address type record
dnsaction	Configuration for DNS action
dnsaction64	Configuration for dns64 action
dnsaddrec	Configuration for address type record
dnscnamerec	Configuration for CNAME record
dnsglobal_binding	Binding object showing the resources that can be bound to dnsglobal
dnsglobal_dnspolicy_binding	Binding object showing the dnspolicy that can be bound to dnsglobal
dnskey	Configuration for dns key
dnsmxrec	Configuration for MX record
dnsnameserver	Configuration for name server
dnsnaptrrec	Configuration for NAPTR record
dnsnsecrec	
dnsnsrec	Configuration for name server record
dnsparameter	Configuration for DNS parameter
dnspolicy	Configuration for DNS policy
dnspolicy64	Configuration for dns64 policy
dnspolicy64_binding	Binding object showing the resources that can be bound to dnspolicy64
dnspolicy64_lbvserver_binding	Binding object showing the lbvserver that can be bound to dnspolicy64
dnspolicy_binding	Binding object showing the resources that can be bound to dnspolicy

NAME	DESCRIPTION
dnspolicy_dnsglobal_binding	Binding object showing the dnsglobal that can be bound to dnspolicy
dnspolicy_dnspolicylabel_binding	Binding object showing the dnspolicylabel that can be bound to dnspolicy
dnspolicylabel	Configuration for dns policy label
dnspolicylabel_binding	Binding object showing the resources that can be bound to dnspolicylabel
dnspolicylabel_dnspolicy_binding	Binding object showing the dnspolicy that can be bound to dnspolicylabel
dnspolicylabel_policybinding_binding	Binding object showing the policybinding that can be bound to dnspolicylabel
dnsprofile	Configuration for DNS profile
dnsproxyrecords	Configuration for proxy record
dnsptrrec	Configuration for PTR record
dnssoarec	Configuration for SOA record
dnssrvrec	Configuration for server record
dnssubnetcache	Configuration for subnet cache
dnssuffix	Configuration for DNS suffix
dnstxtrec	Configuration for TXT record
dnsview	Configuration for DNS view
dnsview_binding	Binding object showing the resources that can be bound to dnsview
dnsview_dnspolicy_binding	Binding object showing the dnspolicy that can be bound to dnsview

NAME	DESCRIPTION
dnsview_gslbservice_binding	Binding object showing the gslbservice that can be bound to dnsview
dnszone	Configuration for DNS zone
dnszone_binding	Binding object showing the resources that can be bound to dnszone
dnszone_dnskey_binding	Binding object showing the dnskey that can be bound to dnszone
dnszone_domain_binding	Binding object showing the domain that can be bound to dnszone

dnsaaaarec

Configuration for IPv6 address type record resource.

Properties

(click to see Operations)

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
hostname	<String>	Read-write	Domain name. Minimum length = 1
ipv6address	<String>	Read-write	One or more IPv6 addresses to assign to the domain name. Minimum length = 1
ttl	<Double>	Read-write	Time to Live (TTL), in seconds, for the record. TTL is the time for which the record must be cached by DNS proxies. The specified TTL is applied to all the

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
			resource records that are of the same record type and belong to the specified domain name. For example, if you add an address record, with a TTL of 36000, to the domain name example.com, the TTLs of all the address records of example.com are changed to 36000. If the TTL is not specified, the Citrix ADC uses either the DNS zone's minimum TTL or, if the SOA record is not available on the appliance, the default value of 3600. Default value: 3600 Minimum value = 0 Maximum value = 2147483647
ecssubnet	<String>	Read-write	Subnet for which the cached records need to be removed.
type	<String>	Read-write	Type of records to display. Available settings function as follows:

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
			* ADNS - Display all authoritative address records. * PROXY - Display all proxy address records. * ALL - Display all address records. Possible values = ALL, ADNS, PROXY
nodeid	<Double>	Read-write	Unique number that identifies the cluster node. Minimum value = 0 Maximum value = 31
vservername	<String>	Read-only	Vitual server name.
authtype	<String>	Read-only	Authentication type. Possible values = ALL, ADNS, PROXY
__count	<Double>	Read-only	count parameter

Operations

(click to see Properties)

ADD| DELETE| GET (ALL)| COUNT

Some options that you can use for each operations:

- **Getting warnings in response:**NITRO allows you to get warnings in an operation by specifying the "warning" query parameter as "yes". For example, to get warnings while connecting to the Citrix ADC appliance, the URL is as follows:

  http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/login?warning=yes

  If any, the warnings are displayed in the response payload with the HTTP code "209 X-NITRO-WARNING".

- **Authenticated access for individual NITRO operations:**NITRO allows you to logon to the Citrix ADC appliance to perform individual operations. You can use this option instead of creating a NITRO session (using the login object) and then using that session to perform all operations,

  To do this, you must specify the username and password in the request header of the NITRO request as follows:

  X-NITRO-USER:<username>

  X-NITRO-PASS:<password>

  Note: In such cases, make sure that the request header DOES not include the following:

  Cookie:NITRO_AUTH_TOKEN=<tokenvalue>

`*Note:`*

Mandatory parameters are marked in redand placeholder content is marked in <green>.

add

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/dnsaaaarec

HTTP Method:POST

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Content-Type:application/json

Request Payload:

```
{"dnsaaaarec":{
```

```
<b>"hostname":<String_value>,

</b><b>"ipv6address":<String_value>,

</b>"ttl":<Double_value>

}}
```

Response:

HTTP Status Code on Success: 201 Created HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error

delete

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/dnsaaaarec/hostname_value<String>

HTTP Method:DELETE

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue>

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error

get (all)

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/dnsaaaarec

Query-parameters:

args

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/dnsaaaarec?
args=hostname:<String_value>,ipv6address:<String_value>,type:<String_value>,nodeid:<Double_valu

Use this query-parameter to get dnsaaaarec resources based on additional properties.

attrs

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/dnsaaaarec?**attrs=property-name1,property-name2**

Use this query parameter to specify the resource details that you want to retrieve.

filter

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/dnsaaaarec?**filter=property-name1:property-val1,property-name2:property-val2**

Use this query-parameter to get the filtered set of dnsaaaarec resources configured on Citrix ADC. Filtering can be done on any of the properties of the resource.

view

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/dnsaaaarec?**view=summary**

Note: By default, the retrieved results are displayed in detail view (?view=detail).

pagination

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/dnsaaaarec?**pagesize=#no;pageno=#no**

Use this query-parameter to get the dnsaaaarec resources in chunks.

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "dnsaaaarec": [ {
```

```
hostname:<String_value>,ipv6address:<String_value>,type:<String_value>,nod
eid:<Double_value>"ecssubnet":<String_value>,

"ttl":<Double_value>,

"vservername":<String_value>,

"authtype":<String_value>

}]}
```

count

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/dnsaaaarec?**count=yes**

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "dnsaaaarec": [ { "__count": "#no"} ] }
```

dnsaction

Configuration for DNS action resource.

Properties

(click to see Operations)

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
actionname	<String>	Read-write	Name of the dns action.
actiontype	<String>	Read-write	The type of DNS action that is being configured. Possible values = ViewName, GslbPrefLoc, noop, Drop, Cache_Bypass, Rewrite_Response
ipaddress	<String[]>	Read-write	List of IP address to be returned in case of rewrite_response actiontype. They can be of IPV4 or IPV6 type. In case of set command We will remove all the IP address previously present in the action and will add new once given in set dns action command.
ttl	<Double>	Read-write	Time to live, in seconds. Default value: 3600 Minimum value = 0

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
			Maximum value = 2147483647
viewname	<String>	Read-write	The view name that must be used for the given action.
preferredloclist	<String[]>	Read-write	The location list in priority order used for the given action. Minimum length = 1
dnsprofilename	<String>	Read-write	Name of the DNS profile to be associated with the transaction for which the action is chosen. Minimum length = 1 Maximum length = 127
drop	<String>	Read-only	The dns packet must be dropped. Possible values = YES, NO
cachebypass	<String>	Read-only	By pass dns cache for this. Possible values = YES, NO
__count	<Double>	Read-only	count parameter

Operations

(click to see Properties)

ADD| DELETE| UPDATE| UNSET| GET (ALL)| GET| COUNT

Some options that you can use for each operations:

- **Getting warnings in response:**NITRO allows you to get warnings in an operation by specifying the "warning" query parameter as "yes". For example, to get warnings while connecting to the Citrix ADC appliance, the URL is as follows:

  http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/login?warning=yes

  If any, the warnings are displayed in the response payload with the HTTP code "209 X-NITRO-WARNING".

- **Authenticated access for individual NITRO operations:**NITRO allows you to logon to the Citrix ADC appliance to perform individual operations. You can use this option instead of creating a NITRO session (using the login object) and then using that session to perform all operations,

  To do this, you must specify the username and password in the request header of the NITRO request as follows:

  X-NITRO-USER:<username>

  X-NITRO-PASS:<password>

  Note: In such cases, make sure that the request header DOES not include the following:

  Cookie:NITRO_AUTH_TOKEN=<tokenvalue>

`*Note:`*

Mandatory parameters are marked in redand placeholder content is marked in <green>.

add

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/dnsaction

HTTP Method:POST

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Content-Type:application/json

Request Payload:

```
{"dnsaction":{
```

```
<b>"actionname":<String_value>,

</b><b>"actiontype":<String_value>,

</b>"ipaddress":<String[]_value>,

"ttl":<Double_value>,

"viewname":<String_value>,

"preferredloclist":<String[]_value>,

"dnsprofilename":<String_value>

}}
```

Response:

HTTP Status Code on Success: 201 Created HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error

delete

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/dnsaction/actionname_value<String>

HTTP Method:DELETE

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue>

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error

update

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/dnsaction

HTTP Method:PUT

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Content-Type:application/json

Request Payload:

```
{"dnsaction":{

<b>"actionname":<String_value>,

</b>"ipaddress":<String[]_value>,

"ttl":<Double_value>,

"viewname":<String_value>,

"preferredloclist":<String[]_value>,

"dnsprofilename":<String_value>

}}
```

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error

unset

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/dnsaction?**action=unset**

HTTP Method:POST

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Content-Type:application/json

Request Payload:

```
{"dnsaction":{

<b>"actionname":<String_value>,

</b>"ttl":true,

"dnsprofilename":true

}}
```

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error

get (all)

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/dnsaction

Query-parameters:

attrs

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/dnsaction?**attrs=property-name1,property-name2**

Use this query parameter to specify the resource details that you want to retrieve.

filter

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/dnsaction?**filter=property-name1:property-val1,property-name2:property-val2**

Use this query-parameter to get the filtered set of dnsaction resources configured on Citrix ADC. Filtering can be done on any of the properties of the resource.

view

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/dnsaction?**view=summary**

Note: By default, the retrieved results are displayed in detail view (?view=detail).

pagination

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/dnsaction?**pagesize=#no;pageno=#no**

Use this query-parameter to get the dnsaction resources in chunks.

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "dnsaction": [ {

"actionname":<String_value>,

"actiontype":<String_value>,

"ttl":<Double_value>,

"ipaddress":<String[]_value>,

"viewname":<String_value>,

"preferredloclist":<String[]_value>,

"drop":<String_value>,

"cachebypass":<String_value>,
```

```
"dnsprofilename":<String_value>

}]}
```

get

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/dnsaction/actionname_value<String>

Query-parameters:

attrs

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/dnsaction/actionname_value<String>?**attrs=property-name1,property-name2**

Use this query parameter to specify the resource details that you want to retrieve.

view

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/dnsaction/actionname_value<String>?**view=summary**

Note: By default, the retrieved results are displayed in detail view (?view=detail).

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "dnsaction": [ {
```

```
"actionname":<String_value>,

"actiontype":<String_value>,

"ttl":<Double_value>,

"ipaddress":<String[]_value>,

"viewname":<String_value>,

"preferredloclist":<String[]_value>,

"drop":<String_value>,

"cachebypass":<String_value>,

"dnsprofilename":<String_value>

}]}
```

count

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/dnsaction?**count=yes**

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "dnsaction": [ { "__count": "#no"} ] }
```

dnsaction64

Configuration for dns64 action resource.

Properties

(click to see Operations)

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
actionname	<String>	Read-write	Name of the dns64 action.
prefix	<String>	Read-write	The dns64 prefix to be used if the after evaluating the rules.
mappedrule	<String>	Read-write	The expression to select the criteria for ipv4 addresses to be used for synthesis. Only if the mappedrule is evaluated to true the corresponding ipv4 address is used for synthesis using respective prefix, otherwise the A RR is discarded.
excluderule	<String>	Read-write	The expression to select the criteria for eliminating the corresponding ipv6

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
			addresses from the response.
__count	<Double>	Read-only	count parameter

Operations

(click to see Properties)

ADD| DELETE| UPDATE| UNSET| GET (ALL)| GET| COUNT

Some options that you can use for each operations:

- **Getting warnings in response:**NITRO allows you to get warnings in an operation by specifying the "warning" query parameter as "yes". For example, to get warnings while connecting to the Citrix ADC appliance, the URL is as follows:

  http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/login?warning=yes

  If any, the warnings are displayed in the response payload with the HTTP code "209 X-NITRO-WARNING".
- **Authenticated access for individual NITRO operations:**NITRO allows you to logon to the Citrix ADC appliance to perform individual operations. You can use this option instead of creating a NITRO session (using the login object) and then using that session to perform all operations,

  To do this, you must specify the username and password in the request header of the NITRO request as follows:

  X-NITRO-USER:<username>

  X-NITRO-PASS:<password>

  Note: In such cases, make sure that the request header DOES not include the following:

  Cookie:NITRO_AUTH_TOKEN=<tokenvalue>

`*Note:`*

Mandatory parameters are marked in redand placeholder content is marked in <green>.

add

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/dnsaction64

HTTP Method:POST

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Content-Type:application/json

Request Payload:

```
{"dnsaction64":{

<b>"actionname":<String_value>,

</b><b>"prefix":<String_value>,

</b>"mappedrule":<String_value>,

"excluderule":<String_value>

}}
```

Response:

HTTP Status Code on Success: 201 Created HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error

delete

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/dnsaction64/actionname_value<String>

HTTP Method:DELETE

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue>

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error

update

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/dnsaction64

HTTP Method:PUT

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Content-Type:application/json

Request Payload:

```
{"dnsaction64":{

<b>"actionname":<String_value>,

</b>"prefix":<String_value>,

"mappedrule":<String_value>,

"excluderule":<String_value>

}}
```

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error

unset

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/dnsaction64?**action=unset**

HTTP Method:POST

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Content-Type:application/json

Request Payload:

```
{"dnsaction64":{

<b>"actionname":<String_value>,

</b>"prefix":true,

"mappedrule":true,

"excluderule":true

}}
```

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error

get (all)

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/dnsaction64

Query-parameters:

attrs

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/dnsaction64?**attrs=property-name1,property-name2**

Use this query parameter to specify the resource details that you want to retrieve.

filter

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/dnsaction64?**filter=property-name1:property-val1,property-name2:property-val2**

Use this query-parameter to get the filtered set of dnsaction64 resources configured on Citrix ADC. Filtering can be done on any of the properties of the resource.

view

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/dnsaction64?**view=summary**

Note: By default, the retrieved results are displayed in detail view (?view=detail).

pagination

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/dnsaction64?**pagesize=#no;pageno=#no**

Use this query-parameter to get the dnsaction64 resources in chunks.

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "dnsaction64": [ {

"actionname":<String_value>,

"prefix":<String_value>,

"mappedrule":<String_value>,

"excluderule":<String_value>

}]}
```

get

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/dnsaction64/actionname_value<String>

Query-parameters:

attrs

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/dnsaction64/actionname_value<String>?**attrs=property-name1,property-name2**

Use this query parameter to specify the resource details that you want to retrieve.

view

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/dnsaction64/actionname_value<String>?**view=summary**

Note: By default, the retrieved results are displayed in detail view (?view=detail).

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "dnsaction64": [ {

"actionname":<String_value>,

"prefix":<String_value>,
```

```
"mappedrule":<String_value>,

"excluderule":<String_value>

}]}
```

count

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/dnsaction64?**count=yes**

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "dnsaction64": [ { "__count": "#no"} ] }
```

dnsaddrec

Configuration for address type record resource.

Properties

(click to see Operations)

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
hostname	<String>	Read-write	Domain name.

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
			Minimum length = 1
ipaddress	<String>	Read-write	One or more IPv4 addresses to assign to the domain name. Minimum length = 1
ttl	<Double>	Read-write	Time to Live (TTL), in seconds, for the record. TTL is the time for which the record must be cached by DNS proxies. The specified TTL is applied to all the resource records that are of the same record type and belong to the specified domain name. For example, if you add an address record, with a TTL of 36000, to the domain name example.com, the TTLs of all the address records of example.com are changed to 36000. If the TTL is not specified, the Citrix ADC uses either the DNS zone's minimum TTL or, if the SOA record is not available on the appliance, the

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
			default value of 3600. Default value: 3600 Minimum value = 0 Maximum value = 2147483647
ecssubnet	<String>	Read-write	Subnet for which the cached address records need to be removed.
type	<String>	Read-write	The address record type. The type can take 3 values: ADNS - If this is specified, all of the authoritative address records will be displayed. PROXY - If this is specified, all of the proxy address records will be displayed. ALL - If this is specified, all of the address records will be displayed. Possible values = ALL, ADNS, PROXY

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
nodeid	<Double>	Read-write	Unique number that identifies the cluster node. Minimum value = 0 Maximum value = 31
vservername	<String>	Read-only	Vitual server name.
authtype	<String>	Read-only	Authentication type. Possible values = ALL, ADNS, PROXY
__count	<Double>	Read-only	count parameter

Operations

(click to see Properties)

ADD| DELETE| GET (ALL)| GET| COUNT

Some options that you can use for each operations:

- **Getting warnings in response:**NITRO allows you to get warnings in an operation by specifying the "warning" query parameter as "yes". For example, to get warnings while connecting to the Citrix ADC appliance, the URL is as follows:

  http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/login?warning=yes

  If any, the warnings are displayed in the response payload with the HTTP code "209 X-NITRO-WARNING".

- **Authenticated access for individual NITRO operations:**NITRO allows you to logon to the Citrix ADC appliance to perform individual operations. You can use this option instead of creating a NITRO session (using the login object) and then using that session to perform all operations,

  To do this, you must specify the username and password in the request header of the NITRO request as follows:

X-NITRO-USER:<username>

X-NITRO-PASS:<password>

Note: In such cases, make sure that the request header DOES not include the following:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue>

`*Note:` *

Mandatory parameters are marked in redand placeholder content is marked in <green>.

add

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/dnsaddrec

HTTP Method:POST

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Content-Type:application/json

Request Payload:

```
{"dnsaddrec":{

<b>"hostname":<String_value>,

</b><b>"ipaddress":<String_value>,

</b>"ttl":<Double_value>

}}
```

Response:

HTTP Status Code on Success: 201 Created HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error

delete

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/dnsaddrec/hostname_value<String>

HTTP Method:DELETE

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue>

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error

get (all)

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/dnsaddrec

Query-parameters:

args

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/dnsaddrec?**args=hostname:<String_value>,type:<String_value>,nodeid:<Double_value>**

Use this query-parameter to get dnsaddrec resources based on additional properties.

attrs

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/dnsaddrec?**attrs=property-name1,property-name2**

Use this query parameter to specify the resource details that you want to retrieve.

filter

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/dnsaddrec?**filter=property-name1:property-val1,property-name2:property-val2**

Use this query-parameter to get the filtered set of dnsaddrec resources configured on Citrix ADC. Filtering can be done on any of the properties of the resource.

view

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/dnsaddrec?**view=summary**

Note: By default, the retrieved results are displayed in detail view (?view=detail).

pagination

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/dnsaddrec?**pagesize=#no;pageno=#no**

Use this query-parameter to get the dnsaddrec resources in chunks.

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "dnsaddrec": [ {

hostname:<String_value>,type:<String_value>,nodeid:<Double_value>"ipaddres
s":<String_value>,

"ecssubnet":<String_value>,

"ttl":<Double_value>,

"vservername":<String_value>,

"authtype":<String_value>

}]}
```

get

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/dnsaddrec/hostname_value<String>

Query-parameters:

attrs

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/dnsaddrec/hostname_value<String>?**attrs=property-name1,property-name2**

Use this query parameter to specify the resource details that you want to retrieve.

view

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/dnsaddrec/hostname_value<String>?**view=summary**

Note: By default, the retrieved results are displayed in detail view (?view=detail).

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "dnsaddrec": [ {

hostname:<String_value>,type:<String_value>,nodeid:<Double_value>"ipaddres
s":<String_value>,

"ecssubnet":<String_value>,

"ttl":<Double_value>,

"vservername":<String_value>,

"authtype":<String_value>
```

```
}]}
```

count

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/dnsaddrec?**count=yes**

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "dnsaddrec": [ { "__count": "#no"} ] }
```

dnscnamerec

Configuration for CNAME record resource.

Properties

(click to see Operations)

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
aliasname	<String>	Read-write	Alias for the canonical domain name. Minimum length = 1

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
canonicalname	<String>	Read-write	Canonical domain name. Minimum length = 1
ttl	<Double>	Read-write	Time to Live (TTL), in seconds, for the record. TTL is the time for which the record must be cached by DNS proxies. The specified TTL is applied to all the resource records that are of the same record type and belong to the specified domain name. For example, if you add an address record, with a TTL of 36000, to the domain name example.com, the TTLs of all the address records of example.com are changed to 36000. If the TTL is not specified, the Citrix ADC uses either the DNS zone's minimum TTL or, if the SOA record is not available on the appliance, the default value of 3600. Default value: 3600

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
			Minimum value = 0 Maximum value = 2147483647
ecssubnet	<String>	Read-write	Subnet for which the cached CNAME record need to be removed.
type	<String>	Read-write	Type of records to display. Available settings function as follows: * ADNS - Display all authoritative address records. * PROXY - Display all proxy address records. * ALL - Display all address records. Default value: ADNS Possible values = ALL, ADNS, PROXY
nodeid	<Double>	Read-write	Unique number that identifies the cluster node. Minimum value = 0 Maximum value = 31

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
vservername	<String>	Read-only	GSLB Vitual server name to which this domain is bound.
authtype	<String>	Read-only	Record type. Possible values = ALL, ADNS, PROXY
__count	<Double>	Read-only	count parameter

Operations

(click to see Properties)

ADD| DELETE| GET (ALL)| GET| COUNT

Some options that you can use for each operations:

- **Getting warnings in response:**NITRO allows you to get warnings in an operation by specifying the "warning" query parameter as "yes". For example, to get warnings while connecting to the Citrix ADC appliance, the URL is as follows:

  http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/login?warning=yes

  If any, the warnings are displayed in the response payload with the HTTP code "209 X-NITRO-WARNING".

- **Authenticated access for individual NITRO operations:**NITRO allows you to logon to the Citrix ADC appliance to perform individual operations. You can use this option instead of creating a NITRO session (using the login object) and then using that session to perform all operations,

  To do this, you must specify the username and password in the request header of the NITRO request as follows:

  X-NITRO-USER:<username>

  X-NITRO-PASS:<password>

  Note: In such cases, make sure that the request header DOES not include the following:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue>

`*Note:` *

Mandatory parameters are marked in redand placeholder content is marked in <green>.

add

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/dnscnamerec

HTTP Method:POST

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Content-Type:application/json

Request Payload:

```
{"dnscnamerec":{

<b>"aliasname":<String_value>,

</b><b>"canonicalname":<String_value>,

</b>"ttl":<Double_value>

}}
```

Response:

HTTP Status Code on Success: 201 Created HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error

delete

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/dnscnamerec/aliasname_value<String>

HTTP Method:DELETE

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue>

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error

get (all)

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/dnscnamerec

Query-parameters:

args

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/dnscnamerec?**args=aliasname:<String_value>,type:<String_value>,nodeid:<Double_value>**

Use this query-parameter to get dnscnamerec resources based on additional properties.

attrs

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/dnscnamerec?**attrs=property-name1,property-name2**

Use this query parameter to specify the resource details that you want to retrieve.

filter

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/dnscnamerec?**filter=property-name1:property-val1,property-name2:property-val2**

Use this query-parameter to get the filtered set of dnscnamerec resources configured on Citrix ADC. Filtering can be done on any of the properties of the resource.

view

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/dnscnamerec?**view=summary**

Note: By default, the retrieved results are displayed in detail view (?view=detail).

pagination

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/dnscnamerec?**pagesize=#no;pageno=#no**

Use this query-parameter to get the dnscnamerec resources in chunks.

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "dnscnamerec": [ {

aliasname:<String_value>,type:<String_value>,nodeid:<Double_value>"canonic
alname":<String_value>,

"ttl":<Double_value>,

"vservername":<String_value>,

"authtype":<String_value>,

"ecssubnet":<String_value>

}]}
```

get

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/dnscnamerec/aliasname_value<String>

Query-parameters:

attrs

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/dnscnamerec/aliasname_value<String>?**attrs=property-name1,property-name2**

Use this query parameter to specify the resource details that you want to retrieve.

view

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/dnscnamerec/aliasname_value<String>?**view=summary**

Note: By default, the retrieved results are displayed in detail view (?view=detail).

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "dnscnamerec": [ {

aliasname:<String_value>,type:<String_value>,nodeid:<Double_value>"canonic
alname":<String_value>,

"ttl":<Double_value>,

"vservername":<String_value>,

"authtype":<String_value>,

"ecssubnet":<String_value>

}]}
```

count

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/dnscnamerec?**count=yes**

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "dnscnamerec": [ { "__count": "#no"} ] }
```

dnsglobal_binding

Binding object which returns the resources bound to dnsglobal.

Properties

(click to see Operations)

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
dnsglobal_dnspolicy_binding	<dnsglobal_dnspolicy_binding[]>	Read-only	dnspolicy that can be bound to dnsglobal.

Operations

(click to see Properties)

GET| GET (ALL)

Some options that you can use for each operations:

- **Getting warnings in response:**NITRO allows you to get warnings in an operation by specifying the "warning" query parameter as "yes". For example, to get warnings while connecting to the Citrix ADC appliance, the URL is as follows:

  http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/login?warning=yes

  If any, the warnings are displayed in the response payload with the HTTP code "209 X-NITRO-WARNING".

- **Authenticated access for individual NITRO operations:**NITRO allows you to logon to the Citrix ADC appliance to perform individual operations. You can use this option instead of creating a NITRO session (using the login object) and then using that session to perform all operations,

  To do this, you must specify the username and password in the request header of the NITRO request as follows:

  X-NITRO-USER:<username>

  X-NITRO-PASS:<password>

  Note: In such cases, make sure that the request header DOES not include the following:

  Cookie:NITRO_AUTH_TOKEN=<tokenvalue>

`*Note:`*

Mandatory parameters are marked in redand placeholder content is marked in <green>.

get

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/dnsglobal_binding

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "dnsglobal_binding": [ {

"dnsglobal_dnspolicy_binding":<dnsglobal_dnspolicy_binding[]_value>

}]}
```

get (all)

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/dnsglobal_binding

Query-parameters:

bulkbindings

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/dnsglobal_binding?**bulkbindings=yes**

NITRO allows you to fetch bindings in bulk.

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "dnsglobal_binding": [ {

"dnsglobal_dnspolicy_binding":<dnsglobal_dnspolicy_binding[]_value>
```

```
}]}
```

dnsglobal_dnspolicy_binding

Binding object showing the dnspolicy that can be bound to dnsglobal.

Properties

(click to see Operations)

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
priority	<Double>	Read-write	Specifies the priority of the policy with which it is bound. Maximum allowed priority should be less than 65535.
globalbindtype	<String>	Read-write	. Default value: SYSTEM_GLOBAL Possible values = SYSTEM_GLOBAL, VPN_GLOBAL, RNAT_GLOBAL
policyname	<String>	Read-write	Name of the dns policy.
labelname	<String>	Read-write	Name of the label to invoke if the current policy rule evaluates to TRUE.
gotopriorityexpression	<String>	Read-write	Expression or other value specifying the next policy to be evaluated if the current policy

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
			evaluates to TRUE. Specify one of the following values: * NEXT - Evaluate the policy with the next higher priority number. * END - End policy evaluation. * USE_INVOCATION_RESULT - Applicable if this policy invokes another policy label. If the final goto in the invoked policy label has a value of END, the evaluation stops. If the final goto is anything other than END, the current policy label performs a NEXT. * An expression that evaluates to a number. If you specify an expression, the number to which it evaluates determines the next policy to evaluate, as follows: * If the expression evaluates to a higher numbered priority, the policy with that priority is evaluated next. * If the expression evaluates to the priority of the current policy, the policy with the next

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
			higher numbered priority is evaluated next. * If the expression evaluates to a priority number that is numerically higher than the highest numbered priority, policy evaluation ends. An UNDEF event is triggered if: * The expression is invalid. * The expression evaluates to a priority number that is numerically lower than the current policy's priority. * The expression evaluates to a priority number that is between the current policy's priority number (say, 30) and the highest priority number (say, 100), but does not match any configured priority number (for example, the expression evaluates to the number 85). This example assumes that the priority number increments by 10 for every successive policy, and therefore a priority number of

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
			85 does not exist in the policy label. Minimum length = 1
invoke	<Boolean>	Read-write	Invoke flag.
type	<String>	Read-write	Type of global bind point for which to show bound policies. Possible values = REQ_OVERRIDE, REQ_DEFAULT, RES_OVERRIDE, RES_DEFAULT
labeltype	<String>	Read-write	Type of policy label invocation. Possible values = policylabel
flowtype	<Double>	Read-only	flowtype of the bound rewrite policy.
numpol	<Double>	Read-only	The number of policies bound to the bindpoint.
__count	<Double>	Read-only	count parameter

Operations

(click to see Properties)

ADD:| DELETE:| GET| GET (ALL)| COUNT

Some options that you can use for each operations:

- **Getting warnings in response:**NITRO allows you to get warnings in an operation by specifying the "warning" query parameter as "yes". For example, to get warnings while connecting to the Citrix ADC appliance, the URL is as follows:

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/login?warning=yes

If any, the warnings are displayed in the response payload with the HTTP code "209 X-NITRO-WARNING".

- **Authenticated access for individual NITRO operations:**NITRO allows you to logon to the Citrix ADC appliance to perform individual operations. You can use this option instead of creating a NITRO session (using the login object) and then using that session to perform all operations,

  To do this, you must specify the username and password in the request header of the NITRO request as follows:

  X-NITRO-USER:<username>

  X-NITRO-PASS:<password>

  Note: In such cases, make sure that the request header DOES not include the following:

  Cookie:NITRO_AUTH_TOKEN=<tokenvalue>

`*Note:`*

Mandatory parameters are marked in redand placeholder content is marked in <green>.

add

URL:http://<Citrix-ADC-IP-address(NSIP)/nitro/v1/config/dnsglobal_dnspolicy_binding

HTTP Method:PUT

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Content-Type:application/json

Request Payload:

```
{

"dnsglobal_dnspolicy_binding":{

<b>"policyname":<String_value>,

</b><b>"priority":<Double_value>,
```

```
</b>"gotopriorityexpression":<String_value>,

"type":<String_value>,

"invoke":<Boolean_value>,

"labeltype":<String_value>,

"labelname":<String_value>

}}
```

Response:

HTTP Status Code on Success: 201 Created HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error

delete

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/dnsglobal_dnspolicy_binding

Query-parameters:

args

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/dnsglobal_dnspolicy_binding?**args=policyname:<String_value>,type:<String_value>**

HTTP Method:DELETE

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue>

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error

get

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/dnsglobal_dnspolicy_binding

Query-parameters:

filter

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/dnsglobal_dnspolicy_binding?**filter=property-name1:property-value1,property-name2:property-value2**

Use this query-parameter to get the filtered set of dnsglobal_dnspolicy_binding resources configured on Citrix ADC. Filtering can be done on any of the properties of the resource.

pagination

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/dnsglobal_dnspolicy_binding?**pagesize=#no;pageno=#no**

Use this query-parameter to get the dnsglobal_dnspolicy_binding resources in chunks.

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "dnsglobal_dnspolicy_binding": [ {

"priority":<Double_value>,

"globalbindtype":<String_value>,
```

```
"policyname":<String_value>,

"labelname":<String_value>,

"gotopriorityexpression":<String_value>,

"invoke":<Boolean_value>,

"type":<String_value>,

"labeltype":<String_value>,

"flowtype":<Double_value>,

"numpol":<Double_value>

}]}
```

get (all)

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/dnsglobal_dnspolicy_binding

Query-parameters:

bulkbindings

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/dnsglobal_dnspolicy_binding?**bulkbindings=yes**

NITRO allows you to fetch bindings in bulk.

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "dnsglobal_dnspolicy_binding": [ {

"priority":<Double_value>,

"globalbindtype":<String_value>,

"policyname":<String_value>,

"labelname":<String_value>,

"gotopriorityexpression":<String_value>,

"invoke":<Boolean_value>,

"type":<String_value>,

"labeltype":<String_value>,

"flowtype":<Double_value>,

"numpol":<Double_value>

}]}
```

count

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/dnsglobal_dnspolicy_binding?**count=yes**

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "dnsglobal_dnspolicy_binding": [ { "__count": "#no"} ] }
```

dnskey

Configuration for dns key resource.

Properties

(click to see Operations)

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
keyname	<String>	Read-write	Name of the public-private key pair to publish in the zone. Minimum length = 1
publickey	<String>	Read-write	File name of the public key.
privatekey	<String>	Read-write	File name of the private key.
expires	<Double>	Read-write	Time period for which to consider the key valid, after the key is used to sign a zone. Default value: 120 Minimum value = 1

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
			Maximum value = 32767
units1	<String>	Read-write	Units for the expiry period. Default value: DAYS Possible values = MINUTES, HOURS, DAYS
notificationperiod	<Double>	Read-write	Time at which to generate notification of key expiration, specified as number of days, hours, or minutes before expiry. Must be less than the expiry period. The notification is an SNMP trap sent to an SNMP manager. To enable the appliance to send the trap, enable the DNSKEY-EXPIRY SNMP alarm. Default value: 7 Minimum value = 1 Maximum value = 32767
units2	<String>	Read-write	Units for the notification period.

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
			Default value: DAYS Possible values = MINUTES, HOURS, DAYS
ttl	<Double>	Read-write	Time to Live (TTL), in seconds, for the DNSKEY resource record created in the zone. TTL is the time for which the record must be cached by the DNS proxies. If the TTL is not specified, either the DNS zone's minimum TTL or the default value of 3600 is used. Default value: 3600 Minimum value = 0 Maximum value = 2147483647
password	<String>	Read-write	Passphrase for reading the encrypted public/ private DNS keys. Minimum length = 1
zonename	<String>	Read-write	Name of the zone for which to create a key. Minimum length = 1

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
keytype	<String>	Read-write	Type of key to create. Default value: NS_DNSKEY_ZSK Possible values = KSK, KeySigningKey, ZSK, ZoneSigningKey
algorithm	<String>	Read-write	Algorithm to generate for zone signing. Default value: NS_DNSKEYALGO_RSASHA1 Possible values = RSASHA1
keysize	<Double>	Read-write	Size of the key, in bits. Default value: 512
filenameprefix	<String>	Read-write	Common prefix for the names of the generated public and private key files and the Delegation Signer (DS) resource record. During key generation, the .key, .private, and .ds suffixes are appended automatically to the file name prefix to produce the names of the public key, the

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
			private key, and the DS record, respectively.
src	<String>	Read-write	URL (protocol, host, path, and file name) from where the DNS key file will be imported. NOTE: The import fails if the object to be imported is on an HTTPS server that requires client certificate authentication for access. This is a mandatory argument. Minimum length = 1 Maximum length = 2047
__count	<Double>	Read-only	count parameter

Operations

(click to see Properties)

ADD| CREATE| UPDATE| UNSET| DELETE| IMPORT| GET (ALL)| GET| COUNT

Some options that you can use for each operations:

- **Getting warnings in response:**NITRO allows you to get warnings in an operation by specifying the "warning" query parameter as "yes". For example, to get warnings while connecting to the Citrix ADC appliance, the URL is as follows:

  http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/login?warning=yes

  If any, the warnings are displayed in the response payload with the HTTP code "209 X-NITRO-WARNING".

- **Authenticated access for individual NITRO operations:**NITRO allows you to logon to the Citrix ADC appliance to perform individual operations. You can use this option instead of creating a NITRO session (using the login object) and then using that session to perform all operations,

  To do this, you must specify the username and password in the request header of the NITRO request as follows:

  X-NITRO-USER:<username>

  X-NITRO-PASS:<password>

  Note: In such cases, make sure that the request header DOES not include the following:

  Cookie:NITRO_AUTH_TOKEN=<tokenvalue>

`*Note:`*

Mandatory parameters are marked in redand placeholder content is marked in <green>.

add

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/dnskey

HTTP Method:POST

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Content-Type:application/json

Request Payload:

```
{"dnskey":{

<b>"keyname":<String_value>,

</b><b>"publickey":<String_value>,

</b><b>"privatekey":<String_value>,

</b>"expires":<Double_value>,

"units1":<String_value>,
```

```
"notificationperiod":<Double_value>,

"units2":<String_value>,

"ttl":<Double_value>,

"password":<String_value>

}}
```

Response:

HTTP Status Code on Success: 201 Created HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error

create

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/dnskey?**action=create**

HTTP Method:POST

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Content-Type:application/json

Request Payload:

```
{"dnskey":{

<b>"zonename":<String_value>,

</b><b>"keytype":<String_value>,

</b><b>"algorithm":<String_value>,

</b><b>"keysize":<Double_value>,

</b><b>"filenameprefix":<String_value>,
```

```
</b>"password":<String_value>

}}
```

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error

update

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/dnskey

HTTP Method:PUT

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Content-Type:application/json

Request Payload:

```
{"dnskey":{

<b>"keyname":<String_value>,

</b>"expires":<Double_value>,

"units1":<String_value>,

"notificationperiod":<Double_value>,

"units2":<String_value>,

"ttl":<Double_value>

}}
```

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error

unset

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/dnskey?**action=unset**

HTTP Method:POST

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Content-Type:application/json

Request Payload:

```
{"dnskey":{

<b>"keyname":<String_value>,

</b>"expires":true,

"units1":true,

"notificationperiod":true,

"units2":true,

"ttl":true

}}
```

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error

delete

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/dnskey/keyname_value<String>

HTTP Method:DELETE

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue>

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error

Import

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/dnskey?**action=Import**

HTTP Method:POST

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Content-Type:application/json

Request Payload:

```
{"dnskey":{

<b>"keyname":<String_value>,

</b><b>"src":<String_value>

</b>}}
```

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error

get (all)

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/dnskey

Query-parameters:

attrs

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/dnskey?**attrs=property-name1,property-name2**

Use this query parameter to specify the resource details that you want to retrieve.

filter

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/dnskey?**filter=property-name1:property-val1,property-name2:property-val2**

Use this query-parameter to get the filtered set of dnskey resources configured on Citrix ADC. Filtering can be done on any of the properties of the resource.

view

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/dnskey?**view=summary**

Note: By default, the retrieved results are displayed in detail view (?view=detail).

pagination

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/dnskey?**pagesize=#no;pageno=#no**

Use this query-parameter to get the dnskey resources in chunks.

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "dnskey": [ {

"keyname":<String_value>,

"publickey":<String_value>,

"privatekey":<String_value>,

"expires":<Double_value>,

"units1":<String_value>,

"notificationperiod":<Double_value>,

"units2":<String_value>,

"ttl":<Double_value>,

"zonename":<String_value>,

"password":<String_value>

}]}
```

get

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/dnskey/keyname_value<String>

Query-parameters:

attrs

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/dnskey/keyname_value<String>?**attrs=property-name1,property-name2**

Use this query parameter to specify the resource details that you want to retrieve.

view

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/dnskey/keyname_value<String>?**view=summary**

Note: By default, the retrieved results are displayed in detail view (?view=detail).

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "dnskey": [ {

"keyname":<String_value>,

"publickey":<String_value>,

"privatekey":<String_value>,

"expires":<Double_value>,

"units1":<String_value>,

"notificationperiod":<Double_value>,

"units2":<String_value>,

"ttl":<Double_value>,

"zonename":<String_value>,

"password":<String_value>

}]}
```

count

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/dnskey?**count=yes**

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "dnskey": [ { "__count": "#no"} ] }
```

dnsmxrec

Configuration for MX record resource.

Properties

(click to see Operations)

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
domain	<String>	Read-write	Domain name for which to add the MX record. Minimum length = 1

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
mx	<String>	Read-write	Host name of the mail exchange server. Minimum length = 1
pref	<Double>	Read-write	Priority number to assign to the mail exchange server. A domain name can have multiple mail servers, with a priority number assigned to each server. The lower the priority number, the higher the mail server's priority. When other mail servers have to deliver mail to the specified domain, they begin with the mail server with the lowest priority number, and use other configured mail servers, in priority order, as backups. Minimum value = 0 Maximum value = 65535
ttl	<Double>	Read-write	Time to Live (TTL), in seconds, for the record. TTL is the time for which the record must be cached by DNS proxies. The specified TTL is

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
			applied to all the resource records that are of the same record type and belong to the specified domain name. For example, if you add an address record, with a TTL of 36000, to the domain name example.com, the TTLs of all the address records of example.com are changed to 36000. If the TTL is not specified, the Citrix ADC uses either the DNS zone's minimum TTL or, if the SOA record is not available on the appliance, the default value of 3600. Default value: 3600 Minimum value = 0 Maximum value = 2147483647
ecssubnet	<String>	Read-write	Subnet for which the cached MX record need to be removed.
type	<String>	Read-write	Type of records to display. Available settings function as follows:

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
			* ADNS - Display all authoritative address records. * PROXY - Display all proxy address records. * ALL - Display all address records. Default value: ADNS Possible values = ALL, ADNS, PROXY
nodeid	<Double>	Read-write	Unique number that identifies the cluster node. Minimum value = 0 Maximum value = 31
authtype	<String>	Read-only	Record type. Possible values = ALL, ADNS, PROXY
__count	<Double>	Read-only	count parameter

Operations

(click to see Properties)

ADD| DELETE| UPDATE| UNSET| GET (ALL)| GET| COUNT

Some options that you can use for each operations:

- **Getting warnings in response:**NITRO allows you to get warnings in an operation by specifying the "warning" query parameter as "yes". For example, to get warnings while connecting to the Citrix ADC appliance, the URL is as follows:

  http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/login?warning=yes

  If any, the warnings are displayed in the response payload with the HTTP code "209 X-NITRO-WARNING".

- **Authenticated access for individual NITRO operations:**NITRO allows you to logon to the Citrix ADC appliance to perform individual operations. You can use this option instead of creating a NITRO session (using the login object) and then using that session to perform all operations,

  To do this, you must specify the username and password in the request header of the NITRO request as follows:

  X-NITRO-USER:<username>

  X-NITRO-PASS:<password>

  Note: In such cases, make sure that the request header DOES not include the following:

  Cookie:NITRO_AUTH_TOKEN=<tokenvalue>

`*Note:`*

Mandatory parameters are marked in redand placeholder content is marked in <green>.

add

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/dnsmxrec

HTTP Method:POST

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Content-Type:application/json

Request Payload:

```
{"dnsmxrec":{
```

```
<b>"domain":<String_value>,

</b><b>"mx":<String_value>,

</b><b>"pref":<Double_value>,

</b>"ttl":<Double_value>

}}
```

Response:

HTTP Status Code on Success: 201 Created HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error

delete

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/dnsmxrec/domain_value<String>

HTTP Method:DELETE

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue>

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error

update

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/dnsmxrec

HTTP Method:PUT

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Content-Type:application/json

Request Payload:

```
{"dnsmxrec":{

<b>"domain":<String_value>,

</b><b>"mx":<String_value>,

</b>"pref":<Double_value>,

"ttl":<Double_value>

}}
```

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error

unset

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/dnsmxrec?**action=unset**

HTTP Method:POST

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Content-Type:application/json

Request Payload:

```
{"dnsmxrec":{

<b>"domain":<String_value>,

</b><b>"mx":<String_value>,

</b>"ttl":true
```

```
}}
```

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error

get (all)

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/dnsmxrec

Query-parameters:

args

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/dnsmxrec?**args=domain:<String_value>,type:<String_value>,nodeid:<Double_value>**

Use this query-parameter to get dnsmxrec resources based on additional properties.

attrs

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/dnsmxrec?**attrs=property-name1,property-name2**

Use this query parameter to specify the resource details that you want to retrieve.

filter

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/dnsmxrec?**filter=property-name1:property-val1,property-name2:property-val2**

Use this query-parameter to get the filtered set of dnsmxrec resources configured on Citrix ADC. Filtering can be done on any of the properties of the resource.

view

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/dnsmxrec?**view=summary**

Note: By default, the retrieved results are displayed in detail view (?view=detail).

pagination

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/dnsmxrec?**pagesize=#no;pageno=#no**

Use this query-parameter to get the dnsmxrec resources in chunks.

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "dnsmxrec": [ {

domain:<String_value>,type:<String_value>,nodeid:<Double_value>"mx":<Strin
g_value>,

"pref":<Double_value>,

"ttl":<Double_value>,

"authtype":<String_value>,

"ecssubnet":<String_value>

}]}
```

get

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/dnsmxrec/domain_value<String>

Query-parameters:

attrs

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/dnsmxrec/domain_value<String>?**attrs=property-name1,property-name2**

Use this query parameter to specify the resource details that you want to retrieve.

view

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/dnsmxrec/domain_value<String>?**view=summary**

Note: By default, the retrieved results are displayed in detail view (?view=detail).

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "dnsmxrec": [ {

domain:<String_value>,type:<String_value>,nodeid:<Double_value>"mx":<Strin
g_value>,

"pref":<Double_value>,

"ttl":<Double_value>,

"authtype":<String_value>,

"ecssubnet":<String_value>

}]}
```

count

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/dnsmxrec?**count=yes**

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "dnsmxrec": [ { "__count": "#no"} ] }
```

dnsnameserver

Configuration for name server resource.

Properties

(click to see Operations)

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
ip	<String>	Read-write	IP address of an external name server or, if the Local parameter is set, IP address of a local DNS server (LDNS). Minimum length = 1

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
dnsvservername	<String>	Read-write	Name of a DNS virtual server. Overrides any IP address-based name servers configured on the Citrix ADC. Minimum length = 1
local	<Boolean>	Read-write	Mark the IP address as one that belongs to a local recursive DNS server on the Citrix ADC. The appliance recursively resolves queries received on an IP address that is marked as being local. For recursive resolution to work, the global DNS parameter, Recursion, must also be set. If no name server is marked as being local, the appliance functions as a stub resolver and load balances the name servers.
state	<String>	Read-write	Administrative state of the name server. Default value: ENABLED

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
			Possible values = ENABLED, DISABLED
type	<String>	Read-write	Protocol used by the name server. UDP_TCP is not valid if the name server is a DNS virtual server configured on the appliance. Default value: UDP Possible values = UDP, TCP, UDP_TCP
dnsprofilename	<String>	Read-write	Name of the DNS profile to be associated with the name server. Minimum length = 1
servicename	<String>	Read-only	The name of the dns vserver.
port	<Integer>	Read-only	Port of the service. Range 1 - 65535 * in CLI is represented as 65535 in NITRO API
nameserverstate	<String>	Read-only	State of the server. Possible values = UP, DOWN, UNKNOWN, BUSY, OUT OF SERVICE,

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
			GOING OUT OF SERVICE, DOWN WHEN GOING OUT OF SERVICE, NS_EMPTY_STR, Unknown, DISABLED
clmonowner	<Double>	Read-only	Tells the mon owner of the service.
clmonview	<Double>	Read-only	Tells the view id by which state of the service is updated.
__count	<Double>	Read-only	count parameter

Operations

(click to see Properties)

ADD| UPDATE| UNSET| DELETE| ENABLE| DISABLE| GET (ALL)| COUNT

Some options that you can use for each operations:

- **Getting warnings in response:**NITRO allows you to get warnings in an operation by specifying the "warning" query parameter as "yes". For example, to get warnings while connecting to the Citrix ADC appliance, the URL is as follows:

  http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/login?warning=yes

  If any, the warnings are displayed in the response payload with the HTTP code "209 X-NITRO-WARNING".

- **Authenticated access for individual NITRO operations:**NITRO allows you to logon to the Citrix ADC appliance to perform individual operations. You can use this option instead of creating a NITRO session (using the login object) and then using that session to perform all operations,

  To do this, you must specify the username and password in the request header of the NITRO request as follows:

X-NITRO-USER:<username>

X-NITRO-PASS:<password>

Note: In such cases, make sure that the request header DOES not include the following:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue>

`*Note:`*

Mandatory parameters are marked in redand placeholder content is marked in <green>.

add

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/dnsnameserver

HTTP Method:POST

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Content-Type:application/json

Request Payload:

```
{"dnsnameserver":{

"ip":<String_value>,

"dnsvservername":<String_value>,

"local":<Boolean_value>,

"state":<String_value>,

"type":<String_value>,

"dnsprofilename":<String_value>

}}
```

Response:

HTTP Status Code on Success: 201 Created HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error

update

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/dnsnameserver

HTTP Method:PUT

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Content-Type:application/json

Request Payload:

```
{"dnsnameserver":{

<b>"ip":<String_value>,

</b>"dnsprofilename":<String_value>,

"type":<String_value>

}}
```

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error

unset

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/dnsnameserver?**action=unset**

HTTP Method:POST

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Content-Type:application/json

Request Payload:

```
{"dnsnameserver":{

<b>"ip":<String_value>,

</b>"dnsprofilename":true,

"type":<String_value>

}}
```

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error

delete

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/dnsnameserver/ip_value<String>

HTTP Method:DELETE

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue>

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error

enable

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/dnsnameserver?**action=enable**

HTTP Method:POST

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Content-Type:application/json

Request Payload:

```
{"dnsnameserver":{

"ip":<String_value>,

"dnsvservername":<String_value>,

"type":<String_value>

}}
```

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error

disable

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/dnsnameserver?**action=disable**

HTTP Method:POST

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Content-Type:application/json

Request Payload:

```
{"dnsnameserver":{

"ip":<String_value>,

"dnsvservername":<String_value>,

"type":<String_value>
```

```
}}
```

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error

get (all)

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/dnsnameserver

Query-parameters:

attrs

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/dnsnameserver?**attrs=property-name1,property-name2**

Use this query parameter to specify the resource details that you want to retrieve.

filter

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/dnsnameserver?**filter=property-name1:property-val1,property-name2:property-val2**

Use this query-parameter to get the filtered set of dnsnameserver resources configured on Citrix ADC. Filtering can be done on any of the properties of the resource.

view

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/dnsnameserver?**view=summary**

Note: By default, the retrieved results are displayed in detail view (?view=detail).

pagination

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/dnsnameserver?**pagesize=#no;pageno=#no**

Use this query-parameter to get the dnsnameserver resources in chunks.

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "dnsnameserver": [ {

"ip":<String_value>,

"dnsvservername":<String_value>,

"servicename":<String_value>,

"port":<Integer_value>,

"type":<String_value>,

"state":<String_value>,

"nameserverstate":<String_value>,

"local":<Boolean_value>,

"clmonowner":<Double_value>,

"clmonview":<Double_value>,

"dnsprofilename":<String_value>

}]}
```

count

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/dnsnameserver?**count=yes**

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "dnsnameserver": [ { "__count": "#no"} ] }
```

dnsnaptrrec

Configuration for NAPTR record resource.

Properties

(click to see Operations)

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
domain	<String>	Read-write	Name of the domain for the NAPTR record. Minimum length = 1
order	<Double>	Read-write	An integer specifying the order in which the NAPTR records MUST be processed in order to accurately represent the ordered list of Rules.

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
			The ordering is from lowest to highest. Minimum value = 0 Maximum value = 65535
preference	<Double>	Read-write	An integer specifying the preference of this NAPTR among NAPTR records having same order. lower the number, higher the preference. Minimum value = 0 Maximum value = 65535
flags	<String>	Read-write	flags for this NAPTR. Maximum length = 255
services	<String>	Read-write	Service Parameters applicable to this delegation path. Maximum length = 255
regexp	<String>	Read-write	The regular expression, that specifies the substitution expression for this NAPTR. Maximum length = 255

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
replacement	<String>	Read-write	The replacement domain name for this NAPTR. Maximum length = 255
ttl	<Double>	Read-write	Time to Live (TTL), in seconds, for the record. TTL is the time for which the record must be cached by DNS proxies. The specified TTL is applied to all the resource records that are of the same record type and belong to the specified domain name. For example, if you add an address record, with a TTL of 36000, to the domain name example.com, the TTLs of all the address records of example.com are changed to 36000. If the TTL is not specified, the Citrix ADC uses either the DNS zone's minimum TTL or, if the SOA record is not available on the appliance, the default value of 3600.

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
			Default value: 3600 Minimum value = 0 Maximum value = 2147483647
recordid	<Double>	Read-write	Unique, internally generated record ID. View the details of the naptr record to obtain its record ID. Records can be removed by either specifying the domain name and record id OR by specifying domain name and all other naptr record attributes as was supplied during the add command. Minimum value = 1 Maximum value = 65535
ecssubnet	<String>	Read-write	Subnet for which the cached NAPTR record need to be removed.
type	<String>	Read-write	Type of records to display. Available settings function as follows:

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
			* ADNS - Display all authoritative address records. * PROXY - Display all proxy address records. * ALL - Display all address records. Default value: ADNS Possible values = ALL, ADNS, PROXY
nodeid	<Double>	Read-write	Unique number that identifies the cluster node. Minimum value = 0 Maximum value = 31
authtype	<String>	Read-only	Authentication type. Possible values = ALL, ADNS, PROXY
vservername	<String>	Read-only	Virtual server name.
__count	<Double>	Read-only	count parameter

Operations

(click to see Properties)

ADD| DELETE| GET (ALL)| GET| COUNT

Some options that you can use for each operations:

- **Getting warnings in response:**NITRO allows you to get warnings in an operation by specifying the "warning" query parameter as "yes". For example, to get warnings while connecting to the Citrix ADC appliance, the URL is as follows:

  http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/login?warning=yes

  If any, the warnings are displayed in the response payload with the HTTP code "209 X-NITRO-WARNING".

- **Authenticated access for individual NITRO operations:**NITRO allows you to logon to the Citrix ADC appliance to perform individual operations. You can use this option instead of creating a NITRO session (using the login object) and then using that session to perform all operations,

  To do this, you must specify the username and password in the request header of the NITRO request as follows:

  X-NITRO-USER:<username>

  X-NITRO-PASS:<password>

  Note: In such cases, make sure that the request header DOES not include the following:

  Cookie:NITRO_AUTH_TOKEN=<tokenvalue>

`*Note:`*

Mandatory parameters are marked in redand placeholder content is marked in <green>.

add

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/dnsnaptrrec

HTTP Method:POST

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Content-Type:application/json

Request Payload:

```
{"dnsnaptrrec":{

<b>"domain":<String_value>,

</b><b>"order":<Double_value>,

</b><b>"preference":<Double_value>,

</b>"flags":<String_value>,

"services":<String_value>,

"regexp":<String_value>,

"replacement":<String_value>,

"ttl":<Double_value>

}}
```

Response:

HTTP Status Code on Success: 201 Created HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error

delete

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/dnsnaptrrec/domain_value<String>

HTTP Method:DELETE

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue>

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error

get (all)

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/dnsnaptrrec

Query-parameters:

args

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/dnsnaptrrec?**args=domain:<String_value>,type:<String_value>,nodeid:<Double_value>**

Use this query-parameter to get dnsnaptrrec resources based on additional properties.

attrs

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/dnsnaptrrec?**attrs=property-name1,property-name2**

Use this query parameter to specify the resource details that you want to retrieve.

filter

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/dnsnaptrrec?**filter=property-name1:property-val1,property-name2:property-val2**

Use this query-parameter to get the filtered set of dnsnaptrrec resources configured on Citrix ADC. Filtering can be done on any of the properties of the resource.

view

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/dnsnaptrrec?**view=summary**

Note: By default, the retrieved results are displayed in detail view (?view=detail).

pagination

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/dnsnaptrrec?**pagesize=#no;pageno=#no**

Use this query-parameter to get the dnsnaptrrec resources in chunks.

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "dnsnaptrrec": [ {

domain:<String_value>,type:<String_value>,nodeid:<Double_value>"order":<Do
uble_value>,

"preference":<Double_value>,

"flags":<String_value>,

"services":<String_value>,

"regexp":<String_value>,

"replacement":<String_value>,

"ttl":<Double_value>,

"recordid":<Double_value>,

"authtype":<String_value>,

"ecssubnet":<String_value>,

"vservername":<String_value>

}]}
```

get

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/dnsnaptrrec/domain_value<String>

Query-parameters:

attrs

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/dnsnaptrrec/domain_value<String>?**attrs=property-name1,property-name2**

Use this query parameter to specify the resource details that you want to retrieve.

view

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/dnsnaptrrec/domain_value<String>?**view=summary**

Note: By default, the retrieved results are displayed in detail view (?view=detail).

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "dnsnaptrrec": [ {

domain:<String_value>,type:<String_value>,nodeid:<Double_value>"order":<Do
uble_value>,

"preference":<Double_value>,

"flags":<String_value>,

"services":<String_value>,

"regexp":<String_value>,
```

```
"replacement":<String_value>,

"ttl":<Double_value>,

"recordid":<Double_value>,

"authtype":<String_value>,

"ecssubnet":<String_value>,

"vservername":<String_value>

}]}
```

count

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/dnsnaptrrec?**count=yes**

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "dnsnaptrrec": [ { "__count": "#no"} ] }
```

dnsnsecrec

Configuration for 0 resource.

Properties

(click to see Operations)

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
hostname	<String>	Read-write	Name of the domain. Minimum length = 1
type	<String>	Read-write	Type of records to display. Available settings function as follows: * ADNS - Display all authoritative address records. * PROXY - Display all proxy address records. * ALL - Display all address records. Possible values = ALL, ADNS, PROXY
nextnsec	<String>	Read-only	Next nsec record in the chain.
nextrecs	<String[]>	Read-only	An array of record types associated with the nsec record. Possible values = A, NS, CNAME, SOA, MX, AAAA, SRV, RRSIG, NSEC, DNSKEY, PTR, TXT, NAPTR

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
ttl	<Double>	Read-only	Time to Live (TTL), in seconds, for the record.
ecssubnet	<String>	Read-only	Subnet for which this particular record is cached. Subnet caching will occur for responses with EDNS Client Subnet (ECS) option. Applies to resource records obtained through proxy configurations only.
__count	<Double>	Read-only	count parameter

Operations

(click to see Properties)

GET (ALL)| GET| COUNT

Some options that you can use for each operations:

- **Getting warnings in response:**NITRO allows you to get warnings in an operation by specifying the "warning" query parameter as "yes". For example, to get warnings while connecting to the Citrix ADC appliance, the URL is as follows:

  http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/login?warning=yes

  If any, the warnings are displayed in the response payload with the HTTP code "209 X-NITRO-WARNING".

- **Authenticated access for individual NITRO operations:**NITRO allows you to logon to the Citrix ADC appliance to perform individual operations. You can use this option instead of creating a NITRO session (using the login object) and then using that session to perform all operations,

  To do this, you must specify the username and password in the request header of the NITRO request as follows:

X-NITRO-USER:<username>

X-NITRO-PASS:<password>

Note: In such cases, make sure that the request header DOES not include the following:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue>

`*Note:`*

Mandatory parameters are marked in redand placeholder content is marked in <green>.

get (all)

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/dnsnsecrec

Query-parameters:

args

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/dnsnsecrec?**args=hostname:<String_value>,type:<String_value>**

Use this query-parameter to get dnsnsecrec resources based on additional properties.

attrs

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/dnsnsecrec?**attrs=property-name1,property-name2**

Use this query parameter to specify the resource details that you want to retrieve.

filter

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/dnsnsecrec?**filter=property-name1:property-val1,property-name2:property-val2**

Use this query-parameter to get the filtered set of dnsnsecrec resources configured on Citrix ADC. Filtering can be done on any of the properties of the resource.

view

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/dnsnsecrec?**view=summary**

Note: By default, the retrieved results are displayed in detail view (?view=detail).

pagination

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/dnsnsecrec?**pagesize=#no;pageno=#no**

Use this query-parameter to get the dnsnsecrec resources in chunks.

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "dnsnsecrec": [ {

hostname:<String_value>,type:<String_value>"nextnsec":<String_value>,

"nsecbitarray":<String[]_value>,

"nextrecs":<String[]_value>,

"ttl":<Double_value>,

"ecssubnet":<String_value>

}]}
```

get

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/dnsnsecrec/hostname_value<String>

Query-parameters:

attrs

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/dnsnsecrec/hostname_value<String>?**attrs=property-name1,property-name2**

Use this query parameter to specify the resource details that you want to retrieve.

view

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/dnsnsecrec/hostname_value<String>?**view=summary**

Note: By default, the retrieved results are displayed in detail view (?view=detail).

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "dnsnsecrec": [ {

hostname:<String_value>,type:<String_value>"nextnsec":<String_value>,

"nsecbitarray":<String[]_value>,

"nextrecs":<String[]_value>,

"ttl":<Double_value>,

"ecssubnet":<String_value>

}]}
```

count

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/dnsnsecrec?**count=yes**

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "dnsnsecrec": [ { "__count": "#no"} ] }
```

dnsnsrec

Configuration for name server record resource.

Properties

(click to see Operations)

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
domain	<String>	Read-write	Domain name. Minimum length = 1
nameserver	<String>	Read-write	Host name of the name server to add to the domain.

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
			Minimum length = 1
ttl	<Double>	Read-write	Time to Live (TTL), in seconds, for the record. TTL is the time for which the record must be cached by DNS proxies. The specified TTL is applied to all the resource records that are of the same record type and belong to the specified domain name. For example, if you add an address record, with a TTL of 36000, to the domain name example.com, the TTLs of all the address records of example.com are changed to 36000. If the TTL is not specified, the Citrix ADC uses either the DNS zone's minimum TTL or, if the SOA record is not available on the appliance, the default value of 3600. Default value: 3600 Minimum value = 0

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
			Maximum value = 2147483647
ecssubnet	<String>	Read-write	Subnet for which the cached name server record need to be removed.
type	<String>	Read-write	Type of records to display. Available settings function as follows: * ADNS - Display all authoritative address records. * PROXY - Display all proxy address records. * ALL - Display all address records. Possible values = ALL, ADNS, PROXY
nodeid	<Double>	Read-write	Unique number that identifies the cluster node. Minimum value = 0 Maximum value = 31
authtype	<String>	Read-only	Record type. Possible values = ALL, ADNS, PROXY

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
__count	<Double>	Read-only	count parameter

Operations

(click to see Properties)

ADD| DELETE| GET (ALL)| GET| COUNT

Some options that you can use for each operations:

- **Getting warnings in response:**NITRO allows you to get warnings in an operation by specifying the "warning" query parameter as "yes". For example, to get warnings while connecting to the Citrix ADC appliance, the URL is as follows:

  http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/login?warning=yes

  If any, the warnings are displayed in the response payload with the HTTP code "209 X-NITRO-WARNING".
- **Authenticated access for individual NITRO operations:**NITRO allows you to logon to the Citrix ADC appliance to perform individual operations. You can use this option instead of creating a NITRO session (using the login object) and then using that session to perform all operations,

  To do this, you must specify the username and password in the request header of the NITRO request as follows:

  X-NITRO-USER:<username>

  X-NITRO-PASS:<password>

  Note: In such cases, make sure that the request header DOES not include the following:

  Cookie:NITRO_AUTH_TOKEN=<tokenvalue>

`*Note:`*

Mandatory parameters are marked in redand placeholder content is marked in <green>.

add

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/dnsnsrec

HTTP Method:POST

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Content-Type:application/json

Request Payload:

```
{"dnsnsrec":{

<b>"domain":<String_value>,

</b><b>"nameserver":<String_value>,

</b>"ttl":<Double_value>

}}
```

Response:

HTTP Status Code on Success: 201 Created HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error

delete

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/dnsnsrec/domain_value<String>

HTTP Method:DELETE

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue>

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error

get (all)

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/dnsnsrec

Query-parameters:

args

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/dnsnsrec?**args=domain:<String_value>,type:<String_value>,nodeid:<Double_value>**

Use this query-parameter to get dnsnsrec resources based on additional properties.

attrs

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/dnsnsrec?**attrs=property-name1,property-name2**

Use this query parameter to specify the resource details that you want to retrieve.

filter

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/dnsnsrec?**filter=property-name1:property-val1,property-name2:property-val2**

Use this query-parameter to get the filtered set of dnsnsrec resources configured on Citrix ADC. Filtering can be done on any of the properties of the resource.

view

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/dnsnsrec?**view=summary**

Note: By default, the retrieved results are displayed in detail view (?view=detail).

pagination

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/dnsnsrec?**pagesize=#no;pageno=#no**

Use this query-parameter to get the dnsnsrec resources in chunks.

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "dnsnsrec": [ {

domain:<String_value>,type:<String_value>,nodeid:<Double_value>"nameserver
":<String_value>,

"ecssubnet":<String_value>,

"ttl":<Double_value>,

"authtype":<String_value>

}]}
```

get

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/dnsnsrec/domain_value<String>

Query-parameters:

attrs

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/dnsnsrec/domain_value<String>?**attrs=property-name1,property-name2**

Use this query parameter to specify the resource details that you want to retrieve.

view

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/dnsnsrec/domain_value<String>?**view=summary**

Note: By default, the retrieved results are displayed in detail view (?view=detail).

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "dnsnsrec": [ {

domain:<String_value>,type:<String_value>,nodeid:<Double_value>"nameserver
":<String_value>,

"ecssubnet":<String_value>,

"ttl":<Double_value>,

"authtype":<String_value>

}]}
```

count

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/dnsnsrec?**count=yes**

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "dnsnsrec": [ { "__count": "#no"} ] }
```

dnsparameter

Configuration for DNS parameter resource.

Properties

(click to see Operations)

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
retries	<Double>	Read-write	Maximum number of retry attempts when no response is received for a query sent to a name server. Applies to end resolver and forwarder configurations. Default value: 5 Minimum value = 1 Maximum value = 5
minttl	<Double>	Read-write	Minimum permissible time to live (TTL) for all records cached in the DNS cache by DNS proxy, end

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
			resolver, and forwarder configurations. If the TTL of a record that is to be cached is lower than the value configured for minTTL, the TTL of the record is set to the value of minTTL before caching. When you modify this setting, the new value is applied only to those records that are cached after the modification. The TTL values of existing records are not changed. Minimum value = 0 Maximum value = 604800
maxttl	<Double>	Read-write	Maximum time to live (TTL) for all records cached in the DNS cache by DNS proxy, end resolver, and forwarder configurations. If the TTL of a record that is to be cached is higher than the value configured for maxTTL, the TTL of the record is set to the value of maxTTL before caching. When you modify

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
			this setting, the new value is applied only to those records that are cached after the modification. The TTL values of existing records are not changed. Default value: 604800 Minimum value = 1 Maximum value = 604800
cacherecords	<String>	Read-write	Cache resource records in the DNS cache. Applies to resource records obtained through proxy configurations only. End resolver and forwarder configurations always cache records in the DNS cache, and you cannot disable this behavior. When you disable record caching, the appliance stops caching server responses. However, cached records are not flushed. The appliance does not serve requests from the cache until

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
			record caching is enabled again. Default value: YES Possible values = YES, NO
namelookuppriority	<String>	Read-write	Type of lookup (DNS or WINS) to attempt first. If the first-priority lookup fails, the second-priority lookup is attempted. Used only by the SSL VPN feature. Default value: WINS Possible values = WINS, DNS
recursion	<String>	Read-write	Function as an end resolver and recursively resolve queries for domains that are not hosted on the Citrix ADC. Also resolve queries recursively when the external name servers configured on the appliance (for a forwarder configuration) are unavailable. When external name servers are unavailable, the appliance queries a root server and resolves the request recursively, as it

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
			does for an end resolver configuration. Default value: DISABLED Possible values = ENABLED, DISABLED
resolutionorder	<String>	Read-write	Type of DNS queries (A, AAAA, or both) to generate during the routine functioning of certain Citrix ADC features, such as SSL VPN, cache redirection, and the integrated cache. The queries are sent to the external name servers that are configured for the forwarder function. If you specify both query types, you can also specify the order. Available settings function as follows: * OnlyAQuery. Send queries for IPv4 address records (A records) only. * OnlyAAAAQuery. Send queries for IPv6 address records (AAAA records) instead of

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
			queries for IPv4 address records (A records). * AThenAAAAQuery. Send a query for an A record, and then send a query for an AAAA record if the query for the A record results in a NODATA response from the name server. * AAAAThenAQuery. Send a query for an AAAA record, and then send a query for an A record if the query for the AAAA record results in a NODATA response from the name server. Default value: OnlyAQuery Possible values = OnlyAQuery, OnlyAAAAQuery, AThenAAAAQuery, AAAAThenAQuery

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
dnssec	<String>	Read-write	Enable or disable the Domain Name System Security Extensions (DNSSEC) feature on the appliance. Note: Even when the DNSSEC feature is enabled, forwarder configurations (used by internal Citrix ADC features such as SSL VPN and Cache Redirection for name resolution) do not support the DNSSEC OK (DO) bit in the EDNS0 OPT header. Default value: ENABLED Possible values = ENABLED, DISABLED
maxpipeline	<Double>	Read-write	Maximum number of concurrent DNS requests to allow on a single client connection, which is identified by the <clientip:port>-<vserver ip:port> tuple. A value of 0 (zero) applies no limit to the number of concurrent DNS requests allowed on

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
			a single client connection.
dnsrootreferral	<String>	Read-write	Send a root referral if a client queries a domain name that is unrelated to the domains configured/ cached on the Citrix ADC. If the setting is disabled, the appliance sends a blank response instead of a root referral. Applicable to domains for which the appliance is authoritative. Disable the parameter when the appliance is under attack from a client that is sending a flood of queries for unrelated domains. Default value: DISABLED Possible values = ENABLED, DISABLED
dns64timeout	<Double>	Read-write	While doing DNS64 resolution, this parameter specifies the time to wait before sending an A query if no response is received from backend DNS server for AAAA query.

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
			Minimum value = 0 Maximum value = 10000
ecsmaxsubnets	<Double>	Read-write	Maximum number of subnets that can be cached corresponding to a single domain. Subnet caching will occur for responses with EDNS Client Subnet (ECS) option. Caching of such responses can be disabled using DNS profile settings. A value of zero indicates that the number of subnets cached is limited only by existing memory constraints. The default value is zero. Default value: 0 Minimum value = 0 Maximum value = 1280
maxnegcachettl	<Double>	Read-write	Maximum time to live (TTL) for all negative records (NXDONAIN and NODATA) cached in the DNS cache by DNS proxy, end resolver, and

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
			forwarder configurations. If the TTL of a record that is to be cached is higher than the value configured for maxnegcacheTTL, the TTL of the record is set to the value of maxnegcacheTTL before caching. When you modify this setting, the new value is applied only to those records that are cached after the modification. The TTL values of existing records are not changed. Default value: 604800 Minimum value = 1 Maximum value = 604800
cachehitbypass	<String>	Read-write	This parameter is applicable only in proxy mode and if this parameter is enabled we will forward all the client requests to the backend DNS server and the response served will be cached on Citrix ADC.

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
			Default value: DISABLED Possible values = ENABLED, DISABLED
maxcachesize	<Double>	Read-write	Maximum memory, in megabytes, that can be used for dns caching per Packet Engine.
maxnegativecachesize	<Double>	Read-write	Maximum memory, in megabytes, that can be used for caching of negative DNS responses per packet engine.
cachenoexpire	<String>	Read-write	If this flag is set to YES, the existing entries in cache do not age out. On reaching the max limit the cache records are frozen. Default value: DISABLED Possible values = ENABLED, DISABLED
splitpktqueryprocessing	<String>	Read-write	Processing requests split across multiple packets. Default value: ALLOW

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
			Possible values = ALLOW, DROP
cacheecszeroprefix	<String>	Read-write	Cache ECS responses with a Scope Prefix length of zero. Such a cached response will be used for all queries with this domain name and any subnet. When disabled, ECS responses with Scope Prefix length of zero will be cached, but not tied to any subnet. This option has no effect if caching of ECS responses is disabled in the corresponding DNS profile. Default value: ENABLED Possible values = ENABLED, DISABLED
maxudppacketsize	<Double>	Read-write	Maximum UDP packet size that can be handled by Citrix ADC. This is the value advertised by Citrix ADC when responding as an authoritative server and it is also used when Citrix ADC

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
			queries other name servers as a forwarder. When acting as a proxy, requests from clients are limited by this parameter - if a request contains a size greater than this value in the OPT record, it will be replaced. Default value: 1280 Minimum value = 512 Maximum value = 16384

Operations

(click to see Properties)

UPDATE| UNSET| GET (ALL)

Some options that you can use for each operations:

- **Getting warnings in response:**NITRO allows you to get warnings in an operation by specifying the "warning" query parameter as "yes". For example, to get warnings while connecting to the Citrix ADC appliance, the URL is as follows:

  http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/login?warning=yes

  If any, the warnings are displayed in the response payload with the HTTP code "209 X-NITRO-WARNING".

- **Authenticated access for individual NITRO operations:**NITRO allows you to logon to the Citrix ADC appliance to perform individual operations. You can use this option instead of creating a NITRO session (using the login object) and then using that session to perform all operations,

To do this, you must specify the username and password in the request header of the NITRO request as follows:

X-NITRO-USER:<username>

X-NITRO-PASS:<password>

Note: In such cases, make sure that the request header DOES not include the following:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue>

`*Note:`*

Mandatory parameters are marked in redand placeholder content is marked in <green>.

update

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/dnsparameter

HTTP Method:PUT

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Content-Type:application/json

Request Payload:

```
{"dnsparameter":{

"retries":<Double_value>,

"minttl":<Double_value>,

"maxttl":<Double_value>,

"cacherecords":<String_value>,

"namelookuppriority":<String_value>,

"recursion":<String_value>,

"resolutionorder":<String_value>,
```

```
"dnssec":<String_value>,

"maxpipeline":<Double_value>,

"dnsrootreferral":<String_value>,

"dns64timeout":<Double_value>,

"ecsmaxsubnets":<Double_value>,

"maxnegcachettl":<Double_value>,

"cachehitbypass":<String_value>,

"maxcachesize":<Double_value>,

"maxnegativecachesize":<Double_value>,

"cachenoexpire":<String_value>,

"splitpktqueryprocessing":<String_value>,

"cacheecszeroprefix":<String_value>,

"maxudppacketsize":<Double_value>

}}
```

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error

unset

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/dnsparameter?**action=unset**

HTTP Method:POST

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Content-Type:application/json

Request Payload:

```
{"dnsparameter":{

"retries":true,

"minttl":true,

"maxttl":true,

"cacherecords":true,

"namelookuppriority":true,

"recursion":true,

"resolutionorder":true,

"dnssec":true,

"maxpipeline":true,

"dnsrootreferral":true,

"dns64timeout":true,

"ecsmaxsubnets":true,

"maxnegcachettl":true,

"cachehitbypass":true,

"maxcachesize":true,

"maxnegativecachesize":true,

"cachenoexpire":true,

"splitpktqueryprocessing":true,
```

```
"cacheecszeroprefix":true,

"maxudppacketsize":true

}}
```

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error

get (all)

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/dnsparameter

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "dnsparameter": [ {

"retries":<Double_value>,

"minttl":<Double_value>,

"maxttl":<Double_value>,
```

```
"namelookuppriority":<String_value>,

"cacherecords":<String_value>,

"recursion":<String_value>,

"resolutionorder":<String_value>,

"dnssec":<String_value>,

"maxpipeline":<Double_value>,

"dnsrootreferral":<String_value>,

"dns64timeout":<Double_value>,

"ecsmaxsubnets":<Double_value>,

"maxnegcachettl":<Double_value>,

"cachehitbypass":<String_value>,

"maxcachesize":<Double_value>,

"maxnegativecachesize":<Double_value>,

"cachenoexpire":<String_value>,

"splitpktqueryprocessing":<String_value>,

"cacheecszeroprefix":<String_value>,

"maxudppacketsize":<Double_value>

}]}
```

dnspolicy

Configuration for DNS policy resource.

Properties

(click to see Operations)

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
name	<String>	Read-write	Name for the DNS policy.
rule	<String>	Read-write	Expression against which DNS traffic is evaluated. Note: * On the command line interface, if the expression includes blank spaces, the entire expression must be enclosed in double quotation marks. * If the expression itself includes double quotation marks, you must escape the quotations by using the character. * Alternatively, you can use single quotation marks to enclose the rule, in which case you do not have to escape the double quotation marks.

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
			Example: CLIENT.UDP.DNS.DOMAIN.EQ("domainname").
viewname	<String>	Read-write	The view name that must be used for the given policy.
preferredlocation	<String>	Read-write	The location used for the given policy. This is deprecated attribute. Please use -prefLocList.
preferredloclist	<String[]>	Read-write	The location list in priority order used for the given policy. Minimum length = 1
drop	<String>	Read-write	The dns packet must be dropped. Possible values = YES, NO
cachebypass	<String>	Read-write	By pass dns cache for this. Possible values = YES, NO
actionname	<String>	Read-write	Name of the DNS action to perform when the rule evaluates to TRUE. The built in actions function as follows: * dns_default_act_Dr

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
			op. Drop the DNS request. * dns_default_act_Cachebypass. Bypass the DNS cache and forward the request to the name server. You can create custom actions by using the add dns action command in the CLI or the DNS > Actions > Create DNS Action dialog box in the Citrix ADC configuration utility.
logaction	<String>	Read-write	Name of the messagelog action to use for requests that match this policy.
hits	<Double>	Read-only	The number of times the policy has been hit.
undefhits	<Double>	Read-only	Number of Undef hits.
description	<String>	Read-only	Description of the policy.
__count	<Double>	Read-only	count parameter

Operations

(click to see Properties)

ADD| DELETE| UPDATE| UNSET| GET (ALL)| GET| COUNT

Some options that you can use for each operations:

- **Getting warnings in response:**NITRO allows you to get warnings in an operation by specifying the "warning" query parameter as "yes". For example, to get warnings while connecting to the Citrix ADC appliance, the URL is as follows:

  http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/login?warning=yes

  If any, the warnings are displayed in the response payload with the HTTP code "209 X-NITRO-WARNING".

- **Authenticated access for individual NITRO operations:**NITRO allows you to logon to the Citrix ADC appliance to perform individual operations. You can use this option instead of creating a NITRO session (using the login object) and then using that session to perform all operations,

  To do this, you must specify the username and password in the request header of the NITRO request as follows:

  X-NITRO-USER:<username>

  X-NITRO-PASS:<password>

  Note: In such cases, make sure that the request header DOES not include the following:

  Cookie:NITRO_AUTH_TOKEN=<tokenvalue>

`*Note:`*

Mandatory parameters are marked in redand placeholder content is marked in <green>.

add

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/dnspolicy

HTTP Method:POST

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Content-Type:application/json

Request Payload:

```
{"dnspolicy":{

<b>"name":<String_value>,

</b><b>"rule":<String_value>,

</b>"viewname":<String_value>,

"preferredlocation":<String_value>,

"preferredloclist":<String[]_value>,

"drop":<String_value>,

"cachebypass":<String_value>,

"actionname":<String_value>,

"logaction":<String_value>

}}
```

Response:

HTTP Status Code on Success: 201 Created HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error

delete

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/dnspolicy/name_value<String>

HTTP Method:DELETE

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue>

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error

update

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/dnspolicy

HTTP Method:PUT

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Content-Type:application/json

Request Payload:

```
{"dnspolicy":{

<b>"name":<String_value>,

</b>"rule":<String_value>,

"viewname":<String_value>,

"preferredlocation":<String_value>,

"preferredloclist":<String[]_value>,

"drop":<String_value>,

"cachebypass":<String_value>,

"actionname":<String_value>,

"logaction":<String_value>

}}
```

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error

unset

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/dnspolicy?**action=unset**

HTTP Method:POST

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Content-Type:application/json

Request Payload:

```
{"dnspolicy":{

<b>"name":<String_value>,

</b>"logaction":true

}}
```

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error

get (all)

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/dnspolicy

Query-parameters:

attrs

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/dnspolicy?**attrs=property-name1,property-name2**

Use this query parameter to specify the resource details that you want to retrieve.

filter

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/dnspolicy?**filter=property-name1:property-val1,property-name2:property-val2**

Use this query-parameter to get the filtered set of dnspolicy resources configured on Citrix ADC. Filtering can be done on any of the properties of the resource.

view

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/dnspolicy?**view=summary**

Note: By default, the retrieved results are displayed in detail view (?view=detail).

pagination

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/dnspolicy?**pagesize=#no;pageno=#no**

Use this query-parameter to get the dnspolicy resources in chunks.

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "dnspolicy": [ {

"name":<String_value>,

"rule":<String_value>,

"viewname":<String_value>,
```

```
"preferredlocation":<String_value>,

"preferredloclist":<String[]_value>,

"hits":<Double_value>,

"undefhits":<Double_value>,

"drop":<String_value>,

"actionname":<String_value>,

"cachebypass":<String_value>,

"description":<String_value>,

"logaction":<String_value>

}]}
```

get

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/dnspolicy/name_value<String>

Query-parameters:

attrs

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/dnspolicy/name_value<String>?**attrs=property-name1,property-name2**

Use this query parameter to specify the resource details that you want to retrieve.

view

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/dnspolicy/name_value<String>?**view=summary**

Note: By default, the retrieved results are displayed in detail view (?view=detail).

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "dnspolicy": [ {

"name":<String_value>,

"rule":<String_value>,

"viewname":<String_value>,

"preferredlocation":<String_value>,

"preferredloclist":<String[]_value>,

"hits":<Double_value>,

"undefhits":<Double_value>,

"drop":<String_value>,

"actionname":<String_value>,

"cachebypass":<String_value>,

"description":<String_value>,

"logaction":<String_value>

}]}
```

count

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/dnspolicy?**count=yes**

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "dnspolicy": [ { "__count": "#no"} ] }
```

dnspolicy64

Configuration for dns64 policy resource.

Properties

(click to see Operations)

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
name	<String>	Read-write	Name for the DNS64 policy.
rule	<String>	Read-write	Expression against which DNS traffic is evaluated. Note:

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
			* On the command line interface, if the expression includes blank spaces, the entire expression must be enclosed in double quotation marks. * If the expression itself includes double quotation marks, you must escape the quotations by using the character. * Alternatively, you can use single quotation marks to enclose the rule, in which case you do not have to escape the double quotation marks. Example: CLIENT.IP.SRC.IN_SUBENT(23.34.0.0/16).
action	<String>	Read-write	Name of the DNS64 action to perform when the rule evaluates to TRUE. The built in actions function as follows:

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
			* A default dns64 action with prefix <default prefix> and mapped and exclude are any You can create custom actions by using the add dns action command in the CLI or the DNS64 > Actions > Create DNS64 Action dialog box in the Citrix ADC configuration utility.
hits	<Double>	Read-only	The number of times the policy has been hit.
labeltype	<String>	Read-only	Type of policy label invocation. Possible values = reqvserver, resvserver, policylabel
labelname	<String>	Read-only	Name of the label to invoke if the current policy rule evaluates to TRUE.
undefhits	<Double>	Read-only	Number of Undef hits.
description	<String>	Read-only	Description of the policy.

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
__count	<Double>	Read-only	count parameter

Operations

(click to see Properties)

ADD| DELETE| UPDATE| GET (ALL)| GET| COUNT

Some options that you can use for each operations:

- **Getting warnings in response:**NITRO allows you to get warnings in an operation by specifying the "warning" query parameter as "yes". For example, to get warnings while connecting to the Citrix ADC appliance, the URL is as follows:

  http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/login?warning=yes

  If any, the warnings are displayed in the response payload with the HTTP code "209 X-NITRO-WARNING".
- **Authenticated access for individual NITRO operations:**NITRO allows you to logon to the Citrix ADC appliance to perform individual operations. You can use this option instead of creating a NITRO session (using the login object) and then using that session to perform all operations,

  To do this, you must specify the username and password in the request header of the NITRO request as follows:

  X-NITRO-USER:<username>

  X-NITRO-PASS:<password>

  Note: In such cases, make sure that the request header DOES not include the following:

  Cookie:NITRO_AUTH_TOKEN=<tokenvalue>

`*Note:`*

Mandatory parameters are marked in redand placeholder content is marked in <green>.

add

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/dnspolicy64

HTTP Method:POST

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Content-Type:application/json

Request Payload:

```
{"dnspolicy64":{

<b>"name":<String_value>,

</b><b>"rule":<String_value>,

</b>"action":<String_value>

}}
```

Response:

HTTP Status Code on Success: 201 Created HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error

delete

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/dnspolicy64/name_value<String>

HTTP Method:DELETE

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue>

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error

update

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/dnspolicy64

HTTP Method:PUT

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Content-Type:application/json

Request Payload:

```
{"dnspolicy64":{

<b>"name":<String_value>,

</b>"rule":<String_value>,

"action":<String_value>

}}
```

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error

get (all)

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/dnspolicy64

Query-parameters:

attrs

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/dnspolicy64?**attrs=property-name1,property-name2**

Use this query parameter to specify the resource details that you want to retrieve.

filter

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/dnspolicy64?**filter=property-name1:property-val1,property-name2:property-val2**

Use this query-parameter to get the filtered set of dnspolicy64 resources configured on Citrix ADC. Filtering can be done on any of the properties of the resource.

view

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/dnspolicy64?**view=summary**

Note: By default, the retrieved results are displayed in detail view (?view=detail).

pagination

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/dnspolicy64?**pagesize=#no;pageno=#no**

Use this query-parameter to get the dnspolicy64 resources in chunks.

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "dnspolicy64": [ {

"name":<String_value>,

"rule":<String_value>,

"hits":<Double_value>,

"action":<String_value>,

"labeltype":<String_value>,
```

```
"labelname":<String_value>,

"undefhits":<Double_value>,

"description":<String_value>

}]}
```

get

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/dnspolicy64/name_value<String>

Query-parameters:

attrs

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/dnspolicy64/name_value<String>?**attrs=property-name1,property-name2**

Use this query parameter to specify the resource details that you want to retrieve.

view

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/dnspolicy64/name_value<String>?**view=summary**

Note: By default, the retrieved results are displayed in detail view (?view=detail).

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "dnspolicy64": [ {

"name":<String_value>,

"rule":<String_value>,

"hits":<Double_value>,

"action":<String_value>,

"labeltype":<String_value>,

"labelname":<String_value>,

"undefhits":<Double_value>,

"description":<String_value>

}]}
```

count

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/dnspolicy64?**count=yes**

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "dnspolicy64": [ { "__count": "#no"} ] }
```

dnspolicy64_binding

Binding object which returns the resources bound to dnspolicy64.

Properties

(click to see Operations)

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
name	<String>	Read-write	Name of the DNS64 policy.
dnspolicy64_lbvserver_binding	<dnspolicy64_lbvserver_binding[]>	Read-only	lbvserver that can be bound to dnspolicy64.

Operations

(click to see Properties)

GET| GET (ALL)

Some options that you can use for each operations:

- **Getting warnings in response:**NITRO allows you to get warnings in an operation by specifying the "warning" query parameter as "yes". For example, to get warnings while connecting to the Citrix ADC appliance, the URL is as follows:

  http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/login?warning=yes

  If any, the warnings are displayed in the response payload with the HTTP code "209 X-NITRO-WARNING".
- **Authenticated access for individual NITRO operations:**NITRO allows you to logon to the Citrix ADC appliance to perform individual operations. You can use this option instead of creating a NITRO session (using the login object) and then using that session to perform all operations,

To do this, you must specify the username and password in the request header of the NITRO request as follows:

X-NITRO-USER:<username>

X-NITRO-PASS:<password>

Note: In such cases, make sure that the request header DOES not include the following:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue>

`*Note:`*

Mandatory parameters are marked in redand placeholder content is marked in <green>.

get

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/dnspolicy64_binding/name_value<String>

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "dnspolicy64_binding": [ {

"name":<String_value>,

"dnspolicy64_lbvserver_binding":<dnspolicy64_lbvserver_binding[]_value>

}]}
```

get (all)

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/dnspolicy64_binding

Query-parameters:

bulkbindings

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/dnspolicy64_binding?**bulkbindings=yes**

NITRO allows you to fetch bindings in bulk.

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "dnspolicy64_binding": [ {

"name":<String_value>,

"dnspolicy64_lbvserver_binding":<dnspolicy64_lbvserver_binding[]_value>

}]}
```

dnspolicy64_lbvserver_binding

Binding object showing the lbvserver that can be bound to dnspolicy64.

Properties

(click to see Operations)

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
boundto	<String>	Read-write	Location where policy is bound.
name	<String>	Read-write	Name of the DNS64 policy.
priority	<Double>	Read-only	Specifies the priority of the policy.
gotopriorityexpression	<String>	Read-only	Expression specifying the priority of the next policy which will get evaluated if the current policy rule evaluates to TRUE.
__count	<Double>	Read-only	count parameter

Operations

(click to see Properties)

GET| GET (ALL)| COUNT

Some options that you can use for each operations:

- **Getting warnings in response:**NITRO allows you to get warnings in an operation by specifying the "warning" query parameter as "yes". For example, to get warnings while connecting to the Citrix ADC appliance, the URL is as follows:

  http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/login?warning=yes

  If any, the warnings are displayed in the response payload with the HTTP code "209 X-NITRO-WARNING".
- **Authenticated access for individual NITRO operations:**NITRO allows you to logon to the Citrix ADC appliance to perform individual operations. You can use this option instead of

creating a NITRO session (using the login object) and then using that session to perform all operations,

To do this, you must specify the username and password in the request header of the NITRO request as follows:

X-NITRO-USER:<username>

X-NITRO-PASS:<password>

Note: In such cases, make sure that the request header DOES not include the following:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue>

`*Note:`*

Mandatory parameters are marked in redand placeholder content is marked in <green>.

get

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/dnspolicy64_lbvserver_binding/name_value<String>

Query-parameters:

filter

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/dnspolicy64_lbvserver_binding/name_value<String>?**filter=property-name1:property-value1,property-name2:property-value2**

Use this query-parameter to get the filtered set of dnspolicy64_lbvserver_binding resources configured on Citrix ADC. Filtering can be done on any of the properties of the resource.

pagination

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/dnspolicy64_lbvserver_binding/name_value<String>?**pagesize=#no;pageno=#no**

Use this query-parameter to get the dnspolicy64_lbvserver_binding resources in chunks.

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "dnspolicy64_lbvserver_binding": [ {

"boundto":<String_value>,

"name":<String_value>,

"priority":<Double_value>,

"gotopriorityexpression":<String_value>

}]}
```

get (all)

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/dnspolicy64_lbvserver_binding

Query-parameters:

bulkbindings

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/dnspolicy64_lbvserver_binding?**bulkbindings=yes**

NITRO allows you to fetch bindings in bulk.

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "dnspolicy64_lbvserver_binding": [ {

"boundto":<String_value>,

"name":<String_value>,

"priority":<Double_value>,

"gotopriorityexpression":<String_value>

}]}
```

count

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/dnspolicy64_lbvserver_binding/name_value<String>?**count=yes**

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{"dnspolicy64_lbvserver_binding": [ { "__count": "#no"} ] }
```

dnspolicylabel

Configuration for dns policy label resource.

Properties

(click to see Operations)

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
labelname	<String>	Read-write	Name of the dns policy label.
transform	<String>	Read-write	The type of transformations allowed by the policies bound to the label. Possible values = dns_req, dns_res
newname	<String>	Read-write	The new name of the dns policylabel. Minimum length = 1
numpol	<Double>	Read-only	Number of polices bound to label.
hits	<Double>	Read-only	Number of times policy label was invoked.
priority	<Double>	Read-only	Specifies the priority of the policy.
gotopriorityexpression	<String>	Read-only	Expression specifying the priority of the next policy

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
			which will get evaluated if the current policy rule evaluates to TRUE.
labeltype	<String>	Read-only	Type of policy label invocation. Possible values = policylabel
invoke_labelname	<String>	Read-only	Name of the label to invoke if the current policy rule evaluates to TRUE.
flowtype	<Double>	Read-only	Flowtype of the bound dns policy.
description	<String>	Read-only	Description of the policylabel.
isdefault	<Boolean>	Read-only	A value of true is returned if it is a default dns policylabel.
__count	<Double>	Read-only	count parameter

Operations

(click to see Properties)

ADD| DELETE| RENAME| GET (ALL)| GET| COUNT

Some options that you can use for each operations:

- **Getting warnings in response:**NITRO allows you to get warnings in an operation by specifying the "warning" query parameter as "yes". For example, to get warnings while connecting to the Citrix ADC appliance, the URL is as follows:

  http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/login?warning=yes

If any, the warnings are displayed in the response payload with the HTTP code "209 X-NITRO-WARNING".

- **Authenticated access for individual NITRO operations:**NITRO allows you to logon to the Citrix ADC appliance to perform individual operations. You can use this option instead of creating a NITRO session (using the login object) and then using that session to perform all operations,

  To do this, you must specify the username and password in the request header of the NITRO request as follows:

  X-NITRO-USER:<username>

  X-NITRO-PASS:<password>

  Note: In such cases, make sure that the request header DOES not include the following:

  Cookie:NITRO_AUTH_TOKEN=<tokenvalue>

`*Note:`*

Mandatory parameters are marked in redand placeholder content is marked in <green>.

add

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/dnspolicylabel

HTTP Method:POST

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Content-Type:application/json

Request Payload:

```
{"dnspolicylabel":{

<b>"labelname":<String_value>,

</b><b>"transform":<String_value>

</b>}}
```

Response:

HTTP Status Code on Success: 201 Created HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error

delete

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/dnspolicylabel/labelname_value<String>

HTTP Method:DELETE

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue>

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error

rename

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/dnspolicylabel?**action=rename**

HTTP Method:POST

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Content-Type:application/json

Request Payload:

```
{"dnspolicylabel":{

<b>"labelname":<String_value>,

</b><b>"newname":<String_value>

</b>}}
```

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error

get (all)

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/dnspolicylabel

Query-parameters:

attrs

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/dnspolicylabel?**attrs=property-name1,property-name2**

Use this query parameter to specify the resource details that you want to retrieve.

filter

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/dnspolicylabel?**filter=property-name1:property-val1,property-name2:property-val2**

Use this query-parameter to get the filtered set of dnspolicylabel resources configured on Citrix ADC. Filtering can be done on any of the properties of the resource.

view

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/dnspolicylabel?**view=summary**

Note: By default, the retrieved results are displayed in detail view (?view=detail).

pagination

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/dnspolicylabel?**pagesize=#no;pageno=#no**

Use this query-parameter to get the dnspolicylabel resources in chunks.

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "dnspolicylabel": [ {

"labelname":<String_value>,

"transform":<String_value>,

"numpol":<Double_value>,

"hits":<Double_value>,

"priority":<Double_value>,

"gotopriorityexpression":<String_value>,

"labeltype":<String_value>,

"invoke_labelname":<String_value>,

"flowtype":<Double_value>,

"description":<String_value>,

"isdefault":<Boolean_value>

}]}
```

get

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/dnspolicylabel/labelname_value<String>

Query-parameters:

attrs

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/dnspolicylabel/labelname_value<String>?**attrs=property-name1,property-name2**

Use this query parameter to specify the resource details that you want to retrieve.

view

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/dnspolicylabel/labelname_value<String>?**view=summary**

Note: By default, the retrieved results are displayed in detail view (?view=detail).

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "dnspolicylabel": [ {

"labelname":<String_value>,

"transform":<String_value>,

"numpol":<Double_value>,

"hits":<Double_value>,

"priority":<Double_value>,
```

```
"gotopriorityexpression":<String_value>,

"labeltype":<String_value>,

"invoke_labelname":<String_value>,

"flowtype":<Double_value>,

"description":<String_value>,

"isdefault":<Boolean_value>

}]}
```

count

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/dnspolicylabel?**count=yes**

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "dnspolicylabel": [ { "__count": "#no"} ] }
```

dnspolicylabel_binding

Binding object which returns the resources bound to dnspolicylabel.

Properties

(click to see Operations)

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
labelname	<String>	Read-write	Name of the dns policy label.
dnspolicylabel_policybinding_binding	<dnspolicylabel_policybinding_binding[]>	Read-only	policybinding that can be bound to dnspolicylabel.
dnspolicylabel_dnspolicy_binding	<dnspolicylabel_dnspolicy_binding[]>	Read-only	dnspolicy that can be bound to dnspolicylabel.

Operations

(click to see Properties)

GET| GET (ALL)

Some options that you can use for each operations:

- **Getting warnings in response:**NITRO allows you to get warnings in an operation by specifying the "warning" query parameter as "yes". For example, to get warnings while connecting to the Citrix ADC appliance, the URL is as follows:

  http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/login?warning=yes

  If any, the warnings are displayed in the response payload with the HTTP code "209 X-NITRO-WARNING".

- **Authenticated access for individual NITRO operations:**NITRO allows you to logon to the Citrix ADC appliance to perform individual operations. You can use this option instead of creating a NITRO session (using the login object) and then using that session to perform all operations,

  To do this, you must specify the username and password in the request header of the NITRO request as follows:

  X-NITRO-USER:<username>

  X-NITRO-PASS:<password>

Note: In such cases, make sure that the request header DOES not include the following:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue>

`*Note:`*

Mandatory parameters are marked in redand placeholder content is marked in <green>.

get

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/dnspolicylabel_binding/labelname_value<String>

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "dnspolicylabel_binding": [ {

"labelname":<String_value>,

"dnspolicylabel_policybinding_binding":<dnspolicylabel_policybinding_binding[]_value>,

"dnspolicylabel_dnspolicy_binding":<dnspolicylabel_dnspolicy_binding[]_value>

}]}
```

get (all)

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/dnspolicylabel_binding

Query-parameters:

bulkbindings

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/dnspolicylabel_binding?**bulkbindings=yes**

NITRO allows you to fetch bindings in bulk.

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "dnspolicylabel_binding": [ {

"labelname":<String_value>,

"dnspolicylabel_policybinding_binding":<dnspolicylabel_policybinding_bindi
ng[]_value>,

"dnspolicylabel_dnspolicy_binding":<dnspolicylabel_dnspolicy_binding[]_val
ue>

}]}
```

dnspolicylabel_dnspolicy_binding

Binding object showing the dnspolicy that can be bound to dnspolicylabel.

Properties

(click to see Operations)

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
priority	<Double>	Read-write	Specifies the priority of the policy.
gotopriorityexpression	<String>	Read-write	Expression specifying the priority of the next policy which will get evaluated if the current policy rule evaluates to TRUE.
policyname	<String>	Read-write	The dns policy name.
labeltype	<String>	Read-write	Type of policy label invocation. Possible values = policylabel
labelname	<String>	Read-write	Name of the dns policy label.
invoke_labelname	<String>	Read-write	Name of the label to invoke if the current policy rule evaluates to TRUE.
invoke	<Boolean>	Read-write	Invoke flag.
__count	<Double>	Read-write	count parameter

Operations

(click to see Properties)

ADD:| DELETE:| GET| GET (ALL)| COUNT

Some options that you can use for each operations:

- **Getting warnings in response:**NITRO allows you to get warnings in an operation by specifying the "warning" query parameter as "yes". For example, to get warnings while connecting to the Citrix ADC appliance, the URL is as follows:

  http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/login?warning=yes

  If any, the warnings are displayed in the response payload with the HTTP code "209 X-NITRO-WARNING".

- **Authenticated access for individual NITRO operations:**NITRO allows you to logon to the Citrix ADC appliance to perform individual operations. You can use this option instead of creating a NITRO session (using the login object) and then using that session to perform all operations,

  To do this, you must specify the username and password in the request header of the NITRO request as follows:

  X-NITRO-USER:<username>

  X-NITRO-PASS:<password>

  Note: In such cases, make sure that the request header DOES not include the following:

  Cookie:NITRO_AUTH_TOKEN=<tokenvalue>

`*Note:`*

Mandatory parameters are marked in redand placeholder content is marked in <green>.

add

URL:http://<Citrix-ADC-IP-address(NSIP)/nitro/v1/config/dnspolicylabel_dnspolicy_binding

HTTP Method:PUT

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Content-Type:application/json

Request Payload:

```
{

"dnspolicylabel_dnspolicy_binding":{

<b>"labelname":<String_value>,

</b><b>"policyname":<String_value>,

</b><b>"priority":<Double_value>,

</b>"gotopriorityexpression":<String_value>,

"invoke":<Boolean_value>,

"labeltype":<String_value>,

"invoke_labelname":<String_value>

}}
```

Response:

HTTP Status Code on Success: 201 Created HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error

delete

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/dnspolicylabel_dnspolicy_binding/labelname_value<String>

Query-parameters:

args

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/dnspolicylabel_dnspolicy_binding/labelname_value<String>?**args=policyname:<String_value>,priority:<Double_value>**

HTTP Method:DELETE

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue>

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error

get

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/dnspolicylabel_dnspolicy_binding/labelname_value<String>

Query-parameters:

filter

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/dnspolicylabel_dnspolicy_binding/labelname_value<String>?**filter=property-name1:property-value1,property-name2:property-value2**

Use this query-parameter to get the filtered set of dnspolicylabel_dnspolicy_binding resources configured on Citrix ADC. Filtering can be done on any of the properties of the resource.

pagination

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/dnspolicylabel_dnspolicy_binding/labelname_value<String>?**pagesize=#no;pageno=#no**

Use this query-parameter to get the dnspolicylabel_dnspolicy_binding resources in chunks.

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "dnspolicylabel_dnspolicy_binding": [ {

"priority":<Double_value>,

"gotopriorityexpression":<String_value>,

"policyname":<String_value>,

"labeltype":<String_value>,

"labelname":<String_value>,

"invoke_labelname":<String_value>,

"invoke":<Boolean_value>

}]}
```

get (all)

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/dnspolicylabel_dnspolicy_binding

Query-parameters:

bulkbindings

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/dnspolicylabel_dnspolicy_binding?**bulkbindings=yes**

NITRO allows you to fetch bindings in bulk.

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "dnspolicylabel_dnspolicy_binding": [ {

"priority":<Double_value>,

"gotopriorityexpression":<String_value>,

"policyname":<String_value>,

"labeltype":<String_value>,

"labelname":<String_value>,

"invoke_labelname":<String_value>,

"invoke":<Boolean_value>

}]}
```

count

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/dnspolicylabel_dnspolicy_binding/labelname_value<String>?**count=yes**

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{"dnspolicylabel_dnspolicy_binding": [ { "__count": "#no"} ] }
```

dnspolicylabel_policybinding_binding

Binding object showing the policybinding that can be bound to dnspolicylabel.

Properties

(click to see Operations)

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
priority	<Double>	Read-write	Specifies the priority of the policy.
gotopriorityexpression	<String>	Read-write	Expression specifying the priority of the next policy which will get evaluated if the current policy rule evaluates to TRUE.
policyname	<String>	Read-write	The dns policy name.
labeltype	<String>	Read-write	Type of policy label invocation. Possible values = policylabel

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
labelname	<String>	Read-write	Name of the dns policy label.
invoke_labelname	<String>	Read-write	Name of the label to invoke if the current policy rule evaluates to TRUE.
invoke	<Boolean>	Read-write	Invoke flag.
__count	<Double>	Read-write	count parameter

Operations

(click to see Properties)

GET| GET (ALL)| COUNT

Some options that you can use for each operations:

- **Getting warnings in response:**NITRO allows you to get warnings in an operation by specifying the "warning" query parameter as "yes". For example, to get warnings while connecting to the Citrix ADC appliance, the URL is as follows:

  http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/login?warning=yes

  If any, the warnings are displayed in the response payload with the HTTP code "209 X-NITRO-WARNING".

- **Authenticated access for individual NITRO operations:**NITRO allows you to logon to the Citrix ADC appliance to perform individual operations. You can use this option instead of creating a NITRO session (using the login object) and then using that session to perform all operations,

  To do this, you must specify the username and password in the request header of the NITRO request as follows:

  X-NITRO-USER:<username>

  X-NITRO-PASS:<password>

  Note: In such cases, make sure that the request header DOES not include the following:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue>

`*Note:`*

Mandatory parameters are marked in redand placeholder content is marked in <green>.

get

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/dnspolicylabel_policybinding_binding/labelname_value<String>

Query-parameters:

filter

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/dnspolicylabel_policybinding_binding/labelname_value<String>?**filter=property-name1:property-value1,property-name2:property-value2**

Use this query-parameter to get the filtered set of dnspolicylabel_policybinding_binding resources configured on Citrix ADC. Filtering can be done on any of the properties of the resource.

pagination

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/dnspolicylabel_policybinding_binding/labelname_value<String>?**pagesize=#no;pageno=#no**

Use this query-parameter to get the dnspolicylabel_policybinding_binding resources in chunks.

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "dnspolicylabel_policybinding_binding": [ {

"priority":<Double_value>,

"gotopriorityexpression":<String_value>,

"policyname":<String_value>,

"labeltype":<String_value>,

"labelname":<String_value>,

"invoke_labelname":<String_value>,

"invoke":<Boolean_value>

}]}
```

get (all)

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/dnspolicylabel_policybinding_binding

Query-parameters:

bulkbindings

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/dnspolicylabel_policybinding_binding?**bulkbindings=yes**

NITRO allows you to fetch bindings in bulk.

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "dnspolicylabel_policybinding_binding": [ {

"priority":<Double_value>,

"gotopriorityexpression":<String_value>,

"policyname":<String_value>,

"labeltype":<String_value>,

"labelname":<String_value>,

"invoke_labelname":<String_value>,

"invoke":<Boolean_value>

}]}
```

count

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/dnspolicylabel_policybinding_binding/labelname_value<String>?**count=yes**

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{"dnspolicylabel_policybinding_binding": [ { "__count": "#no"} ] }
```

dnspolicy_binding

Binding object which returns the resources bound to dnspolicy.

Properties

(click to see Operations)

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
name	<String>	Read-write	Name of the DNS policy.
dnspolicy_dnspolicylabel_binding	<dnspolicy_dnspolicylabel_binding[]>	Read-only	dnspolicylabel that can be bound to dnspolicy.
dnspolicy_dnsglobal_binding	<dnspolicy_dnsglobal_binding[]>	Read-only	dnsglobal that can be bound to dnspolicy.

Operations

(click to see Properties)

GET| GET (ALL)

Some options that you can use for each operations:

- **Getting warnings in response:**NITRO allows you to get warnings in an operation by specifying the "warning" query parameter as "yes". For example, to get warnings while connecting to the Citrix ADC appliance, the URL is as follows:

  http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/login?warning=yes

  If any, the warnings are displayed in the response payload with the HTTP code "209 X-NITRO-WARNING".

- **Authenticated access for individual NITRO operations:**NITRO allows you to logon to the Citrix ADC appliance to perform individual operations. You can use this option instead of creating a NITRO session (using the login object) and then using that session to perform all operations,

  To do this, you must specify the username and password in the request header of the NITRO request as follows:

  X-NITRO-USER:<username>

  X-NITRO-PASS:<password>

  Note: In such cases, make sure that the request header DOES not include the following:

  Cookie:NITRO_AUTH_TOKEN=<tokenvalue>

`*Note:`*

Mandatory parameters are marked in redand placeholder content is marked in <green>.

get

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/dnspolicy_binding/name_value<String>

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "dnspolicy_binding": [ {

"name":<String_value>,

"dnspolicy_dnsglobal_binding":<dnspolicy_dnsglobal_binding[]_value>,

"dnspolicy_dnspolicylabel_binding":<dnspolicy_dnspolicylabel_binding[]_val
ue>

}]}
```

get (all)

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/dnspolicy_binding

Query-parameters:

bulkbindings

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/dnspolicy_binding?**bulkbindings=yes**

NITRO allows you to fetch bindings in bulk.

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "dnspolicy_binding": [ {

"name":<String_value>,

"dnspolicy_dnsglobal_binding":<dnspolicy_dnsglobal_binding[]_value>,

"dnspolicy_dnspolicylabel_binding":<dnspolicy_dnspolicylabel_binding[]_val
ue>

}]}
```

dnspolicy_dnsglobal_binding

Binding object showing the dnsglobal that can be bound to dnspolicy.

Properties

(click to see Operations)

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
boundto	<String>	Read-write	Location where policy is bound.
name	<String>	Read-write	Name of the DNS policy.
priority	<Double>	Read-only	Specifies the priority of the policy.
labelname	<String>	Read-only	Name of the label to invoke if the current policy rule evaluates to TRUE.
gotopriorityexpression	<String>	Read-only	Expression specifying the priority of the next policy which will get evaluated if the

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
			current policy rule evaluates to TRUE.
labeltype	<String>	Read-only	Type of policy label invocation. Possible values = reqvserver, resvserver, policylabel
activepolicy	<Integer>	Read-only	Indicates whether policy is bound or not.
__count	<Double>	Read-only	count parameter

Operations

(click to see Properties)

GET| GET (ALL)| COUNT

Some options that you can use for each operations:

- **Getting warnings in response:**NITRO allows you to get warnings in an operation by specifying the "warning" query parameter as "yes". For example, to get warnings while connecting to the Citrix ADC appliance, the URL is as follows:

  http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/login?warning=yes

  If any, the warnings are displayed in the response payload with the HTTP code "209 X-NITRO-WARNING".

- **Authenticated access for individual NITRO operations:**NITRO allows you to logon to the Citrix ADC appliance to perform individual operations. You can use this option instead of creating a NITRO session (using the login object) and then using that session to perform all operations,

  To do this, you must specify the username and password in the request header of the NITRO request as follows:

  X-NITRO-USER:<username>

X-NITRO-PASS:<password>

Note: In such cases, make sure that the request header DOES not include the following:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue>

`*Note:`*

Mandatory parameters are marked in redand placeholder content is marked in <green>.

get

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/dnspolicy_dnsglobal_binding/name_value<String>

Query-parameters:

filter

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/dnspolicy_dnsglobal_binding/name_value<String>?**filter=property-name1:property-value1,property-name2:property-value2**

Use this query-parameter to get the filtered set of dnspolicy_dnsglobal_binding resources configured on Citrix ADC. Filtering can be done on any of the properties of the resource.

pagination

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/dnspolicy_dnsglobal_binding/name_value<String>?**pagesize=#no;pageno=#no**

Use this query-parameter to get the dnspolicy_dnsglobal_binding resources in chunks.

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "dnspolicy_dnsglobal_binding": [ {

"boundto":<String_value>,

"name":<String_value>,

"priority":<Double_value>,

"labelname":<String_value>,

"gotopriorityexpression":<String_value>,

"labeltype":<String_value>,

"activepolicy":<Integer_value>

}]}
```

get (all)

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/dnspolicy_dnsglobal_binding

Query-parameters:

bulkbindings

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/dnspolicy_dnsglobal_binding?**bulkbindings=yes**

NITRO allows you to fetch bindings in bulk.

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "dnspolicy_dnsglobal_binding": [ {

"boundto":<String_value>,

"name":<String_value>,

"priority":<Double_value>,

"labelname":<String_value>,

"gotopriorityexpression":<String_value>,

"labeltype":<String_value>,

"activepolicy":<Integer_value>

}]}
```

count

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/dnspolicy_dnsglobal_binding/name_value<String>?**count=yes**

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{"dnspolicy_dnsglobal_binding": [ { "__count": "#no"} ] }
```

dnspolicy_dnspolicylabel_binding

Binding object showing the dnspolicylabel that can be bound to dnspolicy.

Properties

(click to see Operations)

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
boundto	<String>	Read-write	Location where policy is bound.
name	<String>	Read-write	Name of the DNS policy.
priority	<Double>	Read-only	Specifies the priority of the policy.
labelname	<String>	Read-only	Name of the label to invoke if the current policy rule evaluates to TRUE.
gotopriorityexpression	<String>	Read-only	Expression specifying the priority of the next policy which will get evaluated if the

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
			current policy rule evaluates to TRUE.
labeltype	<String>	Read-only	Type of policy label invocation. Possible values = reqvserver, resvserver, policylabel
activepolicy	<Integer>	Read-only	Indicates whether policy is bound or not.
__count	<Double>	Read-only	count parameter

Operations

(click to see Properties)

GET| GET (ALL)| COUNT

Some options that you can use for each operations:

- **Getting warnings in response:**NITRO allows you to get warnings in an operation by specifying the "warning" query parameter as "yes". For example, to get warnings while connecting to the Citrix ADC appliance, the URL is as follows:

  http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/login?warning=yes

  If any, the warnings are displayed in the response payload with the HTTP code "209 X-NITRO-WARNING".

- **Authenticated access for individual NITRO operations:**NITRO allows you to logon to the Citrix ADC appliance to perform individual operations. You can use this option instead of creating a NITRO session (using the login object) and then using that session to perform all operations,

  To do this, you must specify the username and password in the request header of the NITRO request as follows:

  X-NITRO-USER:<username>

X-NITRO-PASS:<password>

Note: In such cases, make sure that the request header DOES not include the following:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue>

`*Note:`*

Mandatory parameters are marked in redand placeholder content is marked in <green>.

get

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/dnspolicy_dnspolicylabel_binding/name_value<String>

Query-parameters:

filter

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/dnspolicy_dnspolicylabel_binding/name_value<String>?**filter=property-name1:property-value1,property-name2:property-value2**

Use this query-parameter to get the filtered set of dnspolicy_dnspolicylabel_binding resources configured on Citrix ADC. Filtering can be done on any of the properties of the resource.

pagination

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/dnspolicy_dnspolicylabel_binding/name_value<String>?**pagesize=#no;pageno=#no**

Use this query-parameter to get the dnspolicy_dnspolicylabel_binding resources in chunks.

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "dnspolicy_dnspolicylabel_binding": [ {

"boundto":<String_value>,

"name":<String_value>,

"priority":<Double_value>,

"labelname":<String_value>,

"gotopriorityexpression":<String_value>,

"labeltype":<String_value>,

"activepolicy":<Integer_value>

}]}
```

get (all)

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/dnspolicy_dnspolicylabel_binding

Query-parameters:

bulkbindings

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/dnspolicy_dnspolicylabel_binding?**bulkbindings=yes**

NITRO allows you to fetch bindings in bulk.

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "dnspolicy_dnspolicylabel_binding": [ {

"boundto":<String_value>,

"name":<String_value>,

"priority":<Double_value>,

"labelname":<String_value>,

"gotopriorityexpression":<String_value>,

"labeltype":<String_value>,

"activepolicy":<Integer_value>

}]}
```

count

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/dnspolicy_dnspolicylabel_binding/name_value<String>?**count=yes**

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{"dnspolicy_dnspolicylabel_binding": [ { "__count": "#no"} ] }
```

dnsprofile

Configuration for DNS profile resource.

Properties

(click to see Operations)

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
dnsprofilename	<String>	Read-write	Name of the DNS profile. Minimum length = 1 Maximum length = 127
dnsquerylogging	<String>	Read-write	DNS query logging; if enabled, DNS query information such as DNS query id, DNS query flags , DNS domain name and DNS query type will be logged. Default value: DISABLED

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
			Possible values = ENABLED, DISABLED
dnsanswersecloggin g	<String>	Read-write	DNS answer section; if enabled, answer section in the response will be logged. Default value: DISABLED Possible values = ENABLED, DISABLED
dnsextendedlogging	<String>	Read-write	DNS extended logging; if enabled, authority and additional section in the response will be logged. Default value: DISABLED Possible values = ENABLED, DISABLED
dnserrorlogging	<String>	Read-write	DNS error logging; if enabled, whenever error is encountered in DNS module reason for the error will be logged. Default value: DISABLED

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
			Possible values = ENABLED, DISABLED
cacherecords	<String>	Read-write	Cache resource records in the DNS cache. Applies to resource records obtained through proxy configurations only. End resolver and forwarder configurations always cache records in the DNS cache, and you cannot disable this behavior. When you disable record caching, the appliance stops caching server responses. However, cached records are not flushed. The appliance does not serve requests from the cache until record caching is enabled again. Default value: ENABLED Possible values = ENABLED, DISABLED
cachenegativerespo nses	<String>	Read-write	Cache negative responses in the DNS cache. When

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
			disabled, the appliance stops caching negative responses except referral records. This applies to all configurations - proxy, end resolver, and forwarder. However, cached responses are not flushed. The appliance does not serve negative responses from the cache until this parameter is enabled again. Default value: ENABLED Possible values = ENABLED, DISABLED
dropmultiqueryrequest	<String>	Read-write	Drop the DNS requests containing multiple queries. When enabled, DNS requests containing multiple queries will be dropped. In case of proxy configuration by default the DNS request containing multiple queries is forwarded to the backend and in case of ADNS and Resolver

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
			configuration NOCODE error response will be sent to the client. Default value: DISABLED Possible values = ENABLED, DISABLED
cacheecsresponses	<String>	Read-write	Cache DNS responses with EDNS Client Subnet(ECS) option in the DNS cache. When disabled, the appliance stops caching responses with ECS option. This is relevant to proxy configuration. Enabling/disabling support of ECS option when Citrix ADC is authoritative for a GSLB domain is supported using a knob in GSLB vserver. In all other modes, ECS option is ignored. Default value: DISABLED Possible values = ENABLED, DISABLED

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
referencecount	<Double>	Read-only	Number of entities using this profile.
__count	<Double>	Read-only	count parameter

Operations

(click to see Properties)

ADD| UPDATE| UNSET| DELETE| GET (ALL)| GET| COUNT

Some options that you can use for each operations:

- **Getting warnings in response:**NITRO allows you to get warnings in an operation by specifying the "warning" query parameter as "yes". For example, to get warnings while connecting to the Citrix ADC appliance, the URL is as follows:

  http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/login?warning=yes

  If any, the warnings are displayed in the response payload with the HTTP code "209 X-NITRO-WARNING".

- **Authenticated access for individual NITRO operations:**NITRO allows you to logon to the Citrix ADC appliance to perform individual operations. You can use this option instead of creating a NITRO session (using the login object) and then using that session to perform all operations,

  To do this, you must specify the username and password in the request header of the NITRO request as follows:

  X-NITRO-USER:<username>

  X-NITRO-PASS:<password>

  Note: In such cases, make sure that the request header DOES not include the following:

  Cookie:NITRO_AUTH_TOKEN=<tokenvalue>

`*Note:`*

Mandatory parameters are marked in redand placeholder content is marked in <green>.

add

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/dnsprofile

HTTP Method:POST

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Content-Type:application/json

Request Payload:

```
{"dnsprofile":{

<b>"dnsprofilename":<String_value>,

</b>"dnsquerylogging":<String_value>,

"dnsanswerseclogging":<String_value>,

"dnsextendedlogging":<String_value>,

"dnserrorlogging":<String_value>,

"cacherecords":<String_value>,

"cachenegativeresponses":<String_value>,

"dropmultiqueryrequest":<String_value>,

"cacheecsresponses":<String_value>

}}
```

Response:

HTTP Status Code on Success: 201 Created HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error

update

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/dnsprofile

HTTP Method:PUT

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Content-Type:application/json

Request Payload:

```
{"dnsprofile":{

<b>"dnsprofilename":<String_value>,

</b>"dnsquerylogging":<String_value>,

"dnsanswerseclogging":<String_value>,

"dnsextendedlogging":<String_value>,

"dnserrorlogging":<String_value>,

"cacherecords":<String_value>,

"cachenegativeresponses":<String_value>,

"dropmultiqueryrequest":<String_value>,

"cacheecsresponses":<String_value>

}}
```

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error

unset

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/dnsprofile?**action=unset**

HTTP Method:POST

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Content-Type:application/json

Request Payload:

```
{"dnsprofile":{

<b>"dnsprofilename":<String_value>,

</b>"dnsquerylogging":true,

"dnsanswerseclogging":true,

"dnsextendedlogging":true,

"dnserrorlogging":true,

"cacherecords":true,

"cachenegativeresponses":true,

"dropmultiqueryrequest":true,

"cacheecsresponses":true

}}
```

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error

delete

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/dnsprofile/dnsprofilename_value<String>

HTTP Method:DELETE

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue>

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error

get (all)

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/dnsprofile

Query-parameters:

attrs

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/dnsprofile?**attrs=property-name1,property-name2**

Use this query parameter to specify the resource details that you want to retrieve.

filter

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/dnsprofile?**filter=property-name1:property-val1,property-name2:property-val2**

Use this query-parameter to get the filtered set of dnsprofile resources configured on Citrix ADC. Filtering can be done on any of the properties of the resource.

view

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/dnsprofile?**view=summary**

Note: By default, the retrieved results are displayed in detail view (?view=detail).

pagination

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/dnsprofile?**pagesize=#no;pageno=#no**

Use this query-parameter to get the dnsprofile resources in chunks.

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "dnsprofile": [ {

"dnsprofilename":<String_value>,

"dnsquerylogging":<String_value>,

"dnsanswerseclogging":<String_value>,

"dnsextendedlogging":<String_value>,

"dnserrorlogging":<String_value>,

"cacherecords":<String_value>,

"cachenegativeresponses":<String_value>,

"dropmultiqueryrequest":<String_value>,

"cacheecsresponses":<String_value>,

"referencecount":<Double_value>

}]}
```

get

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/dnsprofile/dnsprofilename_value<String>

Query-parameters:

attrs

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/dnsprofile/dnsprofilename_value<String>?**attrs=property-name1,property-name2**

Use this query parameter to specify the resource details that you want to retrieve.

view

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/dnsprofile/dnsprofilename_value<String>?**view=summary**

Note: By default, the retrieved results are displayed in detail view (?view=detail).

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "dnsprofile": [ {

"dnsprofilename":<String_value>,
```

```
"dnsquerylogging":<String_value>,

"dnsanswerseclogging":<String_value>,

"dnsextendedlogging":<String_value>,

"dnserrorlogging":<String_value>,

"cacherecords":<String_value>,

"cachenegativeresponses":<String_value>,

"dropmultiqueryrequest":<String_value>,

"cacheecsresponses":<String_value>,

"referencecount":<Double_value>

}]}
```

count

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/dnsprofile?**count=yes**

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "dnsprofile": [ { "__count": "#no"} ] }
```

dnsproxyrecords

Configuration for proxy record resource.

Properties

(click to see Operations)

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
type	<String>	Read-write	Filter the DNS records to be flushed.e.g flush dns proxyRecords -type A will flush only the A records from the cache. . Possible values = A, NS, CNAME, SOA, MX, AAAA, SRV, RRSIG, NSEC, DNSKEY, PTR, TXT, NAPTR
negrectype	<String>	Read-write	Filter the Negative DNS records i.e NXDOMAIN and NODATA entries to be flushed. e.g flush dns proxyRecords NXDOMAIN will flush only the NXDOMAIN entries from the cache.

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
			Possible values = NXDOMAIN, NODATA

Operations

(click to see Properties)

FLUSH

Some options that you can use for each operations:

- **Getting warnings in response:**NITRO allows you to get warnings in an operation by specifying the "warning" query parameter as "yes". For example, to get warnings while connecting to the Citrix ADC appliance, the URL is as follows:

  http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/login?warning=yes

  If any, the warnings are displayed in the response payload with the HTTP code "209 X-NITRO-WARNING".

- **Authenticated access for individual NITRO operations:**NITRO allows you to logon to the Citrix ADC appliance to perform individual operations. You can use this option instead of creating a NITRO session (using the login object) and then using that session to perform all operations,

  To do this, you must specify the username and password in the request header of the NITRO request as follows:

  X-NITRO-USER:<username>

  X-NITRO-PASS:<password>

  Note: In such cases, make sure that the request header DOES not include the following:

  Cookie:NITRO_AUTH_TOKEN=<tokenvalue>

`*Note:` *

Mandatory parameters are marked in redand placeholder content is marked in <green>.

flush

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/dnsproxyrecords?**action=flush**

HTTP Method:POST

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Content-Type:application/json

Request Payload:

```
{"dnsproxyrecords":{

"type":<String_value>,

"negrectype":<String_value>

}}
```

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error

dnsptrrec

Configuration for PTR record resource.

Properties

(click to see Operations)

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
reversedomain	<String>	Read-write	Reversed domain name representation of the IPv4 or IPv6 address for which to create the PTR

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
			record. Use the "in-addr.arpa." suffix for IPv4 addresses and the "ip6.arpa." suffix for IPv6 addresses. Minimum length = 1
domain	<String>	Read-write	Domain name for which to configure reverse mapping. Minimum length = 1
ttl	<Double>	Read-write	Time to Live (TTL), in seconds, for the record. TTL is the time for which the record must be cached by DNS proxies. The specified TTL is applied to all the resource records that are of the same record type and belong to the specified domain name. For example, if you add an address record, with a TTL of 36000, to the domain name example.com, the TTLs of all the address records of example.com are changed to 36000. If the TTL is not specified, the Citrix ADC uses either the DNS zone's minimum TTL or, if

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
			the SOA record is not available on the appliance, the default value of 3600. Default value: 3600 Minimum value = 0 Maximum value = 2147483647
ecssubnet	<String>	Read-write	Subnet for which the cached PTR record need to be removed.
type	<String>	Read-write	Type of records to display. Available settings function as follows: * ADNS - Display all authoritative address records. * PROXY - Display all proxy address records. * ALL - Display all address records. Possible values = ALL, ADNS, PROXY
nodeid	<Double>	Read-write	Unique number that identifies the cluster node. Minimum value = 0

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
			Maximum value = 31
authtype	<String>	Read-only	Authentication type. Possible values = ALL, ADNS, PROXY
__count	<Double>	Read-only	count parameter

Operations

(click to see Properties)

ADD| DELETE| GET (ALL)| GET| COUNT

Some options that you can use for each operations:

- **Getting warnings in response:**NITRO allows you to get warnings in an operation by specifying the "warning" query parameter as "yes". For example, to get warnings while connecting to the Citrix ADC appliance, the URL is as follows:

  http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/login?warning=yes

  If any, the warnings are displayed in the response payload with the HTTP code "209 X-NITRO-WARNING".
- **Authenticated access for individual NITRO operations:**NITRO allows you to logon to the Citrix ADC appliance to perform individual operations. You can use this option instead of creating a NITRO session (using the login object) and then using that session to perform all operations,

  To do this, you must specify the username and password in the request header of the NITRO request as follows:

  X-NITRO-USER:<username>

  X-NITRO-PASS:<password>

  Note: In such cases, make sure that the request header DOES not include the following:

  Cookie:NITRO_AUTH_TOKEN=<tokenvalue>

`*Note:` *

Mandatory parameters are marked in redand placeholder content is marked in <green>.

add

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/dnsptrrec

HTTP Method:POST

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Content-Type:application/json

Request Payload:

```
{"dnsptrrec":{

<b>"reversedomain":<String_value>,

</b><b>"domain":<String_value>,

</b>"ttl":<Double_value>

}}
```

Response:

HTTP Status Code on Success: 201 Created HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error

delete

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/dnsptrrec/reversedomain_value<String>

HTTP Method:DELETE

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue>

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error

get (all)

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/dnsptrrec

Query-parameters:

args

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/dnsptrrec?**args=reversedomain:<String_value>,type:<String_value>,nodeid:<Double_value>**

Use this query-parameter to get dnsptrrec resources based on additional properties.

attrs

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/dnsptrrec?**attrs=property-name1,property-name2**

Use this query parameter to specify the resource details that you want to retrieve.

filter

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/dnsptrrec?**filter=property-name1:property-val1,property-name2:property-val2**

Use this query-parameter to get the filtered set of dnsptrrec resources configured on Citrix ADC. Filtering can be done on any of the properties of the resource.

view

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/dnsptrrec?**view=summary**

Note: By default, the retrieved results are displayed in detail view (?view=detail).

pagination

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/dnsptrrec?**pagesize=#no;pageno=#no**

Use this query-parameter to get the dnsptrrec resources in chunks.

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "dnsptrrec": [ {

reversedomain:<String_value>,type:<String_value>,nodeid:<Double_value>"dom
ain":<String_value>,

"ttl":<Double_value>,

"authtype":<String_value>,

"ecssubnet":<String_value>

}]}
```

get

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/dnsptrrec/reversedomain_value<String>

Query-parameters:

attrs

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/dnsptrrec/reversedomain_value<String>?**attrs=property-name1,property-name2**

Use this query parameter to specify the resource details that you want to retrieve.

view

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/dnsptrrec/reversedomain_value<String>?**view=summary**

Note: By default, the retrieved results are displayed in detail view (?view=detail).

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "dnsptrrec": [ {

reversedomain:<String_value>,type:<String_value>,nodeid:<Double_value>"dom
ain":<String_value>,

"ttl":<Double_value>,

"authtype":<String_value>,

"ecssubnet":<String_value>

}]}
```

count

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/dnsptrrec?**count=yes**

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "dnsptrrec": [ { "__count": "#no"} ] }
```

dnssoarec

Configuration for SOA record resource.

Properties

(click to see Operations)

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
domain	<String>	Read-write	Domain name for which to add the SOA record. Minimum length = 1
originserver	<String>	Read-write	Domain name of the name server that responds authoritatively for the domain. Minimum length = 1
contact	<String>	Read-write	Email address of the contact to whom domain issues can be addressed. In the

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
			email address, replace the @ sign with a period (.). For example, enter domainadmin.example.com instead of domainadmin@example.com. Minimum length = 1
serial	<Double>	Read-write	The secondary server uses this parameter to determine whether it requires a zone transfer from the primary server. Default value: 100 Minimum value = 0 Maximum value = 4294967294
refresh	<Double>	Read-write	Time, in seconds, for which a secondary server must wait between successive checks on the value of the serial number. Default value: 3600 Minimum value = 0 Maximum value = 4294967294
retry	<Double>	Read-write	Time, in seconds, between retries if a secondary server's attempt to contact

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
			the primary server for a zone refresh fails. Default value: 3 Minimum value = 0 Maximum value = 4294967294
expire	<Double>	Read-write	Time, in seconds, after which the zone data on a secondary name server can no longer be considered authoritative because all refresh and retry attempts made during the period have failed. After the expiry period, the secondary server stops serving the zone. Typically one week. Not used by the primary server. Default value: 3600 Minimum value = 0 Maximum value = 4294967294
minimum	<Double>	Read-write	Default time to live (TTL) for all records in the zone. Can be overridden for individual records. Default value: 5

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
			Minimum value = 0 Maximum value = 2147483647
ttl	<Double>	Read-write	Time to Live (TTL), in seconds, for the record. TTL is the time for which the record must be cached by DNS proxies. The specified TTL is applied to all the resource records that are of the same record type and belong to the specified domain name. For example, if you add an address record, with a TTL of 36000, to the domain name example.com, the TTLs of all the address records of example.com are changed to 36000. If the TTL is not specified, the Citrix ADC uses either the DNS zone's minimum TTL or, if the SOA record is not available on the appliance, the default value of 3600. Default value: 3600

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
			Minimum value = 0 Maximum value = 2147483647
ecssubnet	<String>	Read-write	Subnet for which the cached SOA record need to be removed.
type	<String>	Read-write	Type of records to display. Available settings function as follows: * ADNS - Display all authoritative address records. * PROXY - Display all proxy address records. * ALL - Display all address records. Possible values = ALL, ADNS, PROXY
nodeid	<Double>	Read-write	Unique number that identifies the cluster node. Minimum value = 0 Maximum value = 31
authtype	<String>	Read-only	Record type. Possible values = ALL, ADNS, PROXY

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
__count	<Double>	Read-only	count parameter

Operations

(click to see Properties)

ADD| DELETE| UPDATE| UNSET| GET (ALL)| GET| COUNT

Some options that you can use for each operations:

- **Getting warnings in response:**NITRO allows you to get warnings in an operation by specifying the "warning" query parameter as "yes". For example, to get warnings while connecting to the Citrix ADC appliance, the URL is as follows:

  http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/login?warning=yes

  If any, the warnings are displayed in the response payload with the HTTP code "209 X-NITRO-WARNING".

- **Authenticated access for individual NITRO operations:**NITRO allows you to logon to the Citrix ADC appliance to perform individual operations. You can use this option instead of creating a NITRO session (using the login object) and then using that session to perform all operations,

  To do this, you must specify the username and password in the request header of the NITRO request as follows:

  X-NITRO-USER:<username>

  X-NITRO-PASS:<password>

  Note: In such cases, make sure that the request header DOES not include the following:

  Cookie:NITRO_AUTH_TOKEN=<tokenvalue>

`*Note:`*

Mandatory parameters are marked in redand placeholder content is marked in <green>.

add

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/dnssoarec

HTTP Method:POST

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Content-Type:application/json

Request Payload:

```
{"dnssoarec":{

<b>"domain":<String_value>,

</b><b>"originserver":<String_value>,

</b><b>"contact":<String_value>,

</b>"serial":<Double_value>,

"refresh":<Double_value>,

"retry":<Double_value>,

"expire":<Double_value>,

"minimum":<Double_value>,

"ttl":<Double_value>

}}
```

Response:

HTTP Status Code on Success: 201 Created HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error

delete

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/dnssoarec/domain_value<String>

HTTP Method:DELETE

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue>

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error

update

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/dnssoarec

HTTP Method:PUT

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Content-Type:application/json

Request Payload:

```
{"dnssoarec":{

<b>"domain":<String_value>,

</b>"originserver":<String_value>,

"contact":<String_value>,

"serial":<Double_value>,

"refresh":<Double_value>,

"retry":<Double_value>,

"expire":<Double_value>,

"minimum":<Double_value>,

"ttl":<Double_value>

}}
```

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error

unset

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/dnssoarec?**action=unset**

HTTP Method:POST

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Content-Type:application/json

Request Payload:

```
{"dnssoarec":{

<b>"domain":<String_value>,

</b>"serial":true,

"refresh":true,

"retry":true,

"expire":true,

"minimum":true,

"ttl":true

}}
```

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error

get (all)

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/dnssoarec

Query-parameters:

args

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/dnssoarec?**args=domain:<String_value>,type:<String_value>,nodeid:<Double_value>**

Use this query-parameter to get dnssoarec resources based on additional properties.

attrs

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/dnssoarec?**attrs=property-name1,property-name2**

Use this query parameter to specify the resource details that you want to retrieve.

filter

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/dnssoarec?**filter=property-name1:property-val1,property-name2:property-val2**

Use this query-parameter to get the filtered set of dnssoarec resources configured on Citrix ADC. Filtering can be done on any of the properties of the resource.

view

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/dnssoarec?**view=summary**

Note: By default, the retrieved results are displayed in detail view (?view=detail).

pagination

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/dnssoarec?**pagesize=#no;pageno=#no**

Use this query-parameter to get the dnssoarec resources in chunks.

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "dnssoarec": [ {

domain:<String_value>,type:<String_value>,nodeid:<Double_value>"originserv
er":<String_value>,

"contact":<String_value>,

"serial":<Double_value>,

"refresh":<Double_value>,

"retry":<Double_value>,

"expire":<Double_value>,

"minimum":<Double_value>,

"ttl":<Double_value>,

"authtype":<String_value>,

"ecssubnet":<String_value>

}]}
```

get

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/dnssoarec/domain_value<String>

Query-parameters:

attrs

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/dnssoarec/domain_value<String>?**attrs=property-name1,property-name2**

Use this query parameter to specify the resource details that you want to retrieve.

view

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/dnssoarec/domain_value<String>?**view=summary**

Note: By default, the retrieved results are displayed in detail view (?view=detail).

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "dnssoarec": [ {

domain:<String_value>,type:<String_value>,nodeid:<Double_value>"originserv
er":<String_value>,

"contact":<String_value>,
```

```
"serial":<Double_value>,

"refresh":<Double_value>,

"retry":<Double_value>,

"expire":<Double_value>,

"minimum":<Double_value>,

"ttl":<Double_value>,

"authtype":<String_value>,

"ecssubnet":<String_value>

}]}
```

count

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/dnssoarec?**count=yes**

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "dnssoarec": [ { "__count": "#no"} ] }
```

dnssrvrec

Configuration for server record resource.

Properties

(click to see Operations)

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
domain	<String>	Read-write	Domain name, which, by convention, is prefixed by the symbolic name of the desired service and the symbolic name of the desired protocol, each with an underscore (_) prepended. For example, if an SRV-aware client wants to discover a SIP service that is provided over UDP, in the domain example.com, the client performs a lookup for _sip._udp.example.com. Minimum length = 1
target	<String>	Read-write	Target host for the specified service.
priority	<Double>	Read-write	Integer specifying the priority of the target host. The lower the number,

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
			the higher the priority. If multiple target hosts have the same priority, selection is based on the Weight parameter. Minimum value = 0 Maximum value = 65535
weight	<Double>	Read-write	Weight for the target host. Aids host selection when two or more hosts have the same priority. A larger number indicates greater weight. Minimum value = 0 Maximum value = 65535
port	<Double>	Read-write	Port on which the target host listens for client requests. Minimum value = 0 Maximum value = 65535
ttl	<Double>	Read-write	Time to Live (TTL), in seconds, for the record. TTL is the time for which the record must be cached by DNS proxies. The specified TTL is

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
			applied to all the resource records that are of the same record type and belong to the specified domain name. For example, if you add an address record, with a TTL of 36000, to the domain name example.com, the TTLs of all the address records of example.com are changed to 36000. If the TTL is not specified, the Citrix ADC uses either the DNS zone's minimum TTL or, if the SOA record is not available on the appliance, the default value of 3600. Default value: 3600 Minimum value = 0 Maximum value = 2147483647
ecssubnet	<String>	Read-write	Subnet for which the cached SRV record need to be removed.
type	<String>	Read-write	Type of records to display. Available settings function as follows:

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
			* ADNS - Display all authoritative address records. * PROXY - Display all proxy address records. * ALL - Display all address records. Possible values = ALL, ADNS, PROXY
nodeid	<Double>	Read-write	Unique number that identifies the cluster node. Minimum value = 0 Maximum value = 31
authtype	<String>	Read-only	Record type. Possible values = ALL, ADNS, PROXY
__count	<Double>	Read-only	count parameter

Operations

(click to see Properties)

ADD| DELETE| UPDATE| UNSET| GET (ALL)| COUNT

Some options that you can use for each operations:

- **Getting warnings in response:**NITRO allows you to get warnings in an operation by specifying the "warning" query parameter as "yes". For example, to get warnings while connecting to the Citrix ADC appliance, the URL is as follows:

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/login?warning=yes

If any, the warnings are displayed in the response payload with the HTTP code "209 X-NITRO-WARNING".

- **Authenticated access for individual NITRO operations:**NITRO allows you to logon to the Citrix ADC appliance to perform individual operations. You can use this option instead of creating a NITRO session (using the login object) and then using that session to perform all operations,

  To do this, you must specify the username and password in the request header of the NITRO request as follows:

  X-NITRO-USER:<username>

  X-NITRO-PASS:<password>

  Note: In such cases, make sure that the request header DOES not include the following:

  Cookie:NITRO_AUTH_TOKEN=<tokenvalue>

`*Note:`*

Mandatory parameters are marked in redand placeholder content is marked in <green>.

add

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/dnssrvrec

HTTP Method:POST

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Content-Type:application/json

Request Payload:

```
{"dnssrvrec":{

<b>"domain":<String_value>,

</b><b>"target":<String_value>,

</b><b>"priority":<Double_value>,
```

```
</b><b>"weight":<Double_value>,

</b><b>"port":<Double_value>,

</b>"ttl":<Double_value>

}}
```

Response:

HTTP Status Code on Success: 201 Created HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error

delete

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/dnssrvrec/domain_value<String>

HTTP Method:DELETE

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue>

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error

update

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/dnssrvrec

HTTP Method:PUT

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Content-Type:application/json

Request Payload:

```
{"dnssrvrec":{

<b>"domain":<String_value>,

</b><b>"target":<String_value>,

</b>"priority":<Double_value>,

"weight":<Double_value>,

"port":<Double_value>,

"ttl":<Double_value>

}}
```

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error

unset

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/dnssrvrec?**action=unset**

HTTP Method:POST

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Content-Type:application/json

Request Payload:

```
{"dnssrvrec":{

<b>"domain":<String_value>,
```

```
</b><b>"target":<String_value>,

</b>"ttl":true

}}
```

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error

get (all)

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/dnssrvrec

Query-parameters:

args

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/dnssrvrec?**args=domain:<String_value>,target:<String_value>,type:<String_value>,nodeid:<Double_value>**

Use this query-parameter to get dnssrvrec resources based on additional properties.

attrs

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/dnssrvrec?**attrs=property-name1,property-name2**

Use this query parameter to specify the resource details that you want to retrieve.

filter

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/dnssrvrec?**filter=property-name1:property-val1,property-name2:property-val2**

Use this query-parameter to get the filtered set of dnssrvrec resources configured on Citrix ADC. Filtering can be done on any of the properties of the resource.

view

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/dnssrvrec?**view=summary**

Note: By default, the retrieved results are displayed in detail view (?view=detail).

pagination

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/dnssrvrec?**pagesize=#no;pageno=#no**

Use this query-parameter to get the dnssrvrec resources in chunks.

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "dnssrvrec": [ {

domain:<String_value>,target:<String_value>,type:<String_value>,nodeid:<Do
uble_value>"priority":<Double_value>,

"weight":<Double_value>,

"port":<Double_value>,

"ttl":<Double_value>,

"authtype":<String_value>,

"ecssubnet":<String_value>

}]}
```

count

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/dnssrvrec?**count=yes**

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "dnssrvrec": [ { "__count": "#no"} ] }
```

dnssubnetcache

Configuration for subnet cache resource.

Properties

(click to see Operations)

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
ecssubnet	<String>	Read-write	ECS Subnet.
all	<Boolean>	Read-write	Flush all the ECS subnets from the DNS cache.
nodeid	<Double>	Read-write	Unique number that identifies the cluster node.

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
			Minimum value = 0 Maximum value = 31
hostname	<String>	Read-only	Domain name. Minimum length = 1
nextrecs	<String[]>	Read-only	An array of record types associated with the domain name. Possible values = A, NS, CNAME, SOA, MX, AAAA, SRV, RRSIG, NSEC, DNSKEY, PTR, TXT, NAPTR
__count	<Double>	Read-only	count parameter

Operations

(click to see Properties)

FLUSH| GET (ALL)| GET| COUNT

Some options that you can use for each operations:

- **Getting warnings in response:**NITRO allows you to get warnings in an operation by specifying the "warning" query parameter as "yes". For example, to get warnings while connecting to the Citrix ADC appliance, the URL is as follows:

  http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/login?warning=yes

  If any, the warnings are displayed in the response payload with the HTTP code "209 X-NITRO-WARNING".
- **Authenticated access for individual NITRO operations:**NITRO allows you to logon to the Citrix ADC appliance to perform individual operations. You can use this option instead of

creating a NITRO session (using the login object) and then using that session to perform all operations,

To do this, you must specify the username and password in the request header of the NITRO request as follows:

X-NITRO-USER:<username>

X-NITRO-PASS:<password>

Note: In such cases, make sure that the request header DOES not include the following:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue>

`*Note:` *

Mandatory parameters are marked in redand placeholder content is marked in <green>.

flush

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/dnssubnetcache?**action=flush**

HTTP Method:POST

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Content-Type:application/json

Request Payload:

```
{"dnssubnetcache":{

"ecssubnet":<String_value>,

"all":<Boolean_value>

}}
```

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error

get (all)

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/dnssubnetcache

Query-parameters:

args

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/dnssubnetcache?**args=ecssubnet:<String_value>,nodeid:<Double_value>**

Use this query-parameter to get dnssubnetcache resources based on additional properties.

attrs

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/dnssubnetcache?**attrs=property-name1,property-name2**

Use this query parameter to specify the resource details that you want to retrieve.

filter

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/dnssubnetcache?**filter=property-name1:property-val1,property-name2:property-val2**

Use this query-parameter to get the filtered set of dnssubnetcache resources configured on Citrix ADC. Filtering can be done on any of the properties of the resource.

view

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/dnssubnetcache?**view=summary**

Note: By default, the retrieved results are displayed in detail view (?view=detail).

pagination

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/dnssubnetcache?**pagesize=#no;pageno=#no**

Use this query-parameter to get the dnssubnetcache resources in chunks.

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "dnssubnetcache": [ {

ecssubnet:<String_value>,nodeid:<Double_value>"hostname":<String_value>,

"nextrecs":<String[]_value>

}]}
```

get

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/dnssubnetcache/ecssubnet_value<String>

Query-parameters:

attrs

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/dnssubnetcache/ecssubnet_value<String>?**attrs=property-name1,property-name2**

Use this query parameter to specify the resource details that you want to retrieve.

view

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/dnssubnetcache/ecssubnet_value<String>?**view=summary**

Note: By default, the retrieved results are displayed in detail view (?view=detail).

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "dnssubnetcache": [ {

ecssubnet:<String_value>,nodeid:<Double_value>"hostname":<String_value>,

"nextrecs":<String[]_value>

}]}
```

count

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/dnssubnetcache?**count=yes**

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "dnssubnetcache": [ { "__count": "#no"} ] }
```

dnssuffix

Configuration for DNS suffix resource.

Properties

(click to see Operations)

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
Dnssuffix	<String>	Read-write	Suffix to be appended when resolving domain names that are not fully qualified. Minimum length = 1
__count	<Double>	Read-only	count parameter

Operations

(click to see Properties)

ADD| DELETE| GET (ALL)| GET| COUNT

Some options that you can use for each operations:

- **Getting warnings in response:**NITRO allows you to get warnings in an operation by specifying the "warning" query parameter as "yes". For example, to get warnings while connecting to the Citrix ADC appliance, the URL is as follows:

  http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/login?warning=yes

If any, the warnings are displayed in the response payload with the HTTP code "209 X-NITRO-WARNING".

- **Authenticated access for individual NITRO operations:**NITRO allows you to logon to the Citrix ADC appliance to perform individual operations. You can use this option instead of creating a NITRO session (using the login object) and then using that session to perform all operations,

  To do this, you must specify the username and password in the request header of the NITRO request as follows:

  X-NITRO-USER:<username>

  X-NITRO-PASS:<password>

  Note: In such cases, make sure that the request header DOES not include the following:

  Cookie:NITRO_AUTH_TOKEN=<tokenvalue>

`*Note:`*

Mandatory parameters are marked in redand placeholder content is marked in <green>.

add

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/dnssuffix

HTTP Method:POST

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Content-Type:application/json

Request Payload:

```
{"dnssuffix":{

<b>"Dnssuffix":<String_value>

</b>}}
```

Response:

HTTP Status Code on Success: 201 Created HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error

delete

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/dnssuffix/Dnssuffix_value<String>

HTTP Method:DELETE

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue>

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error

get (all)

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/dnssuffix

Query-parameters:

attrs

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/dnssuffix?**attrs=property-name1,property-name2**

Use this query parameter to specify the resource details that you want to retrieve.

filter

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/dnssuffix?**filter=property-name1:property-val1,property-name2:property-val2**

Use this query-parameter to get the filtered set of dnssuffix resources configured on Citrix ADC. Filtering can be done on any of the properties of the resource.

view

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/dnssuffix?**view=summary**

Note: By default, the retrieved results are displayed in detail view (?view=detail).

pagination

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/dnssuffix?**pagesize=#no;pageno=#no**

Use this query-parameter to get the dnssuffix resources in chunks.

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "dnssuffix": [ {

"Dnssuffix":<String_value>,



}]}
```

get

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/dnssuffix/Dnssuffix_value<String>

Query-parameters:

attrs

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/dnssuffix/Dnssuffix_value<String>?**attrs=property-name1,property-name2**

Use this query parameter to specify the resource details that you want to retrieve.

view

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/dnssuffix/Dnssuffix_value<String>?**view=summary**

Note: By default, the retrieved results are displayed in detail view (?view=detail).

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "dnssuffix": [ {

"Dnssuffix":<String_value>,



}]}
```

count

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/dnssuffix?**count=yes**

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "dnssuffix": [ { "__count": "#no"} ] }
```

dnstxtrec

Configuration for TXT record resource.

Properties

(click to see Operations)

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
domain	<String>	Read-write	Name of the domain for the TXT record. Minimum length = 1
String	<String[]>	Read-write	Information to store in the TXT resource record. Enclose the string in single or double quotation marks. A TXT resource record can contain up to six strings, each of which can contain up to 255 characters. If you want to add a string of more than 255 characters, evaluate whether splitting it into two or

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
			more smaller strings, subject to the six-string limit, works for you. Maximum length = 255
ttl	<Double>	Read-write	Time to Live (TTL), in seconds, for the record. TTL is the time for which the record must be cached by DNS proxies. The specified TTL is applied to all the resource records that are of the same record type and belong to the specified domain name. For example, if you add an address record, with a TTL of 36000, to the domain name example.com, the TTLs of all the address records of example.com are changed to 36000. If the TTL is not specified, the Citrix ADC uses either the DNS zone's minimum TTL or, if the SOA record is not available on the appliance, the

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
			default value of 3600. Default value: 3600 Minimum value = 0 Maximum value = 2147483647
recordid	<Double>	Read-write	Unique, internally generated record ID. View the details of the TXT record to obtain its record ID. Mutually exclusive with the string parameter. Minimum value = 1 Maximum value = 65535
ecssubnet	<String>	Read-write	Subnet for which the cached TXT record need to be removed.
type	<String>	Read-write	Type of records to display. Available settings function as follows: * ADNS - Display all authoritative address records. * PROXY - Display all proxy address records. * ALL - Display all address records.

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
			Default value: ADNS Possible values = ALL, ADNS, PROXY
nodeid	<Double>	Read-write	Unique number that identifies the cluster node. Minimum value = 0 Maximum value = 31
authtype	<String>	Read-only	Authentication type. Possible values = ALL, ADNS, PROXY
__count	<Double>	Read-only	count parameter

Operations

(click to see Properties)

ADD| DELETE| GET (ALL)| GET| COUNT

Some options that you can use for each operations:

- **Getting warnings in response:**NITRO allows you to get warnings in an operation by specifying the "warning" query parameter as "yes". For example, to get warnings while connecting to the Citrix ADC appliance, the URL is as follows:

  http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/login?warning=yes

  If any, the warnings are displayed in the response payload with the HTTP code "209 X-NITRO-WARNING".

- **Authenticated access for individual NITRO operations:**NITRO allows you to logon to the Citrix ADC appliance to perform individual operations. You can use this option instead of

creating a NITRO session (using the login object) and then using that session to perform all operations,

To do this, you must specify the username and password in the request header of the NITRO request as follows:

X-NITRO-USER:<username>

X-NITRO-PASS:<password>

Note: In such cases, make sure that the request header DOES not include the following:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue>

`*Note:`*

Mandatory parameters are marked in redand placeholder content is marked in <green>.

add

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/dnstxtrec

HTTP Method:POST

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Content-Type:application/json

Request Payload:

```
{"dnstxtrec":{

<b>"domain":<String_value>,

</b><b>"String":<String[]_value>,

</b>"ttl":<Double_value>

}}
```

Response:

HTTP Status Code on Success: 201 Created HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error

delete

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/dnstxtrec/domain_value<String>

HTTP Method:DELETE

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue>

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error

get (all)

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/dnstxtrec

Query-parameters:

args

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/dnstxtrec?**args=domain:<String_value>,type:<String_value>,nodeid:<Double_value>**

Use this query-parameter to get dnstxtrec resources based on additional properties.

attrs

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/dnstxtrec?**attrs=property-name1,property-name2**

Use this query parameter to specify the resource details that you want to retrieve.

filter

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/dnstxtrec?**filter=property-name1:property-val1,property-name2:property-val2**

Use this query-parameter to get the filtered set of dnstxtrec resources configured on Citrix ADC. Filtering can be done on any of the properties of the resource.

view

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/dnstxtrec?**view=summary**

Note: By default, the retrieved results are displayed in detail view (?view=detail).

pagination

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/dnstxtrec?**pagesize=#no;pageno=#no**

Use this query-parameter to get the dnstxtrec resources in chunks.

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "dnstxtrec": [ {

domain:<String_value>,type:<String_value>,nodeid:<Double_value>"String":<S
tring[]_value>,

"ecssubnet":<String_value>,

"ttl":<Double_value>,

"recordid":<Double_value>,

"authtype":<String_value>
```

```
}]}
```

get

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/dnstxtrec/domain_value<String>

Query-parameters:

attrs

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/dnstxtrec/domain_value<String>?**attrs=property-name1,property-name2**

Use this query parameter to specify the resource details that you want to retrieve.

view

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/dnstxtrec/domain_value<String>?**view=summary**

Note: By default, the retrieved results are displayed in detail view (?view=detail).

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "dnstxtrec": [ {

domain:<String_value>,type:<String_value>,nodeid:<Double_value>"String":<S
```

```
tring[]_value>,

"ecssubnet":<String_value>,

"ttl":<Double_value>,

"recordid":<Double_value>,

"authtype":<String_value>

}]}
```

count

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/dnstxtrec?**count=yes**

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "dnstxtrec": [ { "__count": "#no"} ] }
```

dnsview

Configuration for DNS view resource.

Properties

(click to see Operations)

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
viewname	<String>	Read-write	Name for the DNS view. Minimum length = 1
flags	<Double>	Read-only	Flags controlling display.
__count	<Double>	Read-only	count parameter

Operations

(click to see Properties)

ADD| DELETE| GET (ALL)| GET| COUNT

Some options that you can use for each operations:

- **Getting warnings in response:**NITRO allows you to get warnings in an operation by specifying the "warning" query parameter as "yes". For example, to get warnings while connecting to the Citrix ADC appliance, the URL is as follows:

  http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/login?warning=yes

  If any, the warnings are displayed in the response payload with the HTTP code "209 X-NITRO-WARNING".

- **Authenticated access for individual NITRO operations:**NITRO allows you to logon to the Citrix ADC appliance to perform individual operations. You can use this option instead of creating a NITRO session (using the login object) and then using that session to perform all operations,

  To do this, you must specify the username and password in the request header of the NITRO request as follows:

  X-NITRO-USER:<username>

  X-NITRO-PASS:<password>

  Note: In such cases, make sure that the request header DOES not include the following:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue>

`*Note:` *

Mandatory parameters are marked in redand placeholder content is marked in <green>.

add

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/dnsview

HTTP Method:POST

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Content-Type:application/json

Request Payload:

```
{"dnsview":{

<b>"viewname":<String_value>

</b>}}
```

Response:

HTTP Status Code on Success: 201 Created HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error

delete

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/dnsview/viewname_value<String>

HTTP Method:DELETE

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue>

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error

get (all)

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/dnsview

Query-parameters:

attrs

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/dnsview?**attrs=property-name1,property-name2**

Use this query parameter to specify the resource details that you want to retrieve.

filter

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/dnsview?**filter=property-name1:property-val1,property-name2:property-val2**

Use this query-parameter to get the filtered set of dnsview resources configured on Citrix ADC. Filtering can be done on any of the properties of the resource.

view

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/dnsview?**view=summary**

Note: By default, the retrieved results are displayed in detail view (?view=detail).

pagination

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/dnsview?**pagesize=#no;pageno=#no**

Use this query-parameter to get the dnsview resources in chunks.

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "dnsview": [ {

"viewname":<String_value>,

"servicename":<String_value>,

"flags":<Double_value>

}]}
```

get

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/dnsview/viewname_value<String>

Query-parameters:

attrs

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/dnsview/viewname_value<String>?**attrs=property-name1,property-name2**

Use this query parameter to specify the resource details that you want to retrieve.

view

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/dnsview/viewname_value<String>?**view=summary**

Note: By default, the retrieved results are displayed in detail view (?view=detail).

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "dnsview": [ {

"viewname":<String_value>,

"servicename":<String_value>,

"flags":<Double_value>

}]}
```

count

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/dnsview?**count=yes**

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "dnsview": [ { "__count": "#no"} ] }
```

dnsview_binding

Binding object which returns the resources bound to dnsview.

Properties

(click to see Operations)

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
viewname	<String>	Read-write	Name of the view to display. Minimum length = 1
dnsview_gslbservice_binding	<dnsview_gslbservice_binding[]>	Read-only	gslbservice that can be bound to dnsview.
dnsview_dnspolicy_binding	<dnsview_dnspolicy_binding[]>	Read-only	dnspolicy that can be bound to dnsview.

Operations

(click to see Properties)

GET| GET (ALL)

Some options that you can use for each operations:

- **Getting warnings in response:**NITRO allows you to get warnings in an operation by specifying the "warning" query parameter as "yes". For example, to get warnings while connecting to the Citrix ADC appliance, the URL is as follows:

  http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/login?warning=yes

  If any, the warnings are displayed in the response payload with the HTTP code "209 X-NITRO-WARNING".

- **Authenticated access for individual NITRO operations:**NITRO allows you to logon to the Citrix ADC appliance to perform individual operations. You can use this option instead of creating a NITRO session (using the login object) and then using that session to perform all operations,

  To do this, you must specify the username and password in the request header of the NITRO request as follows:

  X-NITRO-USER:<username>

  X-NITRO-PASS:<password>

  Note: In such cases, make sure that the request header DOES not include the following:

  Cookie:NITRO_AUTH_TOKEN=<tokenvalue>

`*Note:`*

Mandatory parameters are marked in redand placeholder content is marked in <green>.

get

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/dnsview_binding/viewname_value<String>

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "dnsview_binding": [ {
```

```
"viewname":<String_value>,

"dnsview_gslbservice_binding":<dnsview_gslbservice_binding[]_value>,

"dnsview_dnspolicy_binding":<dnsview_dnspolicy_binding[]_value>

}]}
```

get (all)

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/dnsview_binding

Query-parameters:

bulkbindings

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/dnsview_binding?**bulkbindings=yes**

NITRO allows you to fetch bindings in bulk.

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "dnsview_binding": [ {

"viewname":<String_value>,

"dnsview_gslbservice_binding":<dnsview_gslbservice_binding[]_value>,
```

```
"dnsview_dnspolicy_binding":<dnsview_dnspolicy_binding[]_value>

}]}
```

dnsview_dnspolicy_binding

Binding object showing the dnspolicy that can be bound to dnsview.

Properties

(click to see Operations)

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
viewname	<String>	Read-write	Name of the view to display. Minimum length = 1
dnspolicyname	<String>	Read-write	dnspolicy name of this view.
__count	<Double>	Read-write	count parameter

Operations

(click to see Properties)

GET| GET (ALL)| COUNT

Some options that you can use for each operations:

- **Getting warnings in response:**NITRO allows you to get warnings in an operation by specifying the "warning" query parameter as "yes". For example, to get warnings while connecting to the Citrix ADC appliance, the URL is as follows:

  http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/login?warning=yes

  If any, the warnings are displayed in the response payload with the HTTP code "209 X-NITRO-WARNING".

- **Authenticated access for individual NITRO operations:**NITRO allows you to logon to the Citrix ADC appliance to perform individual operations. You can use this option instead of

creating a NITRO session (using the login object) and then using that session to perform all operations,

To do this, you must specify the username and password in the request header of the NITRO request as follows:

X-NITRO-USER:<username>

X-NITRO-PASS:<password>

Note: In such cases, make sure that the request header DOES not include the following:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue>

`*Note:`*

Mandatory parameters are marked in redand placeholder content is marked in <green>.

get

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/dnsview_dnspolicy_binding/viewname_value<String>

Query-parameters:

filter

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/dnsview_dnspolicy_binding/viewname_value<String>?**filter=property-name1:property-value1,property-name2:property-value2**

Use this query-parameter to get the filtered set of dnsview_dnspolicy_binding resources configured on Citrix ADC. Filtering can be done on any of the properties of the resource.

pagination

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/dnsview_dnspolicy_binding/viewname_value<String>?**pagesize=#no;pageno=#no**

Use this query-parameter to get the dnsview_dnspolicy_binding resources in chunks.

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "dnsview_dnspolicy_binding": [ {

"viewname":<String_value>,

"dnspolicyname":<String_value>

}]}
```

get (all)

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/dnsview_dnspolicy_binding

Query-parameters:

bulkbindings

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/dnsview_dnspolicy_binding?**bulkbindings=yes**

NITRO allows you to fetch bindings in bulk.

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "dnsview_dnspolicy_binding": [ {

"viewname":<String_value>,

"dnspolicyname":<String_value>

}]}
```

count

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/dnsview_dnspolicy_binding/viewname_value<String>?**count=yes**

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{"dnsview_dnspolicy_binding": [ { "__count": "#no"} ] }
```

dnsview_gslbservice_binding

Binding object showing the gslbservice that can be bound to dnsview.

Properties

(click to see Operations)

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
viewname	<String>	Read-write	Name of the view to display. Minimum length = 1
gslbservicename	<String>	Read-write	Service name of the service using this view.
ipaddress	<String>	Read-only	IP of the service corresponding to the given view.
__count	<Double>	Read-only	count parameter

Operations

(click to see Properties)

GET| GET (ALL)| COUNT

Some options that you can use for each operations:

- **Getting warnings in response:**NITRO allows you to get warnings in an operation by specifying the "warning" query parameter as "yes". For example, to get warnings while connecting to the Citrix ADC appliance, the URL is as follows:

  http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/login?warning=yes

  If any, the warnings are displayed in the response payload with the HTTP code "209 X-NITRO-WARNING".

- **Authenticated access for individual NITRO operations:**NITRO allows you to logon to the Citrix ADC appliance to perform individual operations. You can use this option instead of creating a NITRO session (using the login object) and then using that session to perform all operations,

  To do this, you must specify the username and password in the request header of the NITRO request as follows:

  X-NITRO-USER:<username>

  X-NITRO-PASS:<password>

  Note: In such cases, make sure that the request header DOES not include the following:

  Cookie:NITRO_AUTH_TOKEN=<tokenvalue>

`*Note:`*

Mandatory parameters are marked in redand placeholder content is marked in <green>.

get

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/dnsview_gslbservice_binding/viewname_value<String>

Query-parameters:

filter

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/dnsview_gslbservice_binding/viewname_value<String>?**filter=property-name1:property-value1,property-name2:property-value2**

Use this query-parameter to get the filtered set of dnsview_gslbservice_binding resources configured on Citrix ADC. Filtering can be done on any of the properties of the resource.

pagination

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/dnsview_gslbservice_binding/viewname_value<String>?**pagesize=#no;pageno=#no**

Use this query-parameter to get the dnsview_gslbservice_binding resources in chunks.

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "dnsview_gslbservice_binding": [ {

"viewname":<String_value>,

"gslbservicename":<String_value>,

"ipaddress":<String_value>

}]}
```

get (all)

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/dnsview_gslbservice_binding

Query-parameters:

bulkbindings

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/dnsview_gslbservice_binding?**bulkbindings=yes**

NITRO allows you to fetch bindings in bulk.

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "dnsview_gslbservice_binding": [ {

"viewname":<String_value>,

"gslbservicename":<String_value>,

"ipaddress":<String_value>

}]}
```

count

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/dnsview_gslbservice_binding/viewname_value<String>?**count=yes**

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{"dnsview_gslbservice_binding": [ { "__count": "#no"} ] }
```

dnszone

Configuration for DNS zone resource.

Properties

(click to see Operations)

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
zonename	<String>	Read-write	Name of the zone to create. Minimum length = 1
proxymode	<String>	Read-write	Deploy the zone in proxy mode. Enable in the following scenarios: * The load balanced DNS servers are authoritative for the zone and all resource records that are part of the zone. * The load balanced DNS servers are authoritative for the zone, but the Citrix ADC owns a subset of the resource records that belong to the zone (partial zone ownership

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
			configuration). Typically seen in global server load balancing (GSLB) configurations, in which the appliance responds authoritatively to queries for GSLB domain names but forwards queries for other domain names in the zone to the load balanced servers. In either scenario, do not create the zone's Start of Authority (SOA) and name server (NS) resource records on the appliance. Disable if the appliance is authoritative for the zone, but make sure that you have created the SOA and NS records on the appliance before you create the zone.

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
			Default value: ENABLED Possible values = YES, NO
dnssecoffload	<String>	Read-write	Enable dnssec offload for this zone. Default value: DISABLED Possible values = ENABLED, DISABLED
nsec	<String>	Read-write	Enable nsec generation for dnssec offload. Default value: DISABLED Possible values = ENABLED, DISABLED
keyname	<String[]>	Read-write	Name of the public/ private DNS key pair with which to sign the zone. You can sign a zone with up to four keys. Minimum length = 1
type	<String>	Read-write	Type of zone to display. Mutually exclusive with the DNS Zone (zoneName) parameter. Available

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
			settings function as follows: * ADNS - Display all the zones for which the Citrix ADC is authoritative. * PROXY - Display all the zones for which the Citrix ADC is functioning as a proxy server. * ALL - Display all the zones configured on the appliance. Possible values = ALL, ADNS, PROXY
flags	<Double>	Read-only	Flags controlling display.
__count	<Double>	Read-only	count parameter

Operations

(click to see Properties)

ADD| UPDATE| UNSET| DELETE| SIGN| UNSIGN| GET (ALL)| GET| COUNT

Some options that you can use for each operations:

- **Getting warnings in response:**NITRO allows you to get warnings in an operation by specifying the "warning" query parameter as "yes". For example, to get warnings while connecting to the Citrix ADC appliance, the URL is as follows:

  http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/login?warning=yes

If any, the warnings are displayed in the response payload with the HTTP code "209 X-NITRO-WARNING".

- **Authenticated access for individual NITRO operations:**NITRO allows you to logon to the Citrix ADC appliance to perform individual operations. You can use this option instead of creating a NITRO session (using the login object) and then using that session to perform all operations,

  To do this, you must specify the username and password in the request header of the NITRO request as follows:

  X-NITRO-USER:<username>

  X-NITRO-PASS:<password>

  Note: In such cases, make sure that the request header DOES not include the following:

  Cookie:NITRO_AUTH_TOKEN=<tokenvalue>

`*Note:`*

Mandatory parameters are marked in redand placeholder content is marked in <green>.

add

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/dnszone

HTTP Method:POST

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Content-Type:application/json

Request Payload:

```
{"dnszone":{

<b>"zonename":<String_value>,

</b><b>"proxymode":<String_value>,

</b>"dnssecoffload":<String_value>,
```

```
"nsec":<String_value>

}}
```

Response:

HTTP Status Code on Success: 201 Created HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error

update

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/dnszone

HTTP Method:PUT

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Content-Type:application/json

Request Payload:

```
{"dnszone":{

<b>"zonename":<String_value>,

</b>"proxymode":<String_value>,

"dnssecoffload":<String_value>,

"nsec":<String_value>

}}
```

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error

unset

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/dnszone?**action=unset**

HTTP Method:POST

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Content-Type:application/json

Request Payload:

```
{"dnszone":{

<b>"zonename":<String_value>,

</b>"proxymode":true,

"dnssecoffload":true,

"nsec":true

}}
```

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error

delete

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/dnszone/zonename_value<String>

HTTP Method:DELETE

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue>

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error

sign

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/dnszone?**action=sign**

HTTP Method:POST

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Content-Type:application/json

Request Payload:

```
{"dnszone":{

<b>"zonename":<String_value>,

</b>"keyname":<String[]_value>

}}
```

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error

unsign

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/dnszone?**action=unsign**

HTTP Method:POST

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Content-Type:application/json

Request Payload:

```
{"dnszone":{

<b>"zonename":<String_value>,

</b>"keyname":<String[]_value>

}}
```

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error

get (all)

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/dnszone

Query-parameters:

args

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/dnszone?**args=zonename:<String_value>,type:<String_value>**

Use this query-parameter to get dnszone resources based on additional properties.

attrs

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/dnszone?**attrs=property-name1,property-name2**

Use this query parameter to specify the resource details that you want to retrieve.

filter

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/dnszone?**filter=property-name1:property-val1,property-name2:property-val2**

Use this query-parameter to get the filtered set of dnszone resources configured on Citrix ADC. Filtering can be done on any of the properties of the resource.

view

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/dnszone?**view=summary**

Note: By default, the retrieved results are displayed in detail view (?view=detail).

pagination

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/dnszone?**pagesize=#no;pageno=#no**

Use this query-parameter to get the dnszone resources in chunks.

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "dnszone": [ {

zonename:<String_value>,type:<String_value>"proxymode":<String_value>,

"flags":<Double_value>,

"nsecbitarray":<String[]_value>,

"dnssecoffload":<String_value>,

"nsec":<String_value>,

"keyname":<String[]_value>

}]}
```

get

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/dnszone/zonename_value<String>

Query-parameters:

attrs

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/dnszone/zonename_value<String>?**attrs=property-name1,property-name2**

Use this query parameter to specify the resource details that you want to retrieve.

view

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/dnszone/zonename_value<String>?**view=summary**

Note: By default, the retrieved results are displayed in detail view (?view=detail).

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "dnszone": [ {

zonename:<String_value>,type:<String_value>"proxymode":<String_value>,

"flags":<Double_value>,
```

```
"nsecbitarray":<String[]_value>,

"dnssecoffload":<String_value>,

"nsec":<String_value>,

"keyname":<String[]_value>

}]}
```

count

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/dnszone?**count=yes**

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "dnszone": [ { "__count": "#no"} ] }
```

dnszone_binding

Binding object which returns the resources bound to dnszone.

Properties

(click to see Operations)

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
zonename	<String>	Read-write	Name of the zone. Mutually exclusive with the type parameter. Minimum length = 1
dnszone_domain_binding	<dnszone_domain_binding[]>	Read-only	domain that can be bound to dnszone.
dnszone_dnskey_binding	<dnszone_dnskey_binding[]>	Read-only	dnskey that can be bound to dnszone.

Operations

(click to see Properties)

GET| GET (ALL)

Some options that you can use for each operations:

- **Getting warnings in response:**NITRO allows you to get warnings in an operation by specifying the "warning" query parameter as "yes". For example, to get warnings while connecting to the Citrix ADC appliance, the URL is as follows:

  http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/login?warning=yes

  If any, the warnings are displayed in the response payload with the HTTP code "209 X-NITRO-WARNING".

- **Authenticated access for individual NITRO operations:**NITRO allows you to logon to the Citrix ADC appliance to perform individual operations. You can use this option instead of creating a NITRO session (using the login object) and then using that session to perform all operations,

  To do this, you must specify the username and password in the request header of the NITRO request as follows:

  X-NITRO-USER:<username>

  X-NITRO-PASS:<password>

  Note: In such cases, make sure that the request header DOES not include the following:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue>

`*Note:` *

Mandatory parameters are marked in redand placeholder content is marked in <green>.

get

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/dnszone_binding/zonename_value<String>

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "dnszone_binding": [ {

"zonename":<String_value>,

"dnszone_dnskey_binding":<dnszone_dnskey_binding[]_value>,

"dnszone_domain_binding":<dnszone_domain_binding[]_value>

}]}
```

get (all)

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/dnszone_binding

Query-parameters:

bulkbindings

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/dnszone_binding?**bulkbindings=yes**

NITRO allows you to fetch bindings in bulk.

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "dnszone_binding": [ {

"zonename":<String_value>,

"dnszone_dnskey_binding":<dnszone_dnskey_binding[]_value>,

"dnszone_domain_binding":<dnszone_domain_binding[]_value>

}]}
```

dnszone_dnskey_binding

Binding object showing the dnskey that can be bound to dnszone.

Properties

(click to see Operations)

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
zonename	<String>	Read-write	Name of the zone. Mutually exclusive with the type parameter. Minimum length = 1
keyname	<String[]>	Read-write	Name of the public/ private DNS key pair with which to sign the zone. You can sign a zone with up to four keys. Minimum length = 1
siginceptiontime	<Double[]>	Read-only	The time when sign was done with this key. Minimum value = 1
signed	<Double>	Read-only	Integer which denote status of keys.
expires	<Double>	Read-only	Time period for which to consider the key valid, after the key is used to sign a zone.
__count	<Double>	Read-only	count parameter

Operations

(click to see Properties)

GET| GET (ALL)| COUNT

Some options that you can use for each operations:

- **Getting warnings in response:**NITRO allows you to get warnings in an operation by specifying the "warning" query parameter as "yes". For example, to get warnings while connecting to the Citrix ADC appliance, the URL is as follows:

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/login?warning=yes

If any, the warnings are displayed in the response payload with the HTTP code "209 X-NITRO-WARNING".

- **Authenticated access for individual NITRO operations:**NITRO allows you to logon to the Citrix ADC appliance to perform individual operations. You can use this option instead of creating a NITRO session (using the login object) and then using that session to perform all operations,

  To do this, you must specify the username and password in the request header of the NITRO request as follows:

  X-NITRO-USER:<username>

  X-NITRO-PASS:<password>

  Note: In such cases, make sure that the request header DOES not include the following:

  Cookie:NITRO_AUTH_TOKEN=<tokenvalue>

`*Note:`*

Mandatory parameters are marked in redand placeholder content is marked in <green>.

get

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/dnszone_dnskey_binding/zonename_value<String>

Query-parameters:

filter

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/dnszone_dnskey_binding/zonename_value<String>?**filter=property-name1:property-value1,property-name2:property-value2**

Use this query-parameter to get the filtered set of dnszone_dnskey_binding resources configured on Citrix ADC. Filtering can be done on any of the properties of the resource.

pagination

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/dnszone_dnskey_binding/zonename_value<String>?**pagesize=#no;pageno=#no**

Use this query-parameter to get the dnszone_dnskey_binding resources in chunks.

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "dnszone_dnskey_binding": [ {

"zonename":<String_value>,

"keyname":<String[]_value>,

"siginceptiontime":<Double[]_value>,

"signed":<Double_value>,

"expires":<Double_value>

}]}
```

get (all)

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/dnszone_dnskey_binding

Query-parameters:

bulkbindings

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/dnszone_dnskey_binding?**bulkbindings=yes**

NITRO allows you to fetch bindings in bulk.

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "dnszone_dnskey_binding": [ {

"zonename":<String_value>,

"keyname":<String[]_value>,

"siginceptiontime":<Double[]_value>,

"signed":<Double_value>,

"expires":<Double_value>

}]}
```

count

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/dnszone_dnskey_binding/zonename_value<String>?**count=yes**

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{"dnszone_dnskey_binding": [ { "__count": "#no"} ] }
```

dnszone_domain_binding

Binding object showing the domain that can be bound to dnszone.

Properties

(click to see Operations)

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
zonename	<String>	Read-write	Name of the zone. Mutually exclusive with the type parameter. Minimum length = 1
domain	<String>	Read-write	Domain name that belongs to the given zone.
nextrecs	<String[]>	Read-only	An array of record types associated with the nsec record. Possible values = A, NS, CNAME, SOA, MX, AAAA, SRV,

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
			RRSIG, NSEC, DNSKEY, PTR, TXT, NAPTR
__count	<Double>	Read-only	count parameter

Operations

(click to see Properties)

GET| GET (ALL)| COUNT

Some options that you can use for each operations:

- **Getting warnings in response:**NITRO allows you to get warnings in an operation by specifying the "warning" query parameter as "yes". For example, to get warnings while connecting to the Citrix ADC appliance, the URL is as follows:

  http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/login?warning=yes

  If any, the warnings are displayed in the response payload with the HTTP code "209 X-NITRO-WARNING".

- **Authenticated access for individual NITRO operations:**NITRO allows you to logon to the Citrix ADC appliance to perform individual operations. You can use this option instead of creating a NITRO session (using the login object) and then using that session to perform all operations,

  To do this, you must specify the username and password in the request header of the NITRO request as follows:

  X-NITRO-USER:<username>

  X-NITRO-PASS:<password>

  Note: In such cases, make sure that the request header DOES not include the following:

  Cookie:NITRO_AUTH_TOKEN=<tokenvalue>

`*Note:`*

Mandatory parameters are marked in redand placeholder content is marked in <green>.

get

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/dnszone_domain_binding/zonename_value<String>

Query-parameters:

filter

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/dnszone_domain_binding/zonename_value<String>?**filter=property-name1:property-value1,property-name2:property-value2**

Use this query-parameter to get the filtered set of dnszone_domain_binding resources configured on Citrix ADC. Filtering can be done on any of the properties of the resource.

pagination

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/dnszone_domain_binding/zonename_value<String>?**pagesize=#no;pageno=#no**

Use this query-parameter to get the dnszone_domain_binding resources in chunks.

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "dnszone_domain_binding": [ {

"zonename":<String_value>,
```

```
"domain":<String_value>,

"nextrecs":<String[]_value>

}]}
```

get (all)

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/dnszone_domain_binding

Query-parameters:

bulkbindings

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/dnszone_domain_binding?**bulkbindings=yes**

NITRO allows you to fetch bindings in bulk.

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "dnszone_domain_binding": [ {

"zonename":<String_value>,

"domain":<String_value>,

"nextrecs":<String[]_value>
```

```
}]}
```

count

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/dnszone_domain_binding/zonename_value<String>?**count=yes**

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{"dnszone_domain_binding": [ { "__count": "#no"} ] }
```

Filter

Filter

Filter configuration. Protection from malicious attacks can be achieved for web sites at the content level with the system's content filtering feature. When this feature is enabled, the system inspects every incoming request with the user-configured rules based on any HTTP URL or headers.

NAME	DESCRIPTION
filteraction	Configuration for filter action

NAME	DESCRIPTION
filterglobal_binding	Binding object showing the resources that can be bound to filterglobal
filterglobal_filterpolicy_binding	Binding object showing the filterpolicy that can be bound to filterglobal
filterhtmlinjectionparameter	Configuration for HTML injection parameter
filterhtmlinjectionvariable	Configuration for HTML injection variable
filterpolicy	Configuration for filter policy
filterpolicy_binding	Binding object showing the resources that can be bound to filterpolicy
filterpolicy_crvserver_binding	Binding object showing the crvserver that can be bound to filterpolicy
filterpolicy_csvserver_binding	Binding object showing the csvserver that can be bound to filterpolicy
filterpolicy_filterglobal_binding	Binding object showing the filterglobal that can be bound to filterpolicy
filterpolicy_lbvserver_binding	Binding object showing the lbvserver that can be bound to filterpolicy
filterpostbodyinjection	Configuration for HTML Injection postbody
filterprebodyinjection	Configuration for HTML Injection prebody

filteraction

Configuration for filter action resource.

Properties

(click to see Operations)

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
name	<String>	Read-write	Name for the filtering action. Must begin with a letter, number, or the underscore character (_). Other characters allowed, after the first character, are the hyphen (-), period (.) hash (#), space ( ), at sign (@), equals (=), and colon (:) characters. Choose a name that helps identify the type of action. The name of a filter action cannot be changed after it is created. CLI Users: If the name includes one or more spaces, enclose the name in double or single quotation marks (for example, "my action" or 'my action'). Minimum length = 1
qual	<String>	Read-write	Qualifier, which is the action to be performed. The qualifier cannot be changed after it is set. The available options function as follows:

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
			ADD - Adds the specified HTTP header. RESET - Terminates the connection, sending the appropriate termination notice to the user's browser. FORWARD - Redirects the request to the designated service. You must specify either a service name or a page, but not both. DROP - Silently deletes the request, without sending a response to the user's browser. CORRUPT - Modifies the designated HTTP header to prevent it from performing the function it was intended to perform, then sends the request/response to the server/browser.

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
			ERRORCODE. Returns the designated HTTP error code to the user's browser (for example, 404, the standard HTTP code for a non-existent Web page). Possible values = reset, add, corrupt, forward, errorcode, drop
servicename	<String>	Read-write	Service to which to forward HTTP requests. Required if the qualifier is FORWARD. Minimum length = 1
value	<String>	Read-write	String containing the header_name and header_value. If the qualifier is ADD, specify <header_name>:<header_value>. If the qualifier is CORRUPT, specify only the header_name. Minimum length = 1
respcode	<Double>	Read-write	Response code to be returned for HTTP requests (for use with the

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
			ERRORCODE qualifier). Minimum value = 1
page	<String>	Read-write	HTML page to return for HTTP requests (For use with the ERRORCODE qualifier). Minimum length = 1
isdefault	<Boolean>	Read-only	A value of true is returned if it is a default filteraction.
__count	<Double>	Read-only	count parameter

Operations

(click to see Properties)

ADD| DELETE| UPDATE| UNSET| GET (ALL)| GET| COUNT

Some options that you can use for each operations:

- **Getting warnings in response:**NITRO allows you to get warnings in an operation by specifying the "warning" query parameter as "yes". For example, to get warnings while connecting to the Citrix ADC appliance, the URL is as follows:

  http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/login?warning=yes

  If any, the warnings are displayed in the response payload with the HTTP code "209 X-NITRO-WARNING".

- **Authenticated access for individual NITRO operations:**NITRO allows you to logon to the Citrix ADC appliance to perform individual operations. You can use this option instead of creating a NITRO session (using the login object) and then using that session to perform all operations,

  To do this, you must specify the username and password in the request header of the NITRO request as follows:

X-NITRO-USER:<username>

X-NITRO-PASS:<password>

Note: In such cases, make sure that the request header DOES not include the following:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue>

`*Note:`*

Mandatory parameters are marked in redand placeholder content is marked in <green>.

add

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/filteraction

HTTP Method:POST

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Content-Type:application/json

Request Payload:

```
{"filteraction":{

<b>"name":<String_value>,

</b><b>"qual":<String_value>,

</b>"servicename":<String_value>,

"value":<String_value>,

"respcode":<Double_value>,

"page":<String_value>

}}
```

Response:

HTTP Status Code on Success: 201 Created HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error

delete

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/filteraction/name_value<String>

HTTP Method:DELETE

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue>

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error

update

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/filteraction

HTTP Method:PUT

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Content-Type:application/json

Request Payload:

```
{"filteraction":{

<b>"name":<String_value>,

</b>"servicename":<String_value>,

"value":<String_value>,

"respcode":<Double_value>,

"page":<String_value>
```

```
}}
```

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error

unset

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/filteraction?**action=unset**

HTTP Method:POST

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Content-Type:application/json

Request Payload:

```
{"filteraction":{

<b>"name":<String_value>,

</b>"page":true

}}
```

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error

get (all)

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/filteraction

Query-parameters:

attrs

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/filteraction?**attrs=property-name1,property-name2**

Use this query parameter to specify the resource details that you want to retrieve.

filter

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/filteraction?**filter=property-name1:property-val1,property-name2:property-val2**

Use this query-parameter to get the filtered set of filteraction resources configured on Citrix ADC. Filtering can be done on any of the properties of the resource.

view

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/filteraction?**view=summary**

Note: By default, the retrieved results are displayed in detail view (?view=detail).

pagination

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/filteraction?**pagesize=#no;pageno=#no**

Use this query-parameter to get the filteraction resources in chunks.

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "filteraction": [ {

"name":<String_value>,

"qual":<String_value>,

"servicename":<String_value>,

"value":<String_value>,

"respcode":<Double_value>,

"page":<String_value>,

"isdefault":<Boolean_value>

}]}
```

get

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/filteraction/name_value<String>

Query-parameters:

attrs

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/filteraction/name_value<String>?**attrs=property-name1,property-name2**

Use this query parameter to specify the resource details that you want to retrieve.

view

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/filteraction/name_value<String>?**view=summary**

Note: By default, the retrieved results are displayed in detail view (?view=detail).

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "filteraction": [ {

"name":<String_value>,

"qual":<String_value>,

"servicename":<String_value>,

"value":<String_value>,

"respcode":<Double_value>,

"page":<String_value>,

"isdefault":<Boolean_value>

}]}
```

count

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/filteraction?**count=yes**

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "filteraction": [ { "__count": "#no"} ] }
```

filterglobal_binding

Binding object which returns the resources bound to filterglobal.

Properties

(click to see Operations)

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
filterglobal_filterpolicy_binding	<filterglobal_filterpolicy_binding[]>	Read-only	filterpolicy that can be bound to filterglobal.

Operations

(click to see Properties)

GET| GET (ALL)

Some options that you can use for each operations:

- **Getting warnings in response:**NITRO allows you to get warnings in an operation by specifying the "warning" query parameter as "yes". For example, to get warnings while connecting to the Citrix ADC appliance, the URL is as follows:

  http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/login?warning=yes

  If any, the warnings are displayed in the response payload with the HTTP code "209 X-NITRO-WARNING".

- **Authenticated access for individual NITRO operations:**NITRO allows you to logon to the Citrix ADC appliance to perform individual operations. You can use this option instead of creating a NITRO session (using the login object) and then using that session to perform all operations,

  To do this, you must specify the username and password in the request header of the NITRO request as follows:

  X-NITRO-USER:<username>

  X-NITRO-PASS:<password>

  Note: In such cases, make sure that the request header DOES not include the following:

  Cookie:NITRO_AUTH_TOKEN=<tokenvalue>

`*Note:`*

Mandatory parameters are marked in redand placeholder content is marked in <green>.

get

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/filterglobal_binding

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "filterglobal_binding": [ {

"filterglobal_filterpolicy_binding":<filterglobal_filterpolicy_binding[]_v
```

```
alue>

}]}
```

get (all)

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/filterglobal_binding

Query-parameters:

bulkbindings

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/filterglobal_binding?**bulkbindings=yes**

NITRO allows you to fetch bindings in bulk.

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "filterglobal_binding": [ {

"filterglobal_filterpolicy_binding":<filterglobal_filterpolicy_binding[]_v
alue>

}]}
```

filterglobal_filterpolicy_binding

Binding object showing the filterpolicy that can be bound to filterglobal.

Properties

(click to see Operations)

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
priority	<Double>	Read-write	The priority of the policy.
state	<String>	Read-write	State of the binding. Possible values = ENABLED, DISABLED
policyname	<String>	Read-write	The name of the filter policy.
__count	<Double>	Read-write	count parameter

Operations

(click to see Properties)

ADD:| DELETE:| GET| GET (ALL)| COUNT

Some options that you can use for each operations:

- **Getting warnings in response:**NITRO allows you to get warnings in an operation by specifying the "warning" query parameter as "yes". For example, to get warnings while connecting to the Citrix ADC appliance, the URL is as follows:

  http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/login?warning=yes

  If any, the warnings are displayed in the response payload with the HTTP code "209 X-NITRO-WARNING".

- **Authenticated access for individual NITRO operations:**NITRO allows you to logon to the Citrix ADC appliance to perform individual operations. You can use this option instead of creating a NITRO session (using the login object) and then using that session to perform all operations,

To do this, you must specify the username and password in the request header of the NITRO request as follows:

X-NITRO-USER:<username>

X-NITRO-PASS:<password>

Note: In such cases, make sure that the request header DOES not include the following:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue>

`*Note:`*

Mandatory parameters are marked in redand placeholder content is marked in <green>.

add

URL:http://<Citrix-ADC-IP-address(NSIP)/nitro/v1/config/filterglobal_filterpolicy_binding

HTTP Method:PUT

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Content-Type:application/json

Request Payload:

```
{

"filterglobal_filterpolicy_binding":{

<b>"policyname":<String_value>,

</b>"priority":<Double_value>,

"state":<String_value>

}}
```

Response:

HTTP Status Code on Success: 201 Created HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error

delete

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/filterglobal_filterpolicy_binding

Query-parameters:

args

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/filterglobal_filterpolicy_binding?**args=policyname:<String_value>**

HTTP Method:DELETE

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue>

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error

get

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/filterglobal_filterpolicy_binding

Query-parameters:

filter

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/filterglobal_filterpolicy_binding?**filter=property-name1:property-value1,property-name2:property-value2**

Use this query-parameter to get the filtered set of filterglobal_filterpolicy_binding resources configured on Citrix ADC. Filtering can be done on any of the properties of the resource.

pagination

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/filterglobal_filterpolicy_binding?**pagesize=#no;pageno=#no**

Use this query-parameter to get the filterglobal_filterpolicy_binding resources in chunks.

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "filterglobal_filterpolicy_binding": [ {

"priority":<Double_value>,

"state":<String_value>,

"policyname":<String_value>

}]}
```

get (all)

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/filterglobal_filterpolicy_binding

Query-parameters:

bulkbindings

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/filterglobal_filterpolicy_binding?**bulkbindings=yes**

NITRO allows you to fetch bindings in bulk.

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "filterglobal_filterpolicy_binding": [ {

"priority":<Double_value>,

"state":<String_value>,

"policyname":<String_value>

}]}
```

count

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/filterglobal_filterpolicy_binding?**count=yes**

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "filterglobal_filterpolicy_binding": [ { "__count": "#no"} ] }
```

filterhtmlinjectionparameter

Configuration for HTML injection parameter resource.

Properties

(click to see Operations)

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
rate	<Double>	Read-write	For a rate of x, HTML injection is done for 1 out of x policy matches. Default value: 1 Minimum value = 1
frequency	<Double>	Read-write	For a frequency of x, HTML injection is done at least once per x milliseconds. Default value: 1 Minimum value = 1
strict	<String>	Read-write	Searching for <html> tag. If this parameter is enabled, HTML injection does not insert the prebody or postbody content unless the <html> tag is found.

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
			Default value: ENABLED Possible values = ENABLED, DISABLED
htmlsearchlen	<Double>	Read-write	Number of characters, in the HTTP body, in which to search for the <html> tag if strict mode is set. Default value: 1024 Minimum value = 1

Operations

(click to see Properties)

UPDATE| UNSET| GET (ALL)

Some options that you can use for each operations:

- **Getting warnings in response:**NITRO allows you to get warnings in an operation by specifying the "warning" query parameter as "yes". For example, to get warnings while connecting to the Citrix ADC appliance, the URL is as follows:

  http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/login?warning=yes

  If any, the warnings are displayed in the response payload with the HTTP code "209 X-NITRO-WARNING".

- **Authenticated access for individual NITRO operations:**NITRO allows you to logon to the Citrix ADC appliance to perform individual operations. You can use this option instead of creating a NITRO session (using the login object) and then using that session to perform all operations,

  To do this, you must specify the username and password in the request header of the NITRO request as follows:

X-NITRO-USER:<username>

X-NITRO-PASS:<password>

Note: In such cases, make sure that the request header DOES not include the following:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue>

`*Note:`*

Mandatory parameters are marked in redand placeholder content is marked in <green>.

update

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/filterhtmlinjectionparameter

HTTP Method:PUT

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Content-Type:application/json

Request Payload:

```
{"filterhtmlinjectionparameter":{

"rate":<Double_value>,

"frequency":<Double_value>,

"strict":<String_value>,

"htmlsearchlen":<Double_value>

}}
```

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error

unset

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/filterhtmlinjectionparameter?**action=unset**

HTTP Method:POST

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Content-Type:application/json

Request Payload:

```
{"filterhtmlinjectionparameter":{

"rate":true,

"frequency":true,

"strict":true,

"htmlsearchlen":true

}}
```

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error

get (all)

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/filterhtmlinjectionparameter

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "filterhtmlinjectionparameter": [ {

"rate":<Double_value>,

"frequency":<Double_value>,

"strict":<String_value>,

"htmlsearchlen":<Double_value>

}]}
```

filterhtmlinjectionvariable

Configuration for HTML injection variable resource.

Properties

(click to see Operations)

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
variable	<String>	Read-write	Name for the HTML injection variable to be added. Minimum length = 1 Maximum length = 31
value	<String>	Read-write	Value to be assigned to the new variable.

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
			Minimum length = 1 Maximum length = 31
type	<String>	Read-only	Type of the HTML injection variable. Possible values = System internal, User defined
__count	<Double>	Read-only	count parameter

Operations

(click to see Properties)

ADD| DELETE| UPDATE| UNSET| GET (ALL)| GET| COUNT

Some options that you can use for each operations:

- **Getting warnings in response:**NITRO allows you to get warnings in an operation by specifying the "warning" query parameter as "yes". For example, to get warnings while connecting to the Citrix ADC appliance, the URL is as follows:

  http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/login?warning=yes

  If any, the warnings are displayed in the response payload with the HTTP code "209 X-NITRO-WARNING".

- **Authenticated access for individual NITRO operations:**NITRO allows you to logon to the Citrix ADC appliance to perform individual operations. You can use this option instead of creating a NITRO session (using the login object) and then using that session to perform all operations,

  To do this, you must specify the username and password in the request header of the NITRO request as follows:

  X-NITRO-USER:<username>

  X-NITRO-PASS:<password>

Note: In such cases, make sure that the request header DOES not include the following:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue>

`*Note:` *

Mandatory parameters are marked in redand placeholder content is marked in <green>.

add

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/filterhtmlinjectionvariable

HTTP Method:POST

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Content-Type:application/json

Request Payload:

```
{"filterhtmlinjectionvariable":{

<b>"variable":<String_value>,

</b>"value":<String_value>

}}
```

Response:

HTTP Status Code on Success: 201 Created HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error

delete

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/filterhtmlinjectionvariable/variable_value<String>

HTTP Method:DELETE

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue>

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error

update

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/filterhtmlinjectionvariable

HTTP Method:PUT

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Content-Type:application/json

Request Payload:

```
{"filterhtmlinjectionvariable":{

<b>"variable":<String_value>,

</b>"value":<String_value>

}}
```

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error

unset

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/filterhtmlinjectionvariable?**action=unset**

HTTP Method:POST

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Content-Type:application/json

Request Payload:

```
{"filterhtmlinjectionvariable":{

<b>"variable":<String_value>,

</b>"value":true

}}
```

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error

get (all)

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/filterhtmlinjectionvariable

Query-parameters:

attrs

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/filterhtmlinjectionvariable?**attrs=property-name1,property-name2**

Use this query parameter to specify the resource details that you want to retrieve.

filter

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/filterhtmlinjectionvariable?**filter=property-name1:property-val1,property-name2:property-val2**

Use this query-parameter to get the filtered set of filterhtmlinjectionvariable resources configured on Citrix ADC. Filtering can be done on any of the properties of the resource.

view

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/filterhtmlinjectionvariable?**view=summary**

Note: By default, the retrieved results are displayed in detail view (?view=detail).

pagination

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/filterhtmlinjectionvariable?**pagesize=#no;pageno=#no**

Use this query-parameter to get the filterhtmlinjectionvariable resources in chunks.

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "filterhtmlinjectionvariable": [ {

"variable":<String_value>,

"value":<String_value>,

"type":<String_value>

}]}
```

get

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/filterhtmlinjectionvariable/variable_value<String>

Query-parameters:

attrs

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/filterhtmlinjectionvariable/variable_value<String>?**attrs=property-name1,property-name2**

Use this query parameter to specify the resource details that you want to retrieve.

view

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/filterhtmlinjectionvariable/variable_value<String>?**view=summary**

Note: By default, the retrieved results are displayed in detail view (?view=detail).

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "filterhtmlinjectionvariable": [ {

"variable":<String_value>,

"value":<String_value>,

"type":<String_value>

}]}
```

count

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/filterhtmlinjectionvariable?**count=yes**

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "filterhtmlinjectionvariable": [ { "__count": "#no"} ] }
```

filterpolicy

Configuration for filter policy resource.

Properties

(click to see Operations)

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
name	<String>	Read-write	Name for the filtering action. Must begin with a letter, number, or the underscore character (_). Other characters allowed, after the first

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
			character, are the hyphen (-), period (.) pound (#), space ( ), at (@), equals (=), and colon (:) characters. Choose a name that helps identify the type of action. The name cannot be updated after the policy is created. CLI Users: If the name includes one or more spaces, enclose the name in double or single quotation marks (for example, "my policy" or 'my policy'). Minimum length = 1
rule	<String>	Read-write	Citrix ADC classic expression specifying the type of connections that match this policy. Minimum length = 1
reqaction	<String>	Read-write	Name of the action to be performed on requests that match the policy. Cannot be specified if the rule includes condition to be evaluated for responses.

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
			Minimum length = 1
resaction	<String>	Read-write	The action to be performed on the response. The string value can be a filter action created filter action or a built-in action. Minimum length = 1
hits	<Double>	Read-only	.
__count	<Double>	Read-only	count parameter

Operations

(click to see Properties)

ADD| DELETE| UPDATE| GET (ALL)| GET| COUNT

Some options that you can use for each operations:

- **Getting warnings in response:**NITRO allows you to get warnings in an operation by specifying the "warning" query parameter as "yes". For example, to get warnings while connecting to the Citrix ADC appliance, the URL is as follows:

  http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/login?warning=yes

  If any, the warnings are displayed in the response payload with the HTTP code "209 X-NITRO-WARNING".

- **Authenticated access for individual NITRO operations:**NITRO allows you to logon to the Citrix ADC appliance to perform individual operations. You can use this option instead of creating a NITRO session (using the login object) and then using that session to perform all operations,

  To do this, you must specify the username and password in the request header of the NITRO request as follows:

  X-NITRO-USER:<username>

X-NITRO-PASS:<password>

Note: In such cases, make sure that the request header DOES not include the following:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue>

`*Note:`*

Mandatory parameters are marked in redand placeholder content is marked in <green>.

add

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/filterpolicy

HTTP Method:POST

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Content-Type:application/json

Request Payload:

```
{"filterpolicy":{

<b>"name":<String_value>,

</b><b>"rule":<String_value>,

</b>"reqaction":<String_value>,

"resaction":<String_value>

}}
```

Response:

HTTP Status Code on Success: 201 Created HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error

delete

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/filterpolicy/name_value<String>

HTTP Method:DELETE

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue>

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error

update

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/filterpolicy

HTTP Method:PUT

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Content-Type:application/json

Request Payload:

```
{"filterpolicy":{

<b>"name":<String_value>,

</b>"rule":<String_value>,

"reqaction":<String_value>,

"resaction":<String_value>

}}
```

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error

get (all)

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/filterpolicy

Query-parameters:

attrs

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/filterpolicy?**attrs=property-name1,property-name2**

Use this query parameter to specify the resource details that you want to retrieve.

filter

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/filterpolicy?**filter=property-name1:property-val1,property-name2:property-val2**

Use this query-parameter to get the filtered set of filterpolicy resources configured on Citrix ADC. Filtering can be done on any of the properties of the resource.

view

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/filterpolicy?**view=summary**

Note: By default, the retrieved results are displayed in detail view (?view=detail).

pagination

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/filterpolicy?**pagesize=#no;pageno=#no**

Use this query-parameter to get the filterpolicy resources in chunks.

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "filterpolicy": [ {

"name":<String_value>,

"rule":<String_value>,

"reqaction":<String_value>,

"resaction":<String_value>,

"hits":<Double_value>

}]}
```

get

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/filterpolicy/name_value<String>

Query-parameters:

attrs

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/filterpolicy/name_value<String>?**attrs=property-name1,property-name2**

Use this query parameter to specify the resource details that you want to retrieve.

view

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/filterpolicy/name_value<String>?**view=summary**

Note: By default, the retrieved results are displayed in detail view (?view=detail).

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "filterpolicy": [ {

"name":<String_value>,

"rule":<String_value>,

"reqaction":<String_value>,

"resaction":<String_value>,

"hits":<Double_value>

}]}
```

count

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/filterpolicy?**count=yes**

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "filterpolicy": [ { "__count": "#no"} ] }
```

filterpolicy_binding

Binding object which returns the resources bound to filterpolicy.

Properties

(click to see Operations)

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
name	<String>	Read-write	Name of the filter policy to be displayed. If a name is not provided, information about all the filter policies is shown. Minimum length = 1
filterpolicy_csvserver_binding	<filterpolicy_csvserver_binding[]>	Read-only	csvserver that can be bound to filterpolicy.
filterpolicy_lbvserver_binding	<filterpolicy_lbvserver_binding[]>	Read-only	lbvserver that can be bound to filterpolicy.
filterpolicy_crvserver_binding	<filterpolicy_crvserver_binding[]>	Read-only	crvserver that can be bound to filterpolicy.

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
filterpolicy_filterglobal_binding	<filterpolicy_filterglobal_binding[]>	Read-only	filterglobal that can be bound to filterpolicy.

Operations

(click to see Properties)

GET| GET (ALL)

Some options that you can use for each operations:

- **Getting warnings in response:**NITRO allows you to get warnings in an operation by specifying the "warning" query parameter as "yes". For example, to get warnings while connecting to the Citrix ADC appliance, the URL is as follows:

  http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/login?warning=yes

  If any, the warnings are displayed in the response payload with the HTTP code "209 X-NITRO-WARNING".

- **Authenticated access for individual NITRO operations:**NITRO allows you to logon to the Citrix ADC appliance to perform individual operations. You can use this option instead of creating a NITRO session (using the login object) and then using that session to perform all operations,

  To do this, you must specify the username and password in the request header of the NITRO request as follows:

  X-NITRO-USER:<username>

  X-NITRO-PASS:<password>

  Note: In such cases, make sure that the request header DOES not include the following:

  Cookie:NITRO_AUTH_TOKEN=<tokenvalue>

`*Note:`*

Mandatory parameters are marked in redand placeholder content is marked in <green>.

get

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/filterpolicy_binding/name_value<String>

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "filterpolicy_binding": [ {

"name":<String_value>,

"filterpolicy_crvserver_binding":<filterpolicy_crvserver_binding[]_value>,

"filterpolicy_filterglobal_binding":<filterpolicy_filterglobal_binding[]_v
alue>,

"filterpolicy_lbvserver_binding":<filterpolicy_lbvserver_binding[]_value>,

"filterpolicy_csvserver_binding":<filterpolicy_csvserver_binding[]_value>

}]}
```

get (all)

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/filterpolicy_binding

Query-parameters:

bulkbindings

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/filterpolicy_binding?**bulkbindings=yes**

NITRO allows you to fetch bindings in bulk.

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "filterpolicy_binding": [ {

"name":<String_value>,

"filterpolicy_crvserver_binding":<filterpolicy_crvserver_binding[]_value>,

"filterpolicy_filterglobal_binding":<filterpolicy_filterglobal_binding[]_v
alue>,

"filterpolicy_lbvserver_binding":<filterpolicy_lbvserver_binding[]_value>,

"filterpolicy_csvserver_binding":<filterpolicy_csvserver_binding[]_value>

}]}
```

filterpolicy_crvserver_binding

Binding object showing the crvserver that can be bound to filterpolicy.

Properties

(click to see Operations)

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
boundto	<String>	Read-write	The entity name to which policy is bound.
name	<String>	Read-write	Name of the filter policy to be displayed. If a name is not provided, information about all the filter policies is shown. Minimum length = 1
priority	<Double>	Read-only	.
activepolicy	<Double>	Read-only	.
__count	<Double>	Read-only	count parameter

Operations

(click to see Properties)

GET| GET (ALL)| COUNT

Some options that you can use for each operations:

- **Getting warnings in response:**NITRO allows you to get warnings in an operation by specifying the "warning" query parameter as "yes". For example, to get warnings while connecting to the Citrix ADC appliance, the URL is as follows:

  http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/login?warning=yes

  If any, the warnings are displayed in the response payload with the HTTP code "209 X-NITRO-WARNING".

- **Authenticated access for individual NITRO operations:**NITRO allows you to logon to the Citrix ADC appliance to perform individual operations. You can use this option instead of

creating a NITRO session (using the login object) and then using that session to perform all operations,

To do this, you must specify the username and password in the request header of the NITRO request as follows:

X-NITRO-USER:<username>

X-NITRO-PASS:<password>

Note: In such cases, make sure that the request header DOES not include the following:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue>

`*Note:`*

Mandatory parameters are marked in redand placeholder content is marked in <green>.

get

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/filterpolicy_crvserver_binding/name_value<String>

Query-parameters:

filter

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/filterpolicy_crvserver_binding/name_value<String>?**filter=property-name1:property-value1,property-name2:property-value2**

Use this query-parameter to get the filtered set of filterpolicy_crvserver_binding resources configured on Citrix ADC. Filtering can be done on any of the properties of the resource.

pagination

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/filterpolicy_crvserver_binding/name_value<String>?**pagesize=#no;pageno=#no**

Use this query-parameter to get the filterpolicy_crvserver_binding resources in chunks.

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "filterpolicy_crvserver_binding": [ {

"boundto":<String_value>,

"name":<String_value>,

"priority":<Double_value>,

"activepolicy":<Double_value>

}]}
```

get (all)

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/filterpolicy_crvserver_binding

Query-parameters:

bulkbindings

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/filterpolicy_crvserver_binding?**bulkbindings=yes**

NITRO allows you to fetch bindings in bulk.

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "filterpolicy_crvserver_binding": [ {

"boundto":<String_value>,

"name":<String_value>,

"priority":<Double_value>,

"activepolicy":<Double_value>

}]}
```

count

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/filterpolicy_crvserver_binding/name_value<String>?**count=yes**

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{"filterpolicy_crvserver_binding": [ { "__count": "#no"} ] }
```

filterpolicy_csvserver_binding

Binding object showing the csvserver that can be bound to filterpolicy.

Properties

(click to see Operations)

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
boundto	<String>	Read-write	The entity name to which policy is bound.
name	<String>	Read-write	Name of the filter policy to be displayed. If a name is not provided, information about all the filter policies is shown. Minimum length = 1
priority	<Double>	Read-only	.
activepolicy	<Double>	Read-only	.
__count	<Double>	Read-only	count parameter

Operations

(click to see Properties)

GET| GET (ALL)| COUNT

Some options that you can use for each operations:

- **Getting warnings in response:**NITRO allows you to get warnings in an operation by specifying the "warning" query parameter as "yes". For example, to get warnings while connecting to the Citrix ADC appliance, the URL is as follows:

  http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/login?warning=yes

  If any, the warnings are displayed in the response payload with the HTTP code "209 X-NITRO-WARNING".

- **Authenticated access for individual NITRO operations:**NITRO allows you to logon to the Citrix ADC appliance to perform individual operations. You can use this option instead of creating a NITRO session (using the login object) and then using that session to perform all operations,

  To do this, you must specify the username and password in the request header of the NITRO request as follows:

  X-NITRO-USER:<username>

  X-NITRO-PASS:<password>

  Note: In such cases, make sure that the request header DOES not include the following:

  Cookie:NITRO_AUTH_TOKEN=<tokenvalue>

`*Note:`*

Mandatory parameters are marked in redand placeholder content is marked in <green>.

get

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/filterpolicy_csvserver_binding/name_value<String>

Query-parameters:

filter

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/filterpolicy_csvserver_binding/name_value<String>?**filter=property-name1:property-value1,property-name2:property-value2**

Use this query-parameter to get the filtered set of filterpolicy_csvserver_binding resources configured on Citrix ADC. Filtering can be done on any of the properties of the resource.

pagination

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/filterpolicy_csvserver_binding/name_value<String>?**pagesize=#no;pageno=#no**

Use this query-parameter to get the filterpolicy_csvserver_binding resources in chunks.

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "filterpolicy_csvserver_binding": [ {

"boundto":<String_value>,

"name":<String_value>,

"priority":<Double_value>,

"activepolicy":<Double_value>

}]}
```

get (all)

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/filterpolicy_csvserver_binding

Query-parameters:

bulkbindings

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/filterpolicy_csvserver_binding?**bulkbindings=yes**

NITRO allows you to fetch bindings in bulk.

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "filterpolicy_csvserver_binding": [ {

"boundto":<String_value>,

"name":<String_value>,

"priority":<Double_value>,

"activepolicy":<Double_value>

}]}
```

count

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/filterpolicy_csvserver_binding/name_value<String>?**count=yes**

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{"filterpolicy_csvserver_binding": [ { "__count": "#no"} ] }
```

filterpolicy_filterglobal_binding

Binding object showing the filterglobal that can be bound to filterpolicy.

Properties

(click to see Operations)

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
boundto	<String>	Read-write	The entity name to which policy is bound.
name	<String>	Read-write	Name of the filter policy to be displayed. If a name is not provided, information about all the filter policies is shown. Minimum length = 1
priority	<Double>	Read-only	.
activepolicy	<Double>	Read-only	.

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
__count	<Double>	Read-only	count parameter

Operations

(click to see Properties)

GET| GET (ALL)| COUNT

Some options that you can use for each operations:

- **Getting warnings in response:**NITRO allows you to get warnings in an operation by specifying the "warning" query parameter as "yes". For example, to get warnings while connecting to the Citrix ADC appliance, the URL is as follows:

  http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/login?warning=yes

  If any, the warnings are displayed in the response payload with the HTTP code "209 X-NITRO-WARNING".
- **Authenticated access for individual NITRO operations:**NITRO allows you to logon to the Citrix ADC appliance to perform individual operations. You can use this option instead of creating a NITRO session (using the login object) and then using that session to perform all operations,

  To do this, you must specify the username and password in the request header of the NITRO request as follows:

  X-NITRO-USER:<username>

  X-NITRO-PASS:<password>

  Note: In such cases, make sure that the request header DOES not include the following:

  Cookie:NITRO_AUTH_TOKEN=<tokenvalue>

`*Note:`*

Mandatory parameters are marked in redand placeholder content is marked in <green>.

get

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/filterpolicy_filterglobal_binding/name_value<String>

Query-parameters:

filter

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/filterpolicy_filterglobal_binding/name_value<String>?**filter=property-name1:property-value1,property-name2:property-value2**

Use this query-parameter to get the filtered set of filterpolicy_filterglobal_binding resources configured on Citrix ADC. Filtering can be done on any of the properties of the resource.

pagination

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/filterpolicy_filterglobal_binding/name_value<String>?**pagesize=#no;pageno=#no**

Use this query-parameter to get the filterpolicy_filterglobal_binding resources in chunks.

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "filterpolicy_filterglobal_binding": [ {

"boundto":<String_value>,

"name":<String_value>,

"priority":<Double_value>,

"activepolicy":<Double_value>
```

```
}]}
```

get (all)

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/filterpolicy_filterglobal_binding

Query-parameters:

bulkbindings

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/filterpolicy_filterglobal_binding?**bulkbindings=yes**

NITRO allows you to fetch bindings in bulk.

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "filterpolicy_filterglobal_binding": [ {

"boundto":<String_value>,

"name":<String_value>,

"priority":<Double_value>,

"activepolicy":<Double_value>
```

```
}]}
```

count

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/filterpolicy_filterglobal_binding/name_value<String>?**count=yes**

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{"filterpolicy_filterglobal_binding": [ { "__count": "#no"} ] }
```

filterpolicy_lbvserver_binding

Binding object showing the lbvserver that can be bound to filterpolicy.

Properties

(click to see Operations)

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
boundto	<String>	Read-write	The entity name to which policy is bound.

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
name	<String>	Read-write	Name of the filter policy to be displayed. If a name is not provided, information about all the filter policies is shown. Minimum length = 1
priority	<Double>	Read-only	.
activepolicy	<Double>	Read-only	.
__count	<Double>	Read-only	count parameter

Operations

(click to see Properties)

GET| GET (ALL)| COUNT

Some options that you can use for each operations:

- **Getting warnings in response:**NITRO allows you to get warnings in an operation by specifying the "warning" query parameter as "yes". For example, to get warnings while connecting to the Citrix ADC appliance, the URL is as follows:

  http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/login?warning=yes

  If any, the warnings are displayed in the response payload with the HTTP code "209 X-NITRO-WARNING".

- **Authenticated access for individual NITRO operations:**NITRO allows you to logon to the Citrix ADC appliance to perform individual operations. You can use this option instead of creating a NITRO session (using the login object) and then using that session to perform all operations,

  To do this, you must specify the username and password in the request header of the NITRO request as follows:

  X-NITRO-USER:<username>

X-NITRO-PASS:<password>

Note: In such cases, make sure that the request header DOES not include the following:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue>

`*Note:`*

Mandatory parameters are marked in redand placeholder content is marked in <green>.

get

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/filterpolicy_lbvserver_binding/name_value<String>

Query-parameters:

filter

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/filterpolicy_lbvserver_binding/name_value<String>?**filter=property-name1:property-value1,property-name2:property-value2**

Use this query-parameter to get the filtered set of filterpolicy_lbvserver_binding resources configured on Citrix ADC. Filtering can be done on any of the properties of the resource.

pagination

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/filterpolicy_lbvserver_binding/name_value<String>?**pagesize=#no;pageno=#no**

Use this query-parameter to get the filterpolicy_lbvserver_binding resources in chunks.

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "filterpolicy_lbvserver_binding": [ {

"boundto":<String_value>,

"name":<String_value>,

"priority":<Double_value>,

"activepolicy":<Double_value>

}]}
```

get (all)

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/filterpolicy_lbvserver_binding

Query-parameters:

bulkbindings

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/filterpolicy_lbvserver_binding?**bulkbindings=yes**

NITRO allows you to fetch bindings in bulk.

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "filterpolicy_lbvserver_binding": [ {

"boundto":<String_value>,

"name":<String_value>,

"priority":<Double_value>,

"activepolicy":<Double_value>

}]}
```

count

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/filterpolicy_lbvserver_binding/name_value<String>?**count=yes**

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{"filterpolicy_lbvserver_binding": [ { "__count": "#no"} ] }
```

filterpostbodyinjection

Configuration for HTML Injection postbody resource.

Properties

(click to see Operations)

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
postbody	<String>	Read-write	Name of file whose contents are to be inserted after the response body. Minimum length = 1
systemiid	<String>	Read-only	The system IID of the current Citrix ADC system.

Operations

(click to see Properties)

UPDATE| UNSET| GET (ALL)

Some options that you can use for each operations:

- **Getting warnings in response:**NITRO allows you to get warnings in an operation by specifying the "warning" query parameter as "yes". For example, to get warnings while connecting to the Citrix ADC appliance, the URL is as follows:

  http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/login?warning=yes

  If any, the warnings are displayed in the response payload with the HTTP code "209 X-NITRO-WARNING".

- **Authenticated access for individual NITRO operations:**NITRO allows you to logon to the Citrix ADC appliance to perform individual operations. You can use this option instead of creating a NITRO session (using the login object) and then using that session to perform all operations,

  To do this, you must specify the username and password in the request header of the NITRO request as follows:

X-NITRO-USER:<username>

X-NITRO-PASS:<password>

Note: In such cases, make sure that the request header DOES not include the following:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue>

`*Note:`*

Mandatory parameters are marked in redand placeholder content is marked in <green>.

update

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/filterpostbodyinjection

HTTP Method:PUT

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Content-Type:application/json

Request Payload:

```
{"filterpostbodyinjection":{

<b>"postbody":<String_value>

</b>}}
```

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error

unset

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/filterpostbodyinjection?**action=unset**

HTTP Method:POST

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Content-Type:application/json

Request Payload:

```
{"filterpostbodyinjection":{

"postbody":true

}}
```

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error

get (all)

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/filterpostbodyinjection

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "filterpostbodyinjection": [ {

"postbody":<String_value>,
```

```
"systemiid":<String_value>

}]}
```

filterprebodyinjection

Configuration for HTML Injection prebody resource.

Properties

(click to see Operations)

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
prebody	<String>	Read-write	Name of file whose contents are to be inserted before the response body. Minimum length = 1
systemiid	<String>	Read-only	The system IID of the current Citrix ADC system.

Operations

(click to see Properties)

UPDATE| UNSET| GET (ALL)

Some options that you can use for each operations:

- **Getting warnings in response:**NITRO allows you to get warnings in an operation by specifying the "warning" query parameter as "yes". For example, to get warnings while connecting to the Citrix ADC appliance, the URL is as follows:

  http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/login?warning=yes

  If any, the warnings are displayed in the response payload with the HTTP code "209 X-NITRO-WARNING".

- **Authenticated access for individual NITRO operations:**NITRO allows you to logon to the Citrix ADC appliance to perform individual operations. You can use this option instead of creating a NITRO session (using the login object) and then using that session to perform all operations,

  To do this, you must specify the username and password in the request header of the NITRO request as follows:

  X-NITRO-USER:<username>

  X-NITRO-PASS:<password>

  Note: In such cases, make sure that the request header DOES not include the following:

  Cookie:NITRO_AUTH_TOKEN=<tokenvalue>

`*Note:`*

Mandatory parameters are marked in redand placeholder content is marked in <green>.

update

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/filterprebodyinjection

HTTP Method:PUT

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Content-Type:application/json

Request Payload:

```
{"filterprebodyinjection":{

<b>"prebody":<String_value>

</b>}}
```

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error

unset

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/filterprebodyinjection?**action=unset**

HTTP Method:POST

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Content-Type:application/json

Request Payload:

```
{"filterprebodyinjection":{

"prebody":true

}}
```

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error

get (all)

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/filterprebodyinjection

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "filterprebodyinjection": [ {

"prebody":<String_value>,

"systemiid":<String_value>

}]}
```

Front-end Optimization

Front-end-optimization

Front end optimization configuration. The system's feature to optimize Web content for performance.

NAME	DESCRIPTION
feoaction	Configuration for Front end optimization action
feoglobal_binding	Binding object showing the resources that can be bound to feoglobal
feoglobal_feopolicy_binding	Binding object showing the feopolicy that can be bound to feoglobal
feoparameter	Configuration for FEO parameter
feopolicy	Configuration for Front end optimization policy
feopolicy_binding	Binding object showing the resources that can be bound to feopolicy

NAME	DESCRIPTION
feopolicy_csvserver_binding	Binding object showing the csvserver that can be bound to feopolicy
feopolicy_feoglobal_binding	Binding object showing the feoglobal that can be bound to feopolicy
feopolicy_lbvserver_binding	Binding object showing the lbvserver that can be bound to feopolicy

feoaction

Configuration for Front end optimization action resource.

Properties

(click to see Operations)

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
name	<String>	Read-write	The name of the front end optimization action. Minimum length = 1
pageextendcache	<Boolean>	Read-write	Extend the time period during which the browser can use the cached resource.
cachemaxage	<Double>	Read-write	Maxage for cache extension. Default value: 30 Minimum value = 0 Maximum value = 360
imgshrinktoattrib	<Boolean>	Read-write	Shrink image dimensions as per the height and width

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
			attributes specified in the <img> tag.
imggiftopng	<Boolean>	Read-write	Convert GIF image formats to PNG formats.
imgtowebp	<Boolean>	Read-write	Convert JPEG, GIF, PNG image formats to WEBP format.
imgtojpegxr	<Boolean>	Read-write	Convert JPEG, GIF, PNG image formats to JXR format.
imginline	<Boolean>	Read-write	Inline images whose size is less than 2KB.
cssimginline	<Boolean>	Read-write	Inline small images (less than 2KB) referred within CSS files as background-URLs.
jpgoptimize	<Boolean>	Read-write	Remove non-image data such as comments from JPEG images.
imglazyload	<Boolean>	Read-write	Download images, only when the user scrolls the page to view them.
cssminify	<Boolean>	Read-write	Remove comments and whitespaces from CSSs.
cssinline	<Boolean>	Read-write	Inline CSS files, whose size is less

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
			than 2KB, within the main page.
csscombine	<Boolean>	Read-write	Combine one or more CSS files into one file.
convertimporttolink	<Boolean>	Read-write	Convert CSS import statements to HTML link tags.
jsminify	<Boolean>	Read-write	Remove comments and whitespaces from JavaScript.
jsinline	<Boolean>	Read-write	Convert linked JavaScript files (less than 2KB) to inline JavaScript files.
htmlminify	<Boolean>	Read-write	Remove comments and whitespaces from an HTML page.
cssmovetohead	<Boolean>	Read-write	Move any CSS file present within the body tag of an HTML page to the head tag.
jsmovetoend	<Boolean>	Read-write	Move any JavaScript present in the body tag to the end of the body tag.
domainsharding	<String>	Read-write	Domain name of the server.
dnsshards	<String[]>	Read-write	Set of domain names that replaces the parent domain.

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
clientsidemeasurements	<Boolean>	Read-write	Send AppFlow records about the web pages optimized by this action. The records provide FEO statistics, such as the number of HTTP requests that have been reduced for this page. You must enable the Appflow feature before enabling this parameter.
imgadddimensions	<Boolean>	Read-only	Add dimension attributes to images, if not specified within the <img> tag.
imgshrinkformobile	<Boolean>	Read-only	Serve smaller images for mobile users.
imgweaken	<Boolean>	Read-only	Reduce the image quality.
jpgprogressive	<Boolean>	Read-only	Convert JPEG image formats to progressive formats.
cssflattenimports	<Boolean>	Read-only	Replace CSS import statements with the file content.
jscombine	<Boolean>	Read-only	Combine one or more JavaScript files into one file.

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
htmlrmdefaultattribs	<Boolean>	Read-only	Remove default redundant attributes from an HTML file.
htmlrmattribquotes	<Boolean>	Read-only	Remove unnecessary quotes present within the HTML attributes.
htmltrimurls	<Boolean>	Read-only	Trim URLs.
hits	<Double>	Read-only	The number of times the action has been taken.
undefhits	<Double>	Read-only	Total number of undefined policy hits.
__count	<Double>	Read-only	count parameter

Operations

(click to see Properties)

ADD| UPDATE| UNSET| DELETE| GET (ALL)| GET| COUNT

Some options that you can use for each operations:

- **Getting warnings in response:**NITRO allows you to get warnings in an operation by specifying the "warning" query parameter as "yes". For example, to get warnings while connecting to the Citrix ADC appliance, the URL is as follows:

  http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/login?warning=yes

  If any, the warnings are displayed in the response payload with the HTTP code "209 X-NITRO-WARNING".
- **Authenticated access for individual NITRO operations:**NITRO allows you to logon to the Citrix ADC appliance to perform individual operations. You can use this option instead of creating a NITRO session (using the login object) and then using that session to perform all operations,

To do this, you must specify the username and password in the request header of the NITRO request as follows:

X-NITRO-USER:<username>

X-NITRO-PASS:<password>

Note: In such cases, make sure that the request header DOES not include the following:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue>

`*Note:`*

Mandatory parameters are marked in redand placeholder content is marked in <green>.

add

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/feoaction

HTTP Method:POST

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Content-Type:application/json

Request Payload:

```
{"feoaction":{

<b>"name":<String_value>,

</b>"pageextendcache":<Boolean_value>,

"cachemaxage":<Double_value>,

"imgshrinktoattrib":<Boolean_value>,

"imggiftopng":<Boolean_value>,

"imgtowebp":<Boolean_value>,

"imgtojpegxr":<Boolean_value>,
```

```
"imginline":<Boolean_value>,

"cssimginline":<Boolean_value>,

"jpgoptimize":<Boolean_value>,

"imglazyload":<Boolean_value>,

"cssminify":<Boolean_value>,

"cssinline":<Boolean_value>,

"csscombine":<Boolean_value>,

"convertimporttolink":<Boolean_value>,

"jsminify":<Boolean_value>,

"jsinline":<Boolean_value>,

"htmlminify":<Boolean_value>,

"cssmovetohead":<Boolean_value>,

"jsmovetoend":<Boolean_value>,

"domainsharding":<String_value>,

"dnsshards":<String[]_value>,

"clientsidemeasurements":<Boolean_value>

}}
```

Response:

HTTP Status Code on Success: 201 Created HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error

update

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/feoaction

HTTP Method:PUT

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Content-Type:application/json

Request Payload:

```
{"feoaction":{

<b>"name":<String_value>,

</b>"pageextendcache":<Boolean_value>,

"cachemaxage":<Double_value>,

"imgshrinktoattrib":<Boolean_value>,

"imggiftopng":<Boolean_value>,

"imgtowebp":<Boolean_value>,

"imgtojpegxr":<Boolean_value>,

"imginline":<Boolean_value>,

"cssimginline":<Boolean_value>,

"jpgoptimize":<Boolean_value>,

"imglazyload":<Boolean_value>,

"cssminify":<Boolean_value>,

"cssinline":<Boolean_value>,

"csscombine":<Boolean_value>,
```

```
"convertimporttolink":<Boolean_value>,

"jsminify":<Boolean_value>,

"jsinline":<Boolean_value>,

"htmlminify":<Boolean_value>,

"cssmovetohead":<Boolean_value>,

"jsmovetoend":<Boolean_value>,

"domainsharding":<String_value>,

"dnsshards":<String[]_value>,

"clientsidemeasurements":<Boolean_value>

}}
```

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error

unset

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/feoaction?**action=unset**

HTTP Method:POST

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Content-Type:application/json

Request Payload:

```
{"feoaction":{
```

```
<b>"name":<String_value>,

</b>"pageextendcache":true,

"imgshrinktoattrib":true,

"imggiftopng":true,

"imgtowebp":true,

"imgtojpegxr":true,

"imginline":true,

"cssimginline":true,

"jpgoptimize":true,

"imglazyload":true,

"cssminify":true,

"cssinline":true,

"csscombine":true,

"convertimporttolink":true,

"jsminify":true,

"jsinline":true,

"htmlminify":true,

"cssmovetohead":true,

"jsmovetoend":true,

"clientsidemeasurements":true,

"domainsharding":true

}}
```

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error

delete

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/feoaction/name_value<String>

HTTP Method:DELETE

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue>

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error

get (all)

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/feoaction

Query-parameters:

attrs

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/feoaction?**attrs=property-name1,property-name2**

Use this query parameter to specify the resource details that you want to retrieve.

filter

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/feoaction?**filter=property-name1:property-val1,property-name2:property-val2**

Use this query-parameter to get the filtered set of feoaction resources configured on Citrix ADC. Filtering can be done on any of the properties of the resource.

view

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/feoaction?**view=summary**

Note: By default, the retrieved results are displayed in detail view (?view=detail).

pagination

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/feoaction?**pagesize=#no;pageno=#no**

Use this query-parameter to get the feoaction resources in chunks.

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "feoaction": [ {

"name":<String_value>,

"pageextendcache":<Boolean_value>,

"cachemaxage":<Double_value>,

"imgshrinktoattrib":<Boolean_value>,

"imggiftopng":<Boolean_value>,

"imgtowebp":<Boolean_value>,

"imgtojpegxr":<Boolean_value>,

"imgadddimensions":<Boolean_value>,
```

```
"imgshrinkformobile":<Boolean_value>,

"imgweaken":<Boolean_value>,

"imginline":<Boolean_value>,

"cssimginline":<Boolean_value>,

"jpgoptimize":<Boolean_value>,

"jpgprogressive":<Boolean_value>,

"imglazyload":<Boolean_value>,

"cssminify":<Boolean_value>,

"cssinline":<Boolean_value>,

"csscombine":<Boolean_value>,

"cssflattenimports":<Boolean_value>,

"convertimporttolink":<Boolean_value>,

"jsminify":<Boolean_value>,

"jsinline":<Boolean_value>,

"jscombine":<Boolean_value>,

"htmlminify":<Boolean_value>,

"htmlrmdefaultattribs":<Boolean_value>,

"htmlrmattribquotes":<Boolean_value>,

"htmltrimurls":<Boolean_value>,

"cssmovetohead":<Boolean_value>,

"jsmovetoend":<Boolean_value>,

"domainsharding":<String_value>,
```

```
"dnsshards":<String[]_value>,

"clientsidemeasurements":<Boolean_value>,

"hits":<Double_value>,

"undefhits":<Double_value>

}]}
```

get

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/feoaction/name_value<String>

Query-parameters:

attrs

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/feoaction/name_value<String>?**attrs=property-name1,property-name2**

Use this query parameter to specify the resource details that you want to retrieve.

view

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/feoaction/name_value<String>?**view=summary**

Note: By default, the retrieved results are displayed in detail view (?view=detail).

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "feoaction": [ {

"name":<String_value>,

"pageextendcache":<Boolean_value>,

"cachemaxage":<Double_value>,

"imgshrinktoattrib":<Boolean_value>,

"imggiftopng":<Boolean_value>,

"imgtowebp":<Boolean_value>,

"imgtojpegxr":<Boolean_value>,

"imgadddimensions":<Boolean_value>,

"imgshrinkformobile":<Boolean_value>,

"imgweaken":<Boolean_value>,

"imginline":<Boolean_value>,

"cssimginline":<Boolean_value>,

"jpgoptimize":<Boolean_value>,

"jpgprogressive":<Boolean_value>,

"imglazyload":<Boolean_value>,

"cssminify":<Boolean_value>,

"cssinline":<Boolean_value>,

"csscombine":<Boolean_value>,

"cssflattenimports":<Boolean_value>,
```

```
"convertimporttolink":<Boolean_value>,

"jsminify":<Boolean_value>,

"jsinline":<Boolean_value>,

"jscombine":<Boolean_value>,

"htmlminify":<Boolean_value>,

"htmlrmdefaultattribs":<Boolean_value>,

"htmlrmattribquotes":<Boolean_value>,

"htmltrimurls":<Boolean_value>,

"cssmovetohead":<Boolean_value>,

"jsmovetoend":<Boolean_value>,

"domainsharding":<String_value>,

"dnsshards":<String[]_value>,

"clientsidemeasurements":<Boolean_value>,

"hits":<Double_value>,

"undefhits":<Double_value>

}]}
```

count

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/feoaction?**count=yes**

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "feoaction": [ { "__count": "#no"} ] }
```

feoglobal_binding

Binding object which returns the resources bound to feoglobal.

Properties

(click to see Operations)

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
feoglobal_feopolicy_binding	<feoglobal_feopolicy_binding[]>	Read-only	feopolicy that can be bound to feoglobal.

Operations

(click to see Properties)

GET| GET (ALL)

Some options that you can use for each operations:

- **Getting warnings in response:**NITRO allows you to get warnings in an operation by specifying the "warning" query parameter as "yes". For example, to get warnings while connecting to the Citrix ADC appliance, the URL is as follows:

  http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/login?warning=yes

  If any, the warnings are displayed in the response payload with the HTTP code "209 X-NITRO-WARNING".

- **Authenticated access for individual NITRO operations:**NITRO allows you to logon to the Citrix ADC appliance to perform individual operations. You can use this option instead of creating a NITRO session (using the login object) and then using that session to perform all operations,

  To do this, you must specify the username and password in the request header of the NITRO request as follows:

  X-NITRO-USER:<username>

  X-NITRO-PASS:<password>

  Note: In such cases, make sure that the request header DOES not include the following:

  Cookie:NITRO_AUTH_TOKEN=<tokenvalue>

`*Note:`*

Mandatory parameters are marked in redand placeholder content is marked in <green>.

get

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/feoglobal_binding

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "feoglobal_binding": [ {

"feoglobal_feopolicy_binding":<feoglobal_feopolicy_binding[]_value>
```

```
}]}
```

get (all)

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/feoglobal_binding

Query-parameters:

bulkbindings

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/feoglobal_binding?**bulkbindings=yes**

NITRO allows you to fetch bindings in bulk.

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "feoglobal_binding": [ {

"feoglobal_feopolicy_binding":<feoglobal_feopolicy_binding[]_value>

}]}
```

feoglobal_feopolicy_binding

Binding object showing the feopolicy that can be bound to feoglobal.

Properties

(click to see Operations)

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
priority	<Double>	Read-write	The priority assigned to the policy binding.
globalbindtype	<String>	Read-write	. Default value: SYSTEM_GLOBAL Possible values = SYSTEM_GLOBAL, VPN_GLOBAL, RNAT_GLOBAL
policyname	<String>	Read-write	The name of the globally bound front end optimization policy.
gotopriorityexpression	<String>	Read-write	Expression specifying the priority of the next policy which will get evaluated if the current policy rule evaluates to TRUE.
type	<String>	Read-write	Bindpoint to which the policy is bound. Possible values = REQ_OVERRIDE, REQ_DEFAULT, RES_OVERRIDE, RES_DEFAULT, NONE

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
numpol	<Double>	Read-only	The number of policies bound to the bindpoint.
__count	<Double>	Read-only	count parameter

Operations

(click to see Properties)

ADD:| DELETE:| GET| GET (ALL)| COUNT

Some options that you can use for each operations:

- **Getting warnings in response:**NITRO allows you to get warnings in an operation by specifying the "warning" query parameter as "yes". For example, to get warnings while connecting to the Citrix ADC appliance, the URL is as follows:

  http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/login?warning=yes

  If any, the warnings are displayed in the response payload with the HTTP code "209 X-NITRO-WARNING".

- **Authenticated access for individual NITRO operations:**NITRO allows you to logon to the Citrix ADC appliance to perform individual operations. You can use this option instead of creating a NITRO session (using the login object) and then using that session to perform all operations,

  To do this, you must specify the username and password in the request header of the NITRO request as follows:

  X-NITRO-USER:<username>

  X-NITRO-PASS:<password>

  Note: In such cases, make sure that the request header DOES not include the following:

  Cookie:NITRO_AUTH_TOKEN=<tokenvalue>

`*Note:`*

Mandatory parameters are marked in redand placeholder content is marked in <green>.

add

URL:http://<Citrix-ADC-IP-address(NSIP)/nitro/v1/config/feoglobal_feopolicy_binding

HTTP Method:PUT

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Content-Type:application/json

Request Payload:

```
{

"feoglobal_feopolicy_binding":{

<b>"policyname":<String_value>,

</b><b>"priority":<Double_value>,

</b>"type":<String_value>,

"gotopriorityexpression":<String_value>

}}
```

Response:

HTTP Status Code on Success: 201 Created HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error

delete

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/feoglobal_feopolicy_binding

Query-parameters:

args

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/feoglobal_feopolicy_binding?**args=policyname:<String_value>,type:<String_value>,priority:<Double_value>**

HTTP Method:DELETE

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue>

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error

get

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/feoglobal_feopolicy_binding

Query-parameters:

filter

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/feoglobal_feopolicy_binding?**filter=property-name1:property-value1,property-name2:property-value2**

Use this query-parameter to get the filtered set of feoglobal_feopolicy_binding resources configured on Citrix ADC. Filtering can be done on any of the properties of the resource.

pagination

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/feoglobal_feopolicy_binding?**pagesize=#no;pageno=#no**

Use this query-parameter to get the feoglobal_feopolicy_binding resources in chunks.

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "feoglobal_feopolicy_binding": [ {

"priority":<Double_value>,

"globalbindtype":<String_value>,

"policyname":<String_value>,

"gotopriorityexpression":<String_value>,

"type":<String_value>,

"numpol":<Double_value>

}]}
```

get (all)

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/feoglobal_feopolicy_binding

Query-parameters:

bulkbindings

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/feoglobal_feopolicy_binding?**bulkbindings=yes**

NITRO allows you to fetch bindings in bulk.

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "feoglobal_feopolicy_binding": [ {

"priority":<Double_value>,

"globalbindtype":<String_value>,

"policyname":<String_value>,

"gotopriorityexpression":<String_value>,

"type":<String_value>,

"numpol":<Double_value>

}]}
```

count

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/feoglobal_feopolicy_binding?**count=yes**

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "feoglobal_feopolicy_binding": [ { "__count": "#no"} ] }
```

feoparameter

Configuration for FEO parameter resource.

Properties

(click to see Operations)

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
jpegqualitypercent	<Double>	Read-write	The percentage value of a JPEG image quality to be reduced. Range: 0 - 100. Default value: 75 Minimum value = 0 Maximum value = 100
cssinlinethressize	<Double>	Read-write	Threshold value of the file size (in bytes) for converting external CSS files to inline CSS files. Default value: 1024 Minimum value = 1 Maximum value = 2048
jsinlinethressize	<Double>	Read-write	Threshold value of the file size (in

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
			bytes), for converting external JavaScript files to inline JavaScript files. Default value: 1024 Minimum value = 1 Maximum value = 2048
imginlinethressize	<Double>	Read-write	Maximum file size of an image (in bytes), for coverting linked images to inline images. Default value: 1024 Minimum value = 1 Maximum value = 2048

Operations

(click to see Properties)

UPDATE| UNSET| GET (ALL)

Some options that you can use for each operations:

- **Getting warnings in response:**NITRO allows you to get warnings in an operation by specifying the "warning" query parameter as "yes". For example, to get warnings while connecting to the Citrix ADC appliance, the URL is as follows:

  http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/login?warning=yes

  If any, the warnings are displayed in the response payload with the HTTP code "209 X-NITRO-WARNING".

- **Authenticated access for individual NITRO operations:**NITRO allows you to logon to the Citrix ADC appliance to perform individual operations. You can use this option instead of

creating a NITRO session (using the login object) and then using that session to perform all operations,

To do this, you must specify the username and password in the request header of the NITRO request as follows:

X-NITRO-USER:<username>

X-NITRO-PASS:<password>

Note: In such cases, make sure that the request header DOES not include the following:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue>

`*Note:`*

Mandatory parameters are marked in redand placeholder content is marked in <green>.

update

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/feoparameter

HTTP Method:PUT

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Content-Type:application/json

Request Payload:

```
{"feoparameter":{

"jpegqualitypercent":<Double_value>,

"cssinlinethressize":<Double_value>,

"jsinlinethressize":<Double_value>,

"imginlinethressize":<Double_value>

}}
```

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error

unset

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/feoparameter?**action=unset**

HTTP Method:POST

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Content-Type:application/json

Request Payload:

```
{"feoparameter":{

"jpegqualitypercent":true,

"cssinlinethressize":true,

"jsinlinethressize":true,

"imginlinethressize":true

}}
```

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error

get (all)

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/feoparameter

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "feoparameter": [ {

"jpegqualitypercent":<Double_value>,

"cssinlinethressize":<Double_value>,

"jsinlinethressize":<Double_value>,

"imginlinethressize":<Double_value>

}]}
```

feopolicy

Configuration for Front end optimization policy resource.

Properties

(click to see Operations)

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
name	<String>	Read-write	The name of the front end optimization policy. Minimum length = 1

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
rule	<String>	Read-write	The rule associated with the front end optimization policy.
action	<String>	Read-write	The front end optimization action that has to be performed when the rule matches. Minimum length = 1
hits	<Double>	Read-only	Total number of hits.
undefhits	<Double>	Read-only	Total number of undefined policy hits.
__count	<Double>	Read-only	count parameter

Operations

(click to see Properties)

ADD| DELETE| UPDATE| UNSET| GET (ALL)| GET| COUNT

Some options that you can use for each operations:

- **Getting warnings in response:**NITRO allows you to get warnings in an operation by specifying the "warning" query parameter as "yes". For example, to get warnings while connecting to the Citrix ADC appliance, the URL is as follows:

  http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/login?warning=yes

  If any, the warnings are displayed in the response payload with the HTTP code "209 X-NITRO-WARNING".

- **Authenticated access for individual NITRO operations:**NITRO allows you to logon to the Citrix ADC appliance to perform individual operations. You can use this option instead of creating a NITRO session (using the login object) and then using that session to perform all operations,

  To do this, you must specify the username and password in the request header of the NITRO request as follows:

X-NITRO-USER:<username>

X-NITRO-PASS:<password>

Note: In such cases, make sure that the request header DOES not include the following:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue>

`*Note:`*

Mandatory parameters are marked in redand placeholder content is marked in <green>.

add

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/feopolicy

HTTP Method:POST

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Content-Type:application/json

Request Payload:

```
{"feopolicy":{

<b>"name":<String_value>,

</b><b>"rule":<String_value>,

</b><b>"action":<String_value>

</b>}}
```

Response:

HTTP Status Code on Success: 201 Created HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error

delete

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/feopolicy/name_value<String>

HTTP Method:DELETE

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue>

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error

update

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/feopolicy

HTTP Method:PUT

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Content-Type:application/json

Request Payload:

```
{"feopolicy":{

<b>"name":<String_value>,

</b>"rule":<String_value>,

"action":<String_value>

}}
```

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error

unset

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/feopolicy?**action=unset**

HTTP Method:POST

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Content-Type:application/json

Request Payload:

```
{"feopolicy":{

<b>"name":<String_value>,

</b>"rule":true,

"action":true

}}
```

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error

get (all)

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/feopolicy

Query-parameters:

attrs

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/feopolicy?**attrs=property-name1,property-name2**

Use this query parameter to specify the resource details that you want to retrieve.

filter

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/feopolicy?**filter=property-name1:property-val1,property-name2:property-val2**

Use this query-parameter to get the filtered set of feopolicy resources configured on Citrix ADC. Filtering can be done on any of the properties of the resource.

view

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/feopolicy?**view=summary**

Note: By default, the retrieved results are displayed in detail view (?view=detail).

pagination

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/feopolicy?**pagesize=#no;pageno=#no**

Use this query-parameter to get the feopolicy resources in chunks.

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "feopolicy": [ {

"name":<String_value>,

"rule":<String_value>,

"action":<String_value>,

"hits":<Double_value>,
```

```
"undefhits":<Double_value>

}]}
```

get

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/feopolicy/name_value<String>

Query-parameters:

attrs

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/feopolicy/name_value<String>?**attrs=property-name1,property-name2**

Use this query parameter to specify the resource details that you want to retrieve.

view

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/feopolicy/name_value<String>?**view=summary**

Note: By default, the retrieved results are displayed in detail view (?view=detail).

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "feopolicy": [ {
```

```
"name":<String_value>,

"rule":<String_value>,

"action":<String_value>,

"hits":<Double_value>,

"undefhits":<Double_value>

}]}
```

count

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/feopolicy?**count=yes**

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "feopolicy": [ { "__count": "#no"} ] }
```

feopolicy_binding

Binding object which returns the resources bound to feopolicy.

Properties

(click to see Operations)

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
name	<String>	Read-write	The name of the front end optimization policy. Minimum length = 1
feopolicy_csvserver_binding	<feopolicy_csvserver_binding[]>	Read-only	csvserver that can be bound to feopolicy.
feopolicy_lbvserver_binding	<feopolicy_lbvserver_binding[]>	Read-only	lbvserver that can be bound to feopolicy.
feopolicy_feoglobal_binding	<feopolicy_feoglobal_binding[]>	Read-only	feoglobal that can be bound to feopolicy.

Operations

(click to see Properties)

GET| GET (ALL)

Some options that you can use for each operations:

- **Getting warnings in response:**NITRO allows you to get warnings in an operation by specifying the "warning" query parameter as "yes". For example, to get warnings while connecting to the Citrix ADC appliance, the URL is as follows:

  http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/login?warning=yes

  If any, the warnings are displayed in the response payload with the HTTP code "209 X-NITRO-WARNING".
- **Authenticated access for individual NITRO operations:**NITRO allows you to logon to the Citrix ADC appliance to perform individual operations. You can use this option instead of creating a NITRO session (using the login object) and then using that session to perform all operations,

To do this, you must specify the username and password in the request header of the NITRO request as follows:

X-NITRO-USER:<username>

X-NITRO-PASS:<password>

Note: In such cases, make sure that the request header DOES not include the following:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue>

`*Note:`*

Mandatory parameters are marked in redand placeholder content is marked in <green>.

get

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/feopolicy_binding/name_value<String>

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "feopolicy_binding": [ {

"name":<String_value>,

"feopolicy_csvserver_binding":<feopolicy_csvserver_binding[]_value>,

"feopolicy_lbvserver_binding":<feopolicy_lbvserver_binding[]_value>,

"feopolicy_feoglobal_binding":<feopolicy_feoglobal_binding[]_value>
```

```
}]}
```

get (all)

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/feopolicy_binding

Query-parameters:

bulkbindings

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/feopolicy_binding?**bulkbindings=yes**

NITRO allows you to fetch bindings in bulk.

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "feopolicy_binding": [ {

"name":<String_value>,

"feopolicy_csvserver_binding":<feopolicy_csvserver_binding[]_value>,

"feopolicy_lbvserver_binding":<feopolicy_lbvserver_binding[]_value>,

"feopolicy_feoglobal_binding":<feopolicy_feoglobal_binding[]_value>

}]}
```

feopolicy_csvserver_binding

Binding object showing the csvserver that can be bound to feopolicy.

Properties

(click to see Operations)

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
boundto	<String>	Read-write	Location where the policy is bound to.
name	<String>	Read-write	The name of the front end optimization policy. Minimum length = 1
priority	<Double>	Read-only	Priority of the policy.
gotopriorityexpression	<String>	Read-only	Expression specifying the priority of the next policy which will get evaluated if the current policy rule evaluates to TRUE.
activepolicy	<Integer>	Read-only	Indicates whether a policy is bound or not.
__count	<Double>	Read-only	count parameter

Operations

(click to see Properties)

GET| GET (ALL)| COUNT

Some options that you can use for each operations:

- **Getting warnings in response:**NITRO allows you to get warnings in an operation by specifying the "warning" query parameter as "yes". For example, to get warnings while connecting to the Citrix ADC appliance, the URL is as follows:

  http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/login?warning=yes

  If any, the warnings are displayed in the response payload with the HTTP code "209 X-NITRO-WARNING".

- **Authenticated access for individual NITRO operations:**NITRO allows you to logon to the Citrix ADC appliance to perform individual operations. You can use this option instead of creating a NITRO session (using the login object) and then using that session to perform all operations,

  To do this, you must specify the username and password in the request header of the NITRO request as follows:

  X-NITRO-USER:<username>

  X-NITRO-PASS:<password>

  Note: In such cases, make sure that the request header DOES not include the following:

  Cookie:NITRO_AUTH_TOKEN=<tokenvalue>

`*Note:`*

Mandatory parameters are marked in redand placeholder content is marked in <green>.

get

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/feopolicy_csvserver_binding/name_value<String>

Query-parameters:

filter

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/feopolicy_csvserver_binding/name_value<String>?**filter=property-name1:property-value1,property-name2:property-value2**

Use this query-parameter to get the filtered set of feopolicy_csvserver_binding resources configured on Citrix ADC. Filtering can be done on any of the properties of the resource.

pagination

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/feopolicy_csvserver_binding/name_value<String>?**pagesize=#no;pageno=#no**

Use this query-parameter to get the feopolicy_csvserver_binding resources in chunks.

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "feopolicy_csvserver_binding": [ {

"boundto":<String_value>,

"name":<String_value>,

"priority":<Double_value>,

"gotopriorityexpression":<String_value>,

"activepolicy":<Integer_value>

}]}
```

get (all)

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/feopolicy_csvserver_binding

Query-parameters:

bulkbindings

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/feopolicy_csvserver_binding?**bulkbindings=yes**

NITRO allows you to fetch bindings in bulk.

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "feopolicy_csvserver_binding": [ {

"boundto":<String_value>,

"name":<String_value>,

"priority":<Double_value>,

"gotopriorityexpression":<String_value>,

"activepolicy":<Integer_value>

}]}
```

count

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/feopolicy_csvserver_binding/name_value<String>?**count=yes**

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{"feopolicy_csvserver_binding": [ { "__count": "#no"} ] }
```

feopolicy_feoglobal_binding

Binding object showing the feoglobal that can be bound to feopolicy.

Properties

(click to see Operations)

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
boundto	<String>	Read-write	Location where the policy is bound to.
name	<String>	Read-write	The name of the front end optimization policy. Minimum length = 1
priority	<Double>	Read-only	Priority of the policy.
gotopriorityexpression	<String>	Read-only	Expression specifying the priority of the next policy

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
			which will get evaluated if the current policy rule evaluates to TRUE.
activepolicy	<Integer>	Read-only	Indicates whether a policy is bound or not.
__count	<Double>	Read-only	count parameter

Operations

(click to see Properties)

GET| GET (ALL)| COUNT

Some options that you can use for each operations:

- **Getting warnings in response:**NITRO allows you to get warnings in an operation by specifying the "warning" query parameter as "yes". For example, to get warnings while connecting to the Citrix ADC appliance, the URL is as follows:

  http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/login?warning=yes

  If any, the warnings are displayed in the response payload with the HTTP code "209 X-NITRO-WARNING".

- **Authenticated access for individual NITRO operations:**NITRO allows you to logon to the Citrix ADC appliance to perform individual operations. You can use this option instead of creating a NITRO session (using the login object) and then using that session to perform all operations,

  To do this, you must specify the username and password in the request header of the NITRO request as follows:

  X-NITRO-USER:<username>

  X-NITRO-PASS:<password>

  Note: In such cases, make sure that the request header DOES not include the following:

  Cookie:NITRO_AUTH_TOKEN=<tokenvalue>

`*Note:`*

Mandatory parameters are marked in redand placeholder content is marked in <green>.

get

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/feopolicy_feoglobal_binding/name_value<String>

Query-parameters:

filter

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/feopolicy_feoglobal_binding/name_value<String>?**filter=property-name1:property-value1,property-name2:property-value2**

Use this query-parameter to get the filtered set of feopolicy_feoglobal_binding resources configured on Citrix ADC. Filtering can be done on any of the properties of the resource.

pagination

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/feopolicy_feoglobal_binding/name_value<String>?**pagesize=#no;pageno=#no**

Use this query-parameter to get the feopolicy_feoglobal_binding resources in chunks.

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "feopolicy_feoglobal_binding": [ {
```

```
"boundto":<String_value>,

"name":<String_value>,

"priority":<Double_value>,

"gotopriorityexpression":<String_value>,

"activepolicy":<Integer_value>

}]}
```

get (all)

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/feopolicy_feoglobal_binding

Query-parameters:

bulkbindings

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/feopolicy_feoglobal_binding?**bulkbindings=yes**

NITRO allows you to fetch bindings in bulk.

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "feopolicy_feoglobal_binding": [ {

"boundto":<String_value>,

"name":<String_value>,

"priority":<Double_value>,

"gotopriorityexpression":<String_value>,

"activepolicy":<Integer_value>

}]}
```

count

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/feopolicy_feoglobal_binding/name_value<String>?**count=yes**

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{"feopolicy_feoglobal_binding": [ { "__count": "#no"} ] }
```

feopolicy_lbvserver_binding

Binding object showing the lbvserver that can be bound to feopolicy.

Properties

(click to see Operations)

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
boundto	<String>	Read-write	Location where the policy is bound to.
name	<String>	Read-write	The name of the front end optimization policy. Minimum length = 1
priority	<Double>	Read-only	Priority of the policy.
gotopriorityexpression	<String>	Read-only	Expression specifying the priority of the next policy which will get evaluated if the current policy rule evaluates to TRUE.
activepolicy	<Integer>	Read-only	Indicates whether a policy is bound or not.
__count	<Double>	Read-only	count parameter

Operations

(click to see Properties)

GET| GET (ALL)| COUNT

Some options that you can use for each operations:

- **Getting warnings in response:**NITRO allows you to get warnings in an operation by specifying the "warning" query parameter as "yes". For example, to get warnings while connecting to the Citrix ADC appliance, the URL is as follows:

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/login?warning=yes

If any, the warnings are displayed in the response payload with the HTTP code "209 X-NITRO-WARNING".

- **Authenticated access for individual NITRO operations:**NITRO allows you to logon to the Citrix ADC appliance to perform individual operations. You can use this option instead of creating a NITRO session (using the login object) and then using that session to perform all operations,

  To do this, you must specify the username and password in the request header of the NITRO request as follows:

  X-NITRO-USER:<username>

  X-NITRO-PASS:<password>

  Note: In such cases, make sure that the request header DOES not include the following:

  Cookie:NITRO_AUTH_TOKEN=<tokenvalue>

`*Note:`*

Mandatory parameters are marked in redand placeholder content is marked in <green>.

get

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/feopolicy_lbvserver_binding/name_value<String>

Query-parameters:

filter

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/feopolicy_lbvserver_binding/name_value<String>?**filter=property-name1:property-value1,property-name2:property-value2**

Use this query-parameter to get the filtered set of feopolicy_lbvserver_binding resources configured on Citrix ADC. Filtering can be done on any of the properties of the resource.

pagination

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/feopolicy_lbvserver_binding/name_value<String>?**pagesize=#no;pageno=#no**

Use this query-parameter to get the feopolicy_lbvserver_binding resources in chunks.

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "feopolicy_lbvserver_binding": [ {

"boundto":<String_value>,

"name":<String_value>,

"priority":<Double_value>,

"gotopriorityexpression":<String_value>,

"activepolicy":<Integer_value>

}]}
```

get (all)

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/feopolicy_lbvserver_binding

Query-parameters:

bulkbindings

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/feopolicy_lbvserver_binding?**bulkbindings=yes**

NITRO allows you to fetch bindings in bulk.

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "feopolicy_lbvserver_binding": [ {

"boundto":<String_value>,

"name":<String_value>,

"priority":<Double_value>,

"gotopriorityexpression":<String_value>,

"activepolicy":<Integer_value>

}]}
```

count

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/feopolicy_lbvserver_binding/name_value<String>?**count=yes**

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{"feopolicy_lbvserver_binding": [ { "__count": "#no"} ] }
```

Global Server Load Balancing

Global Server Load Balancing

Global Server Load Balancing (GSLB) configuration. GSLB feature ensures that client requests are directed to a best performing site available in a global enterprise and distributed Internet environment.

NAME	DESCRIPTION
gslbconfig	Configuration for gslb config
gslbdomain	Configuration for GSLB domain
gslbdomain_binding	Binding object showing the resources that can be bound to gslbdomain
gslbdomain_gslbservice_binding	Binding object showing the gslbservice that can be bound to gslbdomain
gslbdomain_gslbservicegroup_binding	Binding object showing the gslbservicegroup that can be bound to gslbdomain

NAME	DESCRIPTION
gslbdomain_gslbservicegroupmember_binding	Binding object showing the gslbservicegroupmember that can be bound to gslbdomain
gslbdomain_gslbvserver_binding	Binding object showing the gslbvserver that can be bound to gslbdomain
gslbdomain_lbmonitor_binding	Binding object showing the lbmonitor that can be bound to gslbdomain
gslbldnsentries	Configuration for LDNS entry
gslbldnsentry	Configuration for LDNS entry
gslbparameter	Configuration for GSLB parameter
gslbrunningconfig	Configuration for running GSLB configuration
gslbservice	Configuration for GSLB service
gslbservice_binding	Binding object showing the resources that can be bound to gslbservice
gslbservice_dnsview_binding	Binding object showing the dnsview that can be bound to gslbservice
gslbservice_lbmonitor_binding	Binding object showing the lbmonitor that can be bound to gslbservice
gslbservicegroup	Configuration for GSLB service group
gslbservicegroup_binding	Binding object showing the resources that can be bound to gslbservicegroup
gslbservicegroup_gslbservicegroupmember_binding	Binding object showing the gslbservicegroupmember that can be bound to gslbservicegroup
gslbservicegroup_lbmonitor_binding	Binding object showing the lbmonitor that can be bound to gslbservicegroup

NAME	DESCRIPTION
gslbservicegroup_servicegroupentitymonbindings_binding	Binding object showing the servicegroupentitymonbindings that can be bound to gslbservicegroup
gslbsite	Configuration for GSLB site
gslbsite_binding	Binding object showing the resources that can be bound to gslbsite
gslbsite_gslbservice_binding	Binding object showing the gslbservice that can be bound to gslbsite
gslbsite_gslbservicegroup_binding	Binding object showing the gslbservicegroup that can be bound to gslbsite
gslbsite_gslbservicegroupmember_binding	Binding object showing the gslbservicegroupmember that can be bound to gslbsite
gslbsyncstatus	Configuration for sync status
gslbvserver	Configuration for Global Server Load Balancing Virtual Server
gslbvserver_binding	Binding object showing the resources that can be bound to gslbvserver
gslbvserver_domain_binding	Binding object showing the domain that can be bound to gslbvserver
gslbvserver_gslbservice_binding	Binding object showing the gslbservice that can be bound to gslbvserver
gslbvserver_gslbservicegroup_binding	Binding object showing the gslbservicegroup that can be bound to gslbvserver
gslbvserver_gslbservicegroupmember_binding	Binding object showing the gslbservicegroupmember that can be bound to gslbvserver
gslbvserver_spilloverpolicy_binding	Binding object showing the spilloverpolicy that can be bound to gslbvserver

gslbconfig

Configuration for gslb config resource.

Properties

(click to see Operations)

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
preview	<Boolean>	Read-write	Do not synchronize the GSLB sites, but display the commands that would be applied on the slave node upon synchronization. Mutually exclusive with the Save Configuration option.
debug	<Boolean>	Read-write	Generate verbose output when synchronizing the GSLB sites. The Debug option generates more verbose output than the sync gslb config command in which the option is not used, and is useful for analyzing synchronization issues.
forcesync	<String>	Read-write	Force synchronization of the specified site even if a dependent configuration on the remote site is preventing synchronization or if

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
			one or more GSLB entities on the remote site have the same name but are of a different type. You can specify either the name of the remote site that you want to synchronize with the local site, or you can specify All Sites in the configuration utility (the string all-sites in the CLI). If you specify All Sites, all the sites in the GSLB setup are synchronized with the site on the master node. Note: If you select the Force Sync option, the synchronization starts without displaying the commands that are going to be executed.
nowarn	<Boolean>	Read-write	Suppress the warning and the confirmation prompt that are displayed before site synchronization begins. This option can be used in automation scripts

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
			that must not be interrupted by a prompt.
saveconfig	<Boolean>	Read-write	Save the configuration on all the nodes participating in the synchronization process, automatically. The master saves its configuration immediately before synchronization begins. Slave nodes save their configurations after the process of synchronization is complete. A slave node saves its configuration only if the configuration difference was successfully applied to it. Mutually exclusive with the Preview option.
command	<String>	Read-write	Run the specified command on the master node and then on all the slave nodes. You cannot use this option with the force sync and preview options.

Operations

(click to see Properties)

SYNC

Some options that you can use for each operations:

- **Getting warnings in response:**NITRO allows you to get warnings in an operation by specifying the "warning" query parameter as "yes". For example, to get warnings while connecting to the Citrix ADC appliance, the URL is as follows:

  http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/login?warning=yes

  If any, the warnings are displayed in the response payload with the HTTP code "209 X-NITRO-WARNING".

- **Authenticated access for individual NITRO operations:**NITRO allows you to logon to the Citrix ADC appliance to perform individual operations. You can use this option instead of creating a NITRO session (using the login object) and then using that session to perform all operations,

  To do this, you must specify the username and password in the request header of the NITRO request as follows:

  X-NITRO-USER:<username>

  X-NITRO-PASS:<password>

  Note: In such cases, make sure that the request header DOES not include the following:

  Cookie:NITRO_AUTH_TOKEN=<tokenvalue>

`*Note:`*

Mandatory parameters are marked in redand placeholder content is marked in <green>.

sync

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/gslbconfig?**action=sync**

HTTP Method:POST

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Content-Type:application/json

Request Payload:

```
{"gslbconfig":{

"preview":<Boolean_value>,

"debug":<Boolean_value>,

"forcesync":<String_value>,

"nowarn":<Boolean_value>,

"saveconfig":<Boolean_value>,

"command":<String_value>

}}
```

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error

gslbdomain

Configuration for GSLB domain resource.

Properties

(click to see Operations)

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
name	<String>	Read-write	Name of the Domain. Minimum length = 1
__count	<Double>	Read-only	count parameter

Operations

(click to see Properties)

GET (ALL)| GET| COUNT

Some options that you can use for each operations:

- **Getting warnings in response:**NITRO allows you to get warnings in an operation by specifying the "warning" query parameter as "yes". For example, to get warnings while connecting to the Citrix ADC appliance, the URL is as follows:

  http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/login?warning=yes

  If any, the warnings are displayed in the response payload with the HTTP code "209 X-NITRO-WARNING".

- **Authenticated access for individual NITRO operations:**NITRO allows you to logon to the Citrix ADC appliance to perform individual operations. You can use this option instead of creating a NITRO session (using the login object) and then using that session to perform all operations,

  To do this, you must specify the username and password in the request header of the NITRO request as follows:

  X-NITRO-USER:<username>

  X-NITRO-PASS:<password>

  Note: In such cases, make sure that the request header DOES not include the following:

  Cookie:NITRO_AUTH_TOKEN=<tokenvalue>

`*Note:`*

Mandatory parameters are marked in redand placeholder content is marked in <green>.

get (all)

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/gslbdomain

Query-parameters:

attrs

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/gslbdomain?**attrs=property-name1,property-name2**

Use this query parameter to specify the resource details that you want to retrieve.

filter

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/gslbdomain?**filter=property-name1:property-val1,property-name2:property-val2**

Use this query-parameter to get the filtered set of gslbdomain resources configured on Citrix ADC. Filtering can be done on any of the properties of the resource.

view

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/gslbdomain?**view=summary**

Note: By default, the retrieved results are displayed in detail view (?view=detail).

pagination

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/gslbdomain?**pagesize=#no;pageno=#no**

Use this query-parameter to get the gslbdomain resources in chunks.

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "gslbdomain": [ {
```

```
"name":<String_value>,



}]}
```

get

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/gslbdomain/name_value<String>

Query-parameters:

attrs

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/gslbdomain/name_value<String>?**attrs=property-name1,property-name2**

Use this query parameter to specify the resource details that you want to retrieve.

view

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/gslbdomain/name_value<String>?**view=summary**

Note: By default, the retrieved results are displayed in detail view (?view=detail).

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "gslbdomain": [ {

"name":<String_value>,



}]}
```

count

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/gslbdomain?**count=yes**

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "gslbdomain": [ { "__count": "#no"} ] }
```

gslbdomain_binding

Binding object which returns the resources bound to gslbdomain.

Properties

(click to see Operations)

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
name	<String>	Read-write	Name of the Domain. Minimum length = 1
gslbdomain_gslbservicegroup_binding	<gslbdomain_gslbservicegroup_binding[]>	Read-only	gslbservicegroup that can be bound to gslbdomain.
gslbdomain_gslbservice_binding	<gslbdomain_gslbservice_binding[]>	Read-only	gslbservice that can be bound to gslbdomain.
gslbdomain_lbmonitor_binding	<gslbdomain_lbmonitor_binding[]>	Read-only	lbmonitor that can be bound to gslbdomain.
gslbdomain_gslbservicegroupmember_binding	<gslbdomain_gslbservicegroupmember_binding[]>	Read-only	gslbservicegroupmember that can be bound to gslbdomain.
gslbdomain_gslbvserver_binding	<gslbdomain_gslbvserver_binding[]>	Read-only	gslbvserver that can be bound to gslbdomain.

Operations

(click to see Properties)

GET| GET (ALL)

Some options that you can use for each operations:

- **Getting warnings in response:**NITRO allows you to get warnings in an operation by specifying the "warning" query parameter as "yes". For example, to get warnings while connecting to the Citrix ADC appliance, the URL is as follows:

  http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/login?warning=yes

  If any, the warnings are displayed in the response payload with the HTTP code "209 X-NITRO-WARNING".

- **Authenticated access for individual NITRO operations:**NITRO allows you to logon to the Citrix ADC appliance to perform individual operations. You can use this option instead of creating a NITRO session (using the login object) and then using that session to perform all operations,

  To do this, you must specify the username and password in the request header of the NITRO request as follows:

  X-NITRO-USER:<username>

  X-NITRO-PASS:<password>

  Note: In such cases, make sure that the request header DOES not include the following:

  Cookie:NITRO_AUTH_TOKEN=<tokenvalue>

`*Note:`*

Mandatory parameters are marked in redand placeholder content is marked in <green>.

get

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/gslbdomain_binding/name_value<String>

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "gslbdomain_binding": [ {
```

```
"name":<String_value>,

"gslbdomain_gslbservicegroup_binding":<gslbdomain_gslbservicegroup_binding
[]_value>,

"gslbdomain_gslbservicegroupmember_binding":<gslbdomain_gslbservicegroupme
mber_binding[]_value>,

"gslbdomain_lbmonitor_binding":<gslbdomain_lbmonitor_binding[]_value>,

"gslbdomain_gslbvserver_binding":<gslbdomain_gslbvserver_binding[]_value>,

"gslbdomain_gslbservice_binding":<gslbdomain_gslbservice_binding[]_value>

}]}
```

get (all)

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/gslbdomain_binding

Query-parameters:

bulkbindings

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/gslbdomain_binding?**bulkbindings=yes**

NITRO allows you to fetch bindings in bulk.

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "gslbdomain_binding": [ {

"name":<String_value>,

"gslbdomain_gslbservicegroup_binding":<gslbdomain_gslbservicegroup_binding
[]_value>,

"gslbdomain_gslbservicegroupmember_binding":<gslbdomain_gslbservicegroupme
mber_binding[]_value>,

"gslbdomain_lbmonitor_binding":<gslbdomain_lbmonitor_binding[]_value>,

"gslbdomain_gslbvserver_binding":<gslbdomain_gslbvserver_binding[]_value>,

"gslbdomain_gslbservice_binding":<gslbdomain_gslbservice_binding[]_value>

}]}
```

gslbdomain_gslbservice_binding

Binding object showing the gslbservice that can be bound to gslbdomain.

Properties

(click to see Operations)

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
name	<String>	Read-write	Name of the Domain. Minimum length = 1
servicename	<String>	Read-write	The service name.
cnameentry	<String>	Read-only	The cname of the gslb service.
gslbthreshold	<Integer>	Read-only	The threshold value of the service.
port	<Integer>	Read-only	Port Number.

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
			Minimum value = 1 Range 1 - 65535 * in CLI is represented as 65535 in NITRO API
state	<String>	Read-only	The state of the vserver. Possible values = UP, DOWN, UNKNOWN, BUSY, OUT OF SERVICE, GOING OUT OF SERVICE, DOWN WHEN GOING OUT OF SERVICE, NS_EMPTY_STR, Unknown, DISABLED
vservername	<String>	Read-only	.
svreffgslbstate	<String>	Read-only	GSLB server state. Possible values = UP, DOWN, UNKNOWN, BUSY, OUT OF SERVICE, GOING OUT OF SERVICE, DOWN WHEN GOING OUT OF SERVICE, NS_EMPTY_STR, Unknown, DISABLED

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
dynamicconfwt	<Double>	Read-only	dynamic weight.
weight	<Double>	Read-only	weight assigned. Minimum value = 1 Maximum value = 100
ipaddress	<String>	Read-only	The Ip address of the service.
servicetype	<String>	Read-only	The type GSLB service. Possible values = HTTP, FTP, TCP, UDP, SSL, SSL_BRIDGE, SSL_TCP, NNTP, ANY, SIP_UDP, SIP_TCP, SIP_SSL, RADIUS, RDP, RTSP, MYSQL, MSSQL, ORACLE
cumulativeweight	<Double>	Read-only	cumlative weight.
__count	<Double>	Read-only	count parameter

Operations

(click to see Properties)

GET| GET (ALL)| COUNT

Some options that you can use for each operations:

- **Getting warnings in response:**NITRO allows you to get warnings in an operation by specifying the "warning" query parameter as "yes". For example, to get warnings while connecting to the Citrix ADC appliance, the URL is as follows:

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/login?warning=yes

If any, the warnings are displayed in the response payload with the HTTP code "209 X-NITRO-WARNING".

- **Authenticated access for individual NITRO operations:**NITRO allows you to logon to the Citrix ADC appliance to perform individual operations. You can use this option instead of creating a NITRO session (using the login object) and then using that session to perform all operations,

  To do this, you must specify the username and password in the request header of the NITRO request as follows:

  X-NITRO-USER:<username>

  X-NITRO-PASS:<password>

  Note: In such cases, make sure that the request header DOES not include the following:

  Cookie:NITRO_AUTH_TOKEN=<tokenvalue>

`*Note:`*

Mandatory parameters are marked in redand placeholder content is marked in <green>.

get

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/gslbdomain_gslbservice_binding/name_value<String>

Query-parameters:

filter

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/gslbdomain_gslbservice_binding/name_value<String>?**filter=property-name1:property-value1,property-name2:property-value2**

Use this query-parameter to get the filtered set of gslbdomain_gslbservice_binding resources configured on Citrix ADC. Filtering can be done on any of the properties of the resource.

pagination

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/gslbdomain_gslbservice_binding/name_value<String>?**pagesize=#no;pageno=#no**

Use this query-parameter to get the gslbdomain_gslbservice_binding resources in chunks.

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "gslbdomain_gslbservice_binding": [ {

"name":<String_value>,

"servicename":<String_value>,

"cnameentry":<String_value>,

"gslbthreshold":<Integer_value>,

"port":<Integer_value>,

"state":<String_value>,

"vservername":<String_value>,

"svreffgslbstate":<String_value>,

"dynamicconfwt":<Double_value>,

"weight":<Double_value>,

"ipaddress":<String_value>,

"servicetype":<String_value>,
```

```
"cumulativeweight":<Double_value>

}]}
```

get (all)

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/gslbdomain_gslbservice_binding

Query-parameters:

bulkbindings

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/gslbdomain_gslbservice_binding?**bulkbindings=yes**

NITRO allows you to fetch bindings in bulk.

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "gslbdomain_gslbservice_binding": [ {

"name":<String_value>,

"servicename":<String_value>,

"cnameentry":<String_value>,
```

```
"gslbthreshold":<Integer_value>,

"port":<Integer_value>,

"state":<String_value>,

"vservername":<String_value>,

"svreffgslbstate":<String_value>,

"dynamicconfwt":<Double_value>,

"weight":<Double_value>,

"ipaddress":<String_value>,

"servicetype":<String_value>,

"cumulativeweight":<Double_value>

}]}
```

count

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/gslbdomain_gslbservice_binding/name_value<String>?**count=yes**

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{"gslbdomain_gslbservice_binding": [ { "__count": "#no"} ] }
```

gslbdomain_gslbvserver_binding

Binding object showing the gslbvserver that can be bound to gslbdomain.

Properties

(click to see Operations)

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
name	<String>	Read-write	Name of the Domain. Minimum length = 1
vservername	<String>	Read-write	.
sitename	<String>	Read-only	Name of the site to which the service belongs.
backuplbmethod	<String>	Read-only	Indicates the backup method in case the primary fails. Possible values = ROUNDROBIN, LEASTCONNECTION, LEASTRESPONSETIME, SOURCEIPHASH, LEASTBANDWIDTH, LEASTPACKETS, STATICPROXIMITY, RTT, CUSTOMLOAD

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
customheaders	<String>	Read-only	The string that is sent to the service. Applicable to HTTP ,HTTP-ECV and RTSP monitor types.
persistenceid	<Double>	Read-only	Persistence id of the gslb vserver.
state	<String>	Read-only	The state of the vserver. Possible values = UP, DOWN, UNKNOWN, BUSY, OUT OF SERVICE, GOING OUT OF SERVICE, DOWN WHEN GOING OUT OF SERVICE, NS_EMPTY_STR, Unknown, DISABLED
edr	<String>	Read-only	Send clients an empty DNS response when the GSLB virtual server is DOWN. Default value: DISABLED Possible values = ENABLED, DISABLED
lbmethod	<String>	Read-only	The load balancing method set for the virtual server.

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
			Possible values = ROUNDROBIN, LEASTCONNECTION, LEASTRESPONSETIME, SOURCEIPHASH, LEASTBANDWIDTH, LEASTPACKETS, STATICPROXIMITY, RTT, CUSTOMLOAD
cip	<String>	Read-only	Indicates if Client IP option is enabled. Possible values = ENABLED, DISABLED
statechangetimesec	<String>	Read-only	Time since last state change.
netmask	<String>	Read-only	Netmask. Minimum length = 1
dnsrecordtype	<String>	Read-only	The IP type for this GSLB vserver. Possible values = A, AAAA, CNAME, NAPTR
persistmask	<String>	Read-only	The optional IPv4 network mask applied to IPv4 addresses to establish source IP

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
			address based persistence. Minimum length = 1
sitepersistence	<String>	Read-only	Indicates the type of cookie persistence set. Possible values = ConnectionProxy, HTTPRedirect, NONE
servicetype	<String>	Read-only	The type GSLB service. Possible values = HTTP, FTP, TCP, UDP, SSL, SSL_BRIDGE, SSL_TCP, NNTP, ANY, SIP_UDP, SIP_TCP, SIP_SSL, RADIUS, RDP, RTSP, MYSQL, MSSQL, ORACLE
v6persistmasklen	<Double>	Read-only	Number of bits to consider in an IPv6 source IP address when creating source IP address based persistence sessions. Default value: 128 Minimum value = 1 Maximum value = 128

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
v6netmasklen	<Double>	Read-only	Number of bits to consider, in an IPv6 source IP address, for creating the hash that is required by the SOURCEIPHASH load balancing method. Default value: 128 Minimum value = 1 Maximum value = 128
persistencetype	<String>	Read-only	Indicates if persistence is set on the gslb vserver. Possible values = SOURCEIP, NONE
siteprefix	<String>	Read-only	The site prefix string.
dynamicweight	<String>	Read-only	Dynamic weight method of the vserver. Default value: DISABLED Possible values = SERVICECOUNT, SERVICEWEIGHT, DISABLED
mir	<String>	Read-only	. Default value: DISABLED

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
			Possible values = ENABLED, DISABLED
__count	<Double>	Read-only	count parameter

Operations

(click to see Properties)

GET| GET (ALL)| COUNT

Some options that you can use for each operations:

- **Getting warnings in response:**NITRO allows you to get warnings in an operation by specifying the "warning" query parameter as "yes". For example, to get warnings while connecting to the Citrix ADC appliance, the URL is as follows:

  http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/login?warning=yes

  If any, the warnings are displayed in the response payload with the HTTP code "209 X-NITRO-WARNING".

- **Authenticated access for individual NITRO operations:**NITRO allows you to logon to the Citrix ADC appliance to perform individual operations. You can use this option instead of creating a NITRO session (using the login object) and then using that session to perform all operations,

  To do this, you must specify the username and password in the request header of the NITRO request as follows:

  X-NITRO-USER:<username>

  X-NITRO-PASS:<password>

  Note: In such cases, make sure that the request header DOES not include the following:

  Cookie:NITRO_AUTH_TOKEN=<tokenvalue>

`*Note:`*

Mandatory parameters are marked in redand placeholder content is marked in <green>.

get

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/gslbdomain_gslbvserver_binding/name_value<String>

Query-parameters:

filter

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/gslbdomain_gslbvserver_binding/name_value<String>?**filter=property-name1:property-value1,property-name2:property-value2**

Use this query-parameter to get the filtered set of gslbdomain_gslbvserver_binding resources configured on Citrix ADC. Filtering can be done on any of the properties of the resource.

pagination

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/gslbdomain_gslbvserver_binding/name_value<String>?**pagesize=#no;pageno=#no**

Use this query-parameter to get the gslbdomain_gslbvserver_binding resources in chunks.

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "gslbdomain_gslbvserver_binding": [ {

"name":<String_value>,

"vservername":<String_value>,
```

```
"sitename":<String_value>,

"backuplbmethod":<String_value>,

"customheaders":<String_value>,

"persistenceid":<Double_value>,

"state":<String_value>,

"edr":<String_value>,

"lbmethod":<String_value>,

"cip":<String_value>,

"statechangetimesec":<String_value>,

"netmask":<String_value>,

"dnsrecordtype":<String_value>,

"persistmask":<String_value>,

"sitepersistence":<String_value>,

"servicetype":<String_value>,

"v6persistmasklen":<Double_value>,

"v6netmasklen":<Double_value>,

"persistencetype":<String_value>,

"siteprefix":<String_value>,

"dynamicweight":<String_value>,

"mir":<String_value>

}]}
```

get (all)

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/gslbdomain_gslbvserver_binding

Query-parameters:

bulkbindings

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/gslbdomain_gslbvserver_binding?**bulkbindings=yes**

NITRO allows you to fetch bindings in bulk.

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "gslbdomain_gslbvserver_binding": [ {

"name":<String_value>,

"vservername":<String_value>,

"sitename":<String_value>,

"backuplbmethod":<String_value>,

"customheaders":<String_value>,
```

```
"persistenceid":<Double_value>,

"state":<String_value>,

"edr":<String_value>,

"lbmethod":<String_value>,

"cip":<String_value>,

"statechangetimesec":<String_value>,

"netmask":<String_value>,

"dnsrecordtype":<String_value>,

"persistmask":<String_value>,

"sitepersistence":<String_value>,

"servicetype":<String_value>,

"v6persistmasklen":<Double_value>,

"v6netmasklen":<Double_value>,

"persistencetype":<String_value>,

"siteprefix":<String_value>,

"dynamicweight":<String_value>,

"mir":<String_value>

}]}
```

count

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/gslbdomain_gslbvserver_binding/name_value<String>?**count=yes**

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{"gslbdomain_gslbvserver_binding": [ { "__count": "#no"} ] }
```

gslbdomain_lbmonitor_binding

Binding object showing the lbmonitor that can be bound to gslbdomain.

Properties

(click to see Operations)

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
name	<String>	Read-write	Name of the Domain. Minimum length = 1
monitorname	<String>	Read-write	Monitor name.
monstatcode	<Integer>	Read-only	The code indicating the monitor response.
lastresponse	<String>	Read-only	The string form of monstatcode.
customheaders	<String>	Read-only	The string that is sent to the service. Applicable to

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
			HTTP ,HTTP-ECV and RTSP monitor types.
iptunnel	<String>	Read-only	The state of the monitor for tunneled devices. Possible values = YES, NO
servicename	<String>	Read-only	The service name.
monitortotalprobes	<Double>	Read-only	Total monitor probes.
respcode	<String>	Read-only	The response codes.
vservername	<String>	Read-only	.
httprequest	<String>	Read-only	HTTP request to the backend server.
responsetime	<Double>	Read-only	Response time of this monitor.
monitortotalfailedprobes	<Double>	Read-only	Total probes failed.
monitorcurrentfailedprobes	<Double>	Read-only	Total number of current failed probes.
monstate	<String>	Read-only	Monitor state. Possible values = UP, DOWN, UNKNOWN, BUSY, OUT OF SERVICE, GOING OUT OF SERVICE, DOWN WHEN GOING OUT OF SERVICE, NS_EMPTY_STR,

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
			Unknown, DISABLED
__count	<Double>	Read-only	count parameter

Operations

(click to see Properties)

GET| GET (ALL)| COUNT

Some options that you can use for each operations:

- **Getting warnings in response:**NITRO allows you to get warnings in an operation by specifying the "warning" query parameter as "yes". For example, to get warnings while connecting to the Citrix ADC appliance, the URL is as follows:

  http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/login?warning=yes

  If any, the warnings are displayed in the response payload with the HTTP code "209 X-NITRO-WARNING".

- **Authenticated access for individual NITRO operations:**NITRO allows you to logon to the Citrix ADC appliance to perform individual operations. You can use this option instead of creating a NITRO session (using the login object) and then using that session to perform all operations,

  To do this, you must specify the username and password in the request header of the NITRO request as follows:

  X-NITRO-USER:<username>

  X-NITRO-PASS:<password>

  Note: In such cases, make sure that the request header DOES not include the following:

  Cookie:NITRO_AUTH_TOKEN=<tokenvalue>

`*Note:`*

Mandatory parameters are marked in redand placeholder content is marked in <green>.

get

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/gslbdomain_lbmonitor_binding/name_value<String>

Query-parameters:

filter

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/gslbdomain_lbmonitor_binding/name_value<String>?**filter=property-name1:property-value1,property-name2:property-value2**

Use this query-parameter to get the filtered set of gslbdomain_lbmonitor_binding resources configured on Citrix ADC. Filtering can be done on any of the properties of the resource.

pagination

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/gslbdomain_lbmonitor_binding/name_value<String>?**pagesize=#no;pageno=#no**

Use this query-parameter to get the gslbdomain_lbmonitor_binding resources in chunks.

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "gslbdomain_lbmonitor_binding": [ {

"name":<String_value>,

"monitorname":<String_value>,
```

```
"monstatcode":<Integer_value>,

"lastresponse":<String_value>,

"customheaders":<String_value>,

"iptunnel":<String_value>,

"servicename":<String_value>,

"monitortotalprobes":<Double_value>,

"respcode":<String_value>,

"vservername":<String_value>,

"httprequest":<String_value>,

"responsetime":<Double_value>,

"monitortotalfailedprobes":<Double_value>,

"monitorcurrentfailedprobes":<Double_value>,

"monstate":<String_value>

}]}
```

get (all)

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/gslbdomain_lbmonitor_binding

Query-parameters:

bulkbindings

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/gslbdomain_lbmonitor_binding?**bulkbindings=yes**

NITRO allows you to fetch bindings in bulk.

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "gslbdomain_lbmonitor_binding": [ {

"name":<String_value>,

"monitorname":<String_value>,

"monstatcode":<Integer_value>,

"lastresponse":<String_value>,

"customheaders":<String_value>,

"iptunnel":<String_value>,

"servicename":<String_value>,

"monitortotalprobes":<Double_value>,

"respcode":<String_value>,

"vservername":<String_value>,

"httprequest":<String_value>,

"responsetime":<Double_value>,

"monitortotalfailedprobes":<Double_value>,
```

```
"monitorcurrentfailedprobes":<Double_value>,

"monstate":<String_value>

}]}
```

count

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/gslbdomain_lbmonitor_binding/name_value<String>?**count=yes**

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{"gslbdomain_lbmonitor_binding": [ { "__count": "#no"} ] }
```

gslbldnsentries

Configuration for LDNS entry resource.

Properties

(click to see Operations)

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
nodeid	<Double>	Read-write	Unique number that identifies the cluster node. Minimum value = 0 Maximum value = 31
sitename	<String>	Read-only	The GSLB site name.
numsites	<Double>	Read-only	Specifies the number of gslb sites.
ipaddress	<String>	Read-only	IP address of the LDNS server.
ttl	<Double>	Read-only	TTL value of the LDNS entry.
name	<String>	Read-only	Monitor that is currently being used to monitor the LDNS ip..
rtt	<Double[]>	Read-only	RTT value of the LDNS entry for all gslb sites.
__count	<Double>	Read-only	count parameter

Operations

(click to see Properties)

CLEAR| GET (ALL)| COUNT

Some options that you can use for each operations:

- **Getting warnings in response:**NITRO allows you to get warnings in an operation by specifying the "warning" query parameter as "yes". For example, to get warnings while connecting to the Citrix ADC appliance, the URL is as follows:

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/login?warning=yes

If any, the warnings are displayed in the response payload with the HTTP code "209 X-NITRO-WARNING".

- **Authenticated access for individual NITRO operations:**NITRO allows you to logon to the Citrix ADC appliance to perform individual operations. You can use this option instead of creating a NITRO session (using the login object) and then using that session to perform all operations,

  To do this, you must specify the username and password in the request header of the NITRO request as follows:

  X-NITRO-USER:<username>

  X-NITRO-PASS:<password>

  Note: In such cases, make sure that the request header DOES not include the following:

  Cookie:NITRO_AUTH_TOKEN=<tokenvalue>

`*Note:`*

Mandatory parameters are marked in redand placeholder content is marked in <green>.

clear

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/gslbldnsentries?**action=clear**

HTTP Method:POST

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Content-Type:application/json

Request Payload:

```
{"gslbldnsentries":{

}}
```

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error

get (all)

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/gslbldnsentries

Query-parameters:

args

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/gslbldnsentries?**args=nodeid:<Double_value>**

Use this query-parameter to get gslbldnsentries resources based on additional properties.

attrs

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/gslbldnsentries?**attrs=property-name1,property-name2**

Use this query parameter to specify the resource details that you want to retrieve.

filter

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/gslbldnsentries?**filter=property-name1:property-val1,property-name2:property-val2**

Use this query-parameter to get the filtered set of gslbldnsentries resources configured on Citrix ADC. Filtering can be done on any of the properties of the resource.

view

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/gslbldnsentries?**view=summary**

Note: By default, the retrieved results are displayed in detail view (?view=detail).

pagination

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/gslbldnsentries?**pagesize=#no;pageno=#no**

Use this query-parameter to get the gslbldnsentries resources in chunks.

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "gslbldnsentries": [ {

nodeid:<Double_value>"sitename":<String_value>,

"numsites":<Double_value>,

"ipaddress":<String_value>,

"ttl":<Double_value>,

"name":<String_value>,

"rtt":<Double[]_value>

}]}
```

count

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/gslbldnsentries?**count=yes**

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "gslbldnsentries": [ { "__count": "#no"} ] }
```

gslbldnsentry

Configuration for LDNS entry resource.

Properties

(click to see Operations)

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
ipaddress	<String>	Read-write	IP address of the LDNS server. Minimum length = 1

Operations

(click to see Properties)

DELETE

Some options that you can use for each operations:

- **Getting warnings in response:**NITRO allows you to get warnings in an operation by specifying the "warning" query parameter as "yes". For example, to get warnings while connecting to the Citrix ADC appliance, the URL is as follows:

  http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/login?warning=yes

  If any, the warnings are displayed in the response payload with the HTTP code "209 X-NITRO-WARNING".

- **Authenticated access for individual NITRO operations:**NITRO allows you to logon to the Citrix ADC appliance to perform individual operations. You can use this option instead of creating a NITRO session (using the login object) and then using that session to perform all operations,

  To do this, you must specify the username and password in the request header of the NITRO request as follows:

  X-NITRO-USER:<username>

  X-NITRO-PASS:<password>

  Note: In such cases, make sure that the request header DOES not include the following:

  Cookie:NITRO_AUTH_TOKEN=<tokenvalue>

`*Note:`*

Mandatory parameters are marked in redand placeholder content is marked in <green>.

delete

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/gslbldnsentry

HTTP Method:DELETE

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue>

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error

gslbparameter

Configuration for GSLB parameter resource.

Properties

(click to see Operations)

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
ldnsentrytimeout	<Double>	Read-write	Time, in seconds, after which an inactive LDNS entry is removed. Default value: 180 Minimum value = 30 Maximum value = 65534
rtttolerance	<Double>	Read-write	Tolerance, in milliseconds, for newly learned round-trip time (RTT) values. If the difference between the old RTT value and the newly computed RTT value is less than or equal to the specified tolerance value, the LDNS entry in the network metric table is not updated with the new RTT value. Prevents the exchange of metrics when variations in RTT values are negligible. Default value: 5 Minimum value = 1 Maximum value = 100

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
ldnsmask	<String>	Read-write	The IPv4 network mask with which to create LDNS entries. Minimum length = 1
v6ldnsmasklen	<Double>	Read-write	Mask for creating LDNS entries for IPv6 source addresses. The mask is defined as the number of leading bits to consider, in the source IP address, when creating an LDNS entry. Default value: 128 Minimum value = 1 Maximum value = 128
ldnsprobeorder	<String[]>	Read-write	Order in which monitors should be initiated to calculate RTT. Possible values = PING, DNS, TCP
dropldnsreq	<String>	Read-write	Drop LDNS requests if round-trip time (RTT) information is not available. Default value: DISABLED Possible values = ENABLED, DISABLED

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
gslbsvcstatedelaytime	<Double>	Read-write	Amount of delay in updating the state of GSLB service to DOWN when MEP goes down. This parameter is applicable only if monitors are not bound to GSLB services. Default value: 0 Minimum value = 0 Maximum value = 3600
automaticconfigsync	<String>	Read-write	GSLB configuration will be synced automatically to remote gslb sites if enabled. Default value: DISABLED Possible values = ENABLED, DISABLED
flags	<Double>	Read-only	State of the GSLB parameter.
builtin	<String[]>	Read-only	. Possible values = MODIFIABLE, DELETABLE, IMMUTABLE, PARTITION_ALL

Operations

(click to see Properties)

UPDATE| UNSET| GET (ALL)

Some options that you can use for each operations:

- **Getting warnings in response:**NITRO allows you to get warnings in an operation by specifying the "warning" query parameter as "yes". For example, to get warnings while connecting to the Citrix ADC appliance, the URL is as follows:

  http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/login?warning=yes

  If any, the warnings are displayed in the response payload with the HTTP code "209 X-NITRO-WARNING".

- **Authenticated access for individual NITRO operations:**NITRO allows you to logon to the Citrix ADC appliance to perform individual operations. You can use this option instead of creating a NITRO session (using the login object) and then using that session to perform all operations,

  To do this, you must specify the username and password in the request header of the NITRO request as follows:

  X-NITRO-USER:<username>

  X-NITRO-PASS:<password>

  Note: In such cases, make sure that the request header DOES not include the following:

  Cookie:NITRO_AUTH_TOKEN=<tokenvalue>

`*Note:`*

Mandatory parameters are marked in redand placeholder content is marked in <green>.

update

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/gslbparameter

HTTP Method:PUT

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Content-Type:application/json

Request Payload:

```
{"gslbparameter":{

"ldnsentrytimeout":<Double_value>,

"rtttolerance":<Double_value>,

"ldnsmask":<String_value>,

"v6ldnsmasklen":<Double_value>,

"ldnsprobeorder":<String[]_value>,

"dropldnsreq":<String_value>,

"gslbsvcstatedelaytime":<Double_value>,

"automaticconfigsync":<String_value>

}}
```

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error

unset

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/gslbparameter?**action=unset**

HTTP Method:POST

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Content-Type:application/json

Request Payload:

```
{"gslbparameter":{

"ldnsentrytimeout":true,

"rtttolerance":true,

"ldnsmask":true,

"v6ldnsmasklen":true,

"ldnsprobeorder":true,

"dropldnsreq":true,

"gslbsvcstatedelaytime":true,

"automaticconfigsync":true

}}
```

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error

get (all)

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/gslbparameter

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "gslbparameter": [ {

"flags":<Double_value>,

"ldnsentrytimeout":<Double_value>,

"rtttolerance":<Double_value>,

"ldnsmask":<String_value>,

"v6ldnsmasklen":<Double_value>,

"ldnsprobeorder":<String[]_value>,

"dropldnsreq":<String_value>,

"gslbsvcstatedelaytime":<Double_value>,

"automaticconfigsync":<String_value>,

"builtin":<String[]_value>

}]}
```

gslbrunningconfig

Configuration for running GSLB configuration resource.

Properties

(click to see Operations)

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
response	<String>	Read-only	gslb sync status as text blob.

Operations

(click to see Properties)

GET (ALL)

Some options that you can use for each operations:

- **Getting warnings in response:**NITRO allows you to get warnings in an operation by specifying the "warning" query parameter as "yes". For example, to get warnings while connecting to the Citrix ADC appliance, the URL is as follows:

  http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/login?warning=yes

  If any, the warnings are displayed in the response payload with the HTTP code "209 X-NITRO-WARNING".

- **Authenticated access for individual NITRO operations:**NITRO allows you to logon to the Citrix ADC appliance to perform individual operations. You can use this option instead of creating a NITRO session (using the login object) and then using that session to perform all operations,

  To do this, you must specify the username and password in the request header of the NITRO request as follows:

  X-NITRO-USER:<username>

  X-NITRO-PASS:<password>

  Note: In such cases, make sure that the request header DOES not include the following:

  Cookie:NITRO_AUTH_TOKEN=<tokenvalue>

`*Note:`*

Mandatory parameters are marked in redand placeholder content is marked in <green>.

get (all)

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/gslbrunningconfig

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "gslbrunningconfig": [ {

"response":<String_value>

}]}
```

gslbservice

Configuration for GSLB service resource.

Properties

(click to see Operations)

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
servicename	<String>	Read-write	Name for the GSLB service. Must begin with an ASCII alphanumeric or underscore (_) character, and must contain only ASCII alphanumeric, underscore, hash (#), period (.), space, colon (:), at (@), equals (=), and hyphen (-) characters. Can be

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
			changed after the GSLB service is created. CLI Users: If the name includes one or more spaces, enclose the name in double or single quotation marks (for example, "my gslbsvc" or 'my gslbsvc'). Minimum length = 1
cnameentry	<String>	Read-write	Canonical name of the GSLB service. Used in CNAME-based GSLB. Minimum length = 1
ip	<String>	Read-write	IP address for the GSLB service. Should represent a load balancing, content switching, or VPN virtual server on the Citrix ADC, or the IP address of another load balancing device. Minimum length = 1
servername	<String>	Read-write	Name of the server hosting the GSLB service. Minimum length = 1

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
servicetype	<String>	Read-write	Type of service to create. Default value: NSSVC_SERVICE_UNKNOWN Possible values = HTTP, FTP, TCP, UDP, SSL, SSL_BRIDGE, SSL_TCP, NNTP, ANY, SIP_UDP, SIP_TCP, SIP_SSL, RADIUS, RDP, RTSP, MYSQL, MSSQL, ORACLE
port	<Integer>	Read-write	Port on which the load balancing entity represented by this GSLB service listens. Minimum value = 1 Range 1 - 65535 * in CLI is represented as 65535 in NITRO API
publicip	<String>	Read-write	The public IP address that a NAT device translates to the GSLB service's private IP address. Optional.

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
publicport	<Integer>	Read-write	The public port associated with the GSLB service's public IP address. The port is mapped to the service's private port number. Applicable to the local GSLB service. Optional.
maxclient	<Double>	Read-write	The maximum number of open connections that the service can support at any given time. A GSLB service whose connection count reaches the maximum is not considered when a GSLB decision is made, until the connection count drops below the maximum. Minimum value = 0 Maximum value = 4294967294
healthmonitor	<String>	Read-write	Monitor the health of the GSLB service. Default value: YES Possible values = YES, NO

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
sitename	<String>	Read-write	Name of the GSLB site to which the service belongs. Minimum length = 1
state	<String>	Read-write	Enable or disable the service. Default value: ENABLED Possible values = ENABLED, DISABLED
cip	<String>	Read-write	In the request that is forwarded to the GSLB service, insert a header that stores the client's IP address. Client IP header insertion is used in connection-proxy based site persistence. Default value: DISABLED Possible values = ENABLED, DISABLED
cipheader	<String>	Read-write	Name for the HTTP header that stores the client's IP address. Used with the Client IP option. If client IP header insertion is enabled on the service and a name is not specified

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
			for the header, the Citrix ADC uses the name specified by the cipHeader parameter in the set ns param command or, in the GUI, the Client IP Header parameter in the Configure HTTP Parameters dialog box. Minimum length = 1
sitepersistence	<String>	Read-write	Use cookie-based site persistence. Applicable only to HTTP and SSL GSLB services. Possible values = ConnectionProxy, HTTPRedirect, NONE
cookietimeout	<Double>	Read-write	Timeout value, in minutes, for the cookie, when cookie based site persistence is enabled. Minimum value = 0 Maximum value = 1440
siteprefix	<String>	Read-write	The site's prefix string. When the service is bound to a GSLB virtual server, a GSLB site domain is generated

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
			internally for each bound service-domain pair by concatenating the site prefix of the service and the name of the domain. If the special string NONE is specified, the site-prefix string is unset. When implementing HTTP redirect site persistence, the Citrix ADC redirects GSLB requests to GSLB services by using their site domains.
clttimeout	<Double>	Read-write	Idle time, in seconds, after which a client connection is terminated. Applicable if connection proxy based site persistence is used. Minimum value = 0 Maximum value = 31536000
svrtimeout	<Double>	Read-write	Idle time, in seconds, after which a server connection is terminated. Applicable if connection proxy based site persistence is used.

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
			Minimum value = 0 Maximum value = 31536000
maxbandwidth	<Double>	Read-write	Integer specifying the maximum bandwidth allowed for the service. A GSLB service whose bandwidth reaches the maximum is not considered when a GSLB decision is made, until its bandwidth consumption drops below the maximum.
downstateflush	<String>	Read-write	Flush all active transactions associated with the GSLB service when its state transitions from UP to DOWN. Do not enable this option for services that must complete their transactions. Applicable if connection proxy based site persistence is used. Possible values = ENABLED, DISABLED
maxaaausers	<Double>	Read-write	Maximum number of SSL VPN users that can be logged on concurrently to the

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
			VPN virtual server that is represented by this GSLB service. A GSLB service whose user count reaches the maximum is not considered when a GSLB decision is made, until the count drops below the maximum. Minimum value = 0 Maximum value = 65535
monthreshold	<Double>	Read-write	Monitoring threshold value for the GSLB service. If the sum of the weights of the monitors that are bound to this GSLB service and are in the UP state is not equal to or greater than this threshold value, the service is marked as DOWN. Minimum value = 0 Maximum value = 65535
hashid	<Double>	Read-write	Unique hash identifier for the GSLB service, used by hash based load balancing methods. Minimum value = 1

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
comment	<String>	Read-write	Any comments that you might want to associate with the GSLB service.
appflowlog	<String>	Read-write	Enable logging appflow flow information. Default value: ENABLED Possible values = ENABLED, DISABLED
naptrreplacement	<String>	Read-write	The replacement domain name for this NAPTR. Maximum length = 255
naptrorder	<Double>	Read-write	An integer specifying the order in which the NAPTR records MUST be processed in order to accurately represent the ordered list of Rules. The ordering is from lowest to highest. Default value: 1 Minimum value = 1 Maximum value = 65535
naptrservices	<String>	Read-write	Service Parameters applicable to this delegation path.

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
			Maximum length = 255
naptrdomainttl	<Double>	Read-write	Modify the TTL of the internally created naptr domain. Default value: 3600 Minimum value = 1
naptrpreference	<Double>	Read-write	An integer specifying the preference of this NAPTR among NAPTR records having same order. lower the number, higher the preference. Default value: 1 Minimum value = 1 Maximum value = 65535
ipaddress	<String>	Read-write	The new IP address of the service.
viewname	<String>	Read-write	Name of the DNS view of the service. A DNS view is used in global server load balancing (GSLB) to return a predetermined IP address to a specific group of clients, which are identified by using a DNS policy. Minimum length = 1

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
viewip	<String>	Read-write	IP address to be used for the given view.
weight	<Double>	Read-write	Weight to assign to the monitor-service binding. A larger number specifies a greater weight. Contributes to the monitoring threshold, which determines the state of the service. Minimum value = 1 Maximum value = 100
monitor_name_svc	<String>	Read-write	Name of the monitor to bind to the service. Minimum length = 1
newname	<String>	Read-write	New name for the GSLB service. Minimum length = 1
gslb	<String>	Read-only	. Default value: GSLB Possible values = REMOTE, LOCAL
svrstate	<String>	Read-only	Server state. Possible values = UP, DOWN, UNKNOWN, BUSY, OUT OF SERVICE, GOING OUT OF

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
			SERVICE, DOWN WHEN GOING OUT OF SERVICE, NS_EMPTY_STR, Unknown, DISABLED
svreffgslbstate	<String>	Read-only	Effective state of the gslb svc. Possible values = UP, DOWN, UNKNOWN, BUSY, OUT OF SERVICE, GOING OUT OF SERVICE, DOWN WHEN GOING OUT OF SERVICE, NS_EMPTY_STR, Unknown, DISABLED
gslbthreshold	<Integer>	Read-only	Indicates if gslb svc has reached threshold.
gslbsvcstats	<Integer>	Read-only	Indicates if gslb svc has stats of the primary or the whole chain.
monstate	<String>	Read-only	State of the monitor bound to gslb service. Possible values = ENABLED, DISABLED
preferredlocation	<String>	Read-only	Prefered location.

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
monitor_state	<String>	Read-only	The running state of the monitor on this service. Possible values = UP, DOWN, UNKNOWN, BUSY, OUT OF SERVICE, GOING OUT OF SERVICE, DOWN WHEN GOING OUT OF SERVICE, NS_EMPTY_STR, Unknown, DISABLED
statechangetimesec	<String>	Read-only	Time when last state change happened. Seconds part.
tickssincelaststatechange	<Double>	Read-only	Time in 10 millisecond ticks since the last state change.
threshold	<String>	Read-only	. Possible values = ABOVE, BELOW
clmonowner	<Double>	Read-only	Tells the mon owner of the gslb service. Minimum value = 0 Maximum value = 32
clmonview	<Double>	Read-only	Tells the view id of the monitoring owner.

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
gslbsvchealth	<Integer>	Read-only	This parameter displays effective health of a GSLB service.
glsbsvchealthdescr	<String>	Read-only	Displays the warning message related to health a of GSLB service.
nodefaultbindings	<String>	Read-only	to determine if the configuration will have default ssl CIPHER and ECC curve bindings. Default value: NO Possible values = YES, NO
__count	<Double>	Read-only	count parameter

Operations

(click to see Properties)

ADD| DELETE| UPDATE| UNSET| RENAME| GET (ALL)| GET| COUNT

Some options that you can use for each operations:

- **Getting warnings in response:**NITRO allows you to get warnings in an operation by specifying the "warning" query parameter as "yes". For example, to get warnings while connecting to the Citrix ADC appliance, the URL is as follows:

  http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/login?warning=yes

  If any, the warnings are displayed in the response payload with the HTTP code "209 X-NITRO-WARNING".
- **Authenticated access for individual NITRO operations:**NITRO allows you to logon to the Citrix ADC appliance to perform individual operations. You can use this option instead of

creating a NITRO session (using the login object) and then using that session to perform all operations,

To do this, you must specify the username and password in the request header of the NITRO request as follows:

X-NITRO-USER:<username>

X-NITRO-PASS:<password>

Note: In such cases, make sure that the request header DOES not include the following:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue>

`*Note:`*

Mandatory parameters are marked in redand placeholder content is marked in <green>.

add

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/gslbservice

HTTP Method:POST

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Content-Type:application/json

Request Payload:

```
{"gslbservice":{

<b>"servicename":<String_value>,

</b>"cnameentry":<String_value>,

"ip":<String_value>,

"servername":<String_value>,

"servicetype":<String_value>,

"port":<Integer_value>,
```

```
"publicip":<String_value>,

"publicport":<Integer_value>,

"maxclient":<Double_value>,

"healthmonitor":<String_value>,

<b>"sitename":<String_value>,

</b>"state":<String_value>,

"cip":<String_value>,

"cipheader":<String_value>,

"sitepersistence":<String_value>,

"cookietimeout":<Double_value>,

"siteprefix":<String_value>,

"clttimeout":<Double_value>,

"svrtimeout":<Double_value>,

"maxbandwidth":<Double_value>,

"downstateflush":<String_value>,

"maxaaausers":<Double_value>,

"monthreshold":<Double_value>,

"hashid":<Double_value>,

"comment":<String_value>,

"appflowlog":<String_value>,

"naptrreplacement":<String_value>,

"naptrorder":<Double_value>,
```

```
"naptrservices":<String_value>,

"naptrdomainttl":<Double_value>,

"naptrpreference":<Double_value>

}}
```

Response:

HTTP Status Code on Success: 201 Created HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error

delete

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/gslbservice/servicename_value<String>

HTTP Method:DELETE

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue>

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error

update

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/gslbservice

HTTP Method:PUT

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Content-Type:application/json

Request Payload:

```
{"gslbservice":{
<b>"servicename":<String_value>,
</b>"ipaddress":<String_value>,
"publicip":<String_value>,
"publicport":<Integer_value>,
"cip":<String_value>,
"cipheader":<String_value>,
"sitepersistence":<String_value>,
"siteprefix":<String_value>,
"maxclient":<Double_value>,
"healthmonitor":<String_value>,
"maxbandwidth":<Double_value>,
"downstateflush":<String_value>,
"maxaaausers":<Double_value>,
"viewname":<String_value>,
"viewip":<String_value>,
"monthreshold":<Double_value>,
"weight":<Double_value>,
"monitor_name_svc":<String_value>,
"hashid":<Double_value>,
"comment":<String_value>,
```

```
"appflowlog":<String_value>,

"naptrorder":<Double_value>,

"naptrpreference":<Double_value>,

"naptrservices":<String_value>,

"naptrreplacement":<String_value>,

"naptrdomainttl":<Double_value>

}}
```

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error

unset

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/gslbservice?**action=unset**

HTTP Method:POST

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Content-Type:application/json

Request Payload:

```
{"gslbservice":{

<b>"servicename":<String_value>,

</b>"publicip":true,

"publicport":true,
```

```
"cip":true,

"cipheader":true,

"sitepersistence":true,

"siteprefix":true,

"maxclient":true,

"healthmonitor":true,

"maxbandwidth":true,

"downstateflush":true,

"maxaaausers":true,

"monthreshold":true,

"hashid":true,

"comment":true,

"appflowlog":true,

"naptrorder":true,

"naptrpreference":true,

"naptrservices":true,

"naptrreplacement":true,

"naptrdomainttl":true

}}
```


Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error

rename

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/gslbservice?**action=rename**

HTTP Method:POST

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Content-Type:application/json

Request Payload:

```
{"gslbservice":{

<b>"servicename":<String_value>,

</b><b>"newname":<String_value>

</b>}}
```

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error

get (all)

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/gslbservice

Query-parameters:

attrs

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/gslbservice?**attrs=property-name1,property-name2**

Use this query parameter to specify the resource details that you want to retrieve.

filter

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/gslbservice?**filter=property-name1:property-val1,property-name2:property-val2**

Use this query-parameter to get the filtered set of gslbservice resources configured on Citrix ADC. Filtering can be done on any of the properties of the resource.

view

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/gslbservice?**view=summary**

Note: By default, the retrieved results are displayed in detail view (?view=detail).

pagination

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/gslbservice?**pagesize=#no;pageno=#no**

Use this query-parameter to get the gslbservice resources in chunks.

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "gslbservice": [ {

"servicename":<String_value>,

"gslb":<String_value>,

"ipaddress":<String_value>,
```

```
"ip":<String_value>,

"servername":<String_value>,

"servicetype":<String_value>,

"port":<Integer_value>,

"publicip":<String_value>,

"publicport":<Integer_value>,

"maxclient":<Double_value>,

"maxaaausers":<Double_value>,

"sitename":<String_value>,

"svrstate":<String_value>,

"svreffgslbstate":<String_value>,

"gslbthreshold":<Integer_value>,

"gslbsvcstats":<Integer_value>,

"state":<String_value>,

"monstate":<String_value>,

"cip":<String_value>,

"cipheader":<String_value>,

"sitepersistence":<String_value>,

"siteprefix":<String_value>,

"clttimeout":<Double_value>,

"svrtimeout":<Double_value>,

"preferredlocation":<String_value>,
```

```
"maxbandwidth":<Double_value>,

"downstateflush":<String_value>,

"cnameentry":<String_value>,

"viewname":<String_value>,

"viewip":<String_value>,

"weight":<Double_value>,

"monthreshold":<Double_value>,

"monitor_state":<String_value>,

"hashid":<Double_value>,

"comment":<String_value>,

"healthmonitor":<String_value>,

"appflowlog":<String_value>,

"statechangetimesec":<String_value>,

"tickssincelaststatechange":<Double_value>,

"threshold":<String_value>,

"clmonowner":<Double_value>,

"clmonview":<Double_value>,

"naptrorder":<Double_value>,

"naptrpreference":<Double_value>,

"naptrservices":<String_value>,

"naptrreplacement":<String_value>,

"naptrdomainttl":<Double_value>,
```

```
"gslbsvchealth":<Integer_value>,

"glsbsvchealthdescr":<String_value>,

"nodefaultbindings":<String_value>

}]}
```

get

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/gslbservice/servicename_value<String>

Query-parameters:

attrs

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/gslbservice/servicename_value<String>?**attrs=property-name1,property-name2**

Use this query parameter to specify the resource details that you want to retrieve.

view

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/gslbservice/servicename_value<String>?**view=summary**

Note: By default, the retrieved results are displayed in detail view (?view=detail).

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "gslbservice": [ {
"servicename":<String_value>,
"gslb":<String_value>,
"ipaddress":<String_value>,
"ip":<String_value>,
"servername":<String_value>,
"servicetype":<String_value>,
"port":<Integer_value>,
"publicip":<String_value>,
"publicport":<Integer_value>,
"maxclient":<Double_value>,
"maxaaausers":<Double_value>,
"sitename":<String_value>,
"svrstate":<String_value>,
"svreffgslbstate":<String_value>,
"gslbthreshold":<Integer_value>,
"gslbsvcstats":<Integer_value>,
"state":<String_value>,
"monstate":<String_value>,
"cip":<String_value>,
"cipheader":<String_value>,
```

```
"sitepersistence":<String_value>,

"siteprefix":<String_value>,

"clttimeout":<Double_value>,

"svrtimeout":<Double_value>,

"preferredlocation":<String_value>,

"maxbandwidth":<Double_value>,

"downstateflush":<String_value>,

"cnameentry":<String_value>,

"viewname":<String_value>,

"viewip":<String_value>,

"weight":<Double_value>,

"monthreshold":<Double_value>,

"monitor_state":<String_value>,

"hashid":<Double_value>,

"comment":<String_value>,

"healthmonitor":<String_value>,

"appflowlog":<String_value>,

"statechangetimesec":<String_value>,

"tickssincelaststatechange":<Double_value>,

"threshold":<String_value>,

"clmonowner":<Double_value>,

"clmonview":<Double_value>,
```

```
"naptrorder":<Double_value>,

"naptrpreference":<Double_value>,

"naptrservices":<String_value>,

"naptrreplacement":<String_value>,

"naptrdomainttl":<Double_value>,

"gslbsvchealth":<Integer_value>,

"glsbsvchealthdescr":<String_value>,

"nodefaultbindings":<String_value>

}]}
```

count

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/gslbservice?**count=yes**

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "gslbservice": [ { "__count": "#no"} ] }
```

gslbservice_binding

Binding object which returns the resources bound to gslbservice.

Properties

(click to see Operations)

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
servicename	<String>	Read-write	Name of the GSLB service. Minimum length = 1
gslbservice_dnsview_binding	<gslbservice_dnsview_binding[]>	Read-only	dnsview that can be bound to gslbservice.
gslbservice_lbmonitor_binding	<gslbservice_lbmonitor_binding[]>	Read-only	lbmonitor that can be bound to gslbservice.

Operations

(click to see Properties)

GET| GET (ALL)

Some options that you can use for each operations:

- **Getting warnings in response:**NITRO allows you to get warnings in an operation by specifying the "warning" query parameter as "yes". For example, to get warnings while connecting to the Citrix ADC appliance, the URL is as follows:

  http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/login?warning=yes

  If any, the warnings are displayed in the response payload with the HTTP code "209 X-NITRO-WARNING".
- **Authenticated access for individual NITRO operations:**NITRO allows you to logon to the Citrix ADC appliance to perform individual operations. You can use this option instead of creating a NITRO session (using the login object) and then using that session to perform all operations,

To do this, you must specify the username and password in the request header of the NITRO request as follows:

X-NITRO-USER:<username>

X-NITRO-PASS:<password>

Note: In such cases, make sure that the request header DOES not include the following:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue>

`*Note:`*

Mandatory parameters are marked in redand placeholder content is marked in <green>.

get

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/gslbservice_binding/servicename_value<String>

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "gslbservice_binding": [ {

"servicename":<String_value>,

"gslbservice_lbmonitor_binding":<gslbservice_lbmonitor_binding[]_value>,

"gslbservice_dnsview_binding":<gslbservice_dnsview_binding[]_value>
```

```
}]}
```

get (all)

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/gslbservice_binding

Query-parameters:

bulkbindings

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/gslbservice_binding?**bulkbindings=yes**

NITRO allows you to fetch bindings in bulk.

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "gslbservice_binding": [ {

"servicename":<String_value>,

"gslbservice_lbmonitor_binding":<gslbservice_lbmonitor_binding[]_value>,

"gslbservice_dnsview_binding":<gslbservice_dnsview_binding[]_value>

}]}
```

gslbservice_dnsview_binding

Binding object showing the dnsview that can be bound to gslbservice.

Properties

(click to see Operations)

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
viewname	<String>	Read-write	Name of the DNS view of the service. A DNS view is used in global server load balancing (GSLB) to return a predetermined IP address to a specific group of clients, which are identified by using a DNS policy. Minimum length = 1
servicename	<String>	Read-write	Name of the GSLB service. Minimum length = 1
viewip	<String>	Read-write	IP address to be used for the given view.
__count	<Double>	Read-write	count parameter

Operations

(click to see Properties)

ADD:| DELETE:| GET| GET (ALL)| COUNT

Some options that you can use for each operations:

- **Getting warnings in response:**NITRO allows you to get warnings in an operation by specifying the "warning" query parameter as "yes". For example, to get warnings while connecting to the Citrix ADC appliance, the URL is as follows:

  http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/login?warning=yes

  If any, the warnings are displayed in the response payload with the HTTP code "209 X-NITRO-WARNING".

- **Authenticated access for individual NITRO operations:**NITRO allows you to logon to the Citrix ADC appliance to perform individual operations. You can use this option instead of creating a NITRO session (using the login object) and then using that session to perform all operations,

  To do this, you must specify the username and password in the request header of the NITRO request as follows:

  X-NITRO-USER:<username>

  X-NITRO-PASS:<password>

  Note: In such cases, make sure that the request header DOES not include the following:

  Cookie:NITRO_AUTH_TOKEN=<tokenvalue>

`*Note:`*

Mandatory parameters are marked in redand placeholder content is marked in <green>.

add

URL:http://<Citrix-ADC-IP-address(NSIP)/nitro/v1/config/gslbservice_dnsview_binding

HTTP Method:PUT

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Content-Type:application/json

Request Payload:

```
{
```

```
"gslbservice_dnsview_binding":{

<b>"servicename":<String_value>,

</b>"viewname":<String_value>,

"viewip":<String_value>

}}
```

Response:

HTTP Status Code on Success: 201 Created HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error

delete

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/gslbservice_dnsview_binding/servicename_value<String>

Query-parameters:

args

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/gslbservice_dnsview_binding/servicename_value<String>?**args=viewname:<String_value>**

HTTP Method:DELETE

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue>

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error

get

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/gslbservice_dnsview_binding/servicename_value<String>

Query-parameters:

filter

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/gslbservice_dnsview_binding/servicename_value<String>?**filter=property-name1:property-value1,property-name2:property-value2**

Use this query-parameter to get the filtered set of gslbservice_dnsview_binding resources configured on Citrix ADC. Filtering can be done on any of the properties of the resource.

pagination

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/gslbservice_dnsview_binding/servicename_value<String>?**pagesize=#no;pageno=#no**

Use this query-parameter to get the gslbservice_dnsview_binding resources in chunks.

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "gslbservice_dnsview_binding": [ {

"viewname":<String_value>,
```

```
"servicename":<String_value>,

"viewip":<String_value>

}]}
```

get (all)

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/gslbservice_dnsview_binding

Query-parameters:

bulkbindings

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/gslbservice_dnsview_binding?**bulkbindings=yes**

NITRO allows you to fetch bindings in bulk.

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "gslbservice_dnsview_binding": [ {

"viewname":<String_value>,

"servicename":<String_value>,
```

```
"viewip":<String_value>

}]}
```

count

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/gslbservice_dnsview_binding/servicename_value<String>?**count=yes**

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{"gslbservice_dnsview_binding": [ { "__count": "#no"} ] }
```

gslbservice_lbmonitor_binding

Binding object showing the lbmonitor that can be bound to gslbservice.

Properties

(click to see Operations)

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
servicename	<String>	Read-write	Name of the GSLB service.

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
			Minimum length = 1
weight	<Double>	Read-write	Weight to assign to the monitor-service binding. A larger number specifies a greater weight. Contributes to the monitoring threshold, which determines the state of the service. Minimum value = 1 Maximum value = 100
monitor_name	<String>	Read-write	Monitor name.
monstate	<String>	Read-write	State of the monitor bound to gslb service. Possible values = ENABLED, DISABLED
monstatparam2	<Integer>	Read-only	Second parameter for use with message code.
monstatcode	<Integer>	Read-only	The code indicating the monitor response.
lastresponse	<String>	Read-only	Displays the gslb monitor status in string format.
failedprobes	<Double>	Read-only	Number of the current failed monitoring probes.

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
monstatparam3	<Integer>	Read-only	Third parameter for use with message code.
monitor_state	<String>	Read-only	The running state of the monitor on this service. Possible values = UP, DOWN, UNKNOWN, BUSY, OUT OF SERVICE, GOING OUT OF SERVICE, DOWN WHEN GOING OUT OF SERVICE, NS_EMPTY_STR, Unknown, DISABLED
monitortotalprobes	<Double>	Read-only	Total number of probes sent to monitor this service.
monstatparam1	<Integer>	Read-only	First parameter for use with message code.
responsetime	<Double>	Read-only	Response time of this monitor.
monitortotalfailedprobes	<Double>	Read-only	Total number of failed probes.
monitorcurrentfailedprobes	<Double>	Read-only	Total number of currently failed probes.
totalfailedprobes	<Double>	Read-only	The total number of failed probs.

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
__count	<Double>	Read-only	count parameter

Operations

(click to see Properties)

ADD:| DELETE:| GET| GET (ALL)| COUNT

Some options that you can use for each operations:

- **Getting warnings in response:**NITRO allows you to get warnings in an operation by specifying the "warning" query parameter as "yes". For example, to get warnings while connecting to the Citrix ADC appliance, the URL is as follows:

  http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/login?warning=yes

  If any, the warnings are displayed in the response payload with the HTTP code "209 X-NITRO-WARNING".
- **Authenticated access for individual NITRO operations:**NITRO allows you to logon to the Citrix ADC appliance to perform individual operations. You can use this option instead of creating a NITRO session (using the login object) and then using that session to perform all operations,

  To do this, you must specify the username and password in the request header of the NITRO request as follows:

  X-NITRO-USER:<username>

  X-NITRO-PASS:<password>

  Note: In such cases, make sure that the request header DOES not include the following:

  Cookie:NITRO_AUTH_TOKEN=<tokenvalue>

`*Note:`*

Mandatory parameters are marked in redand placeholder content is marked in <green>.

add

URL:http://<Citrix-ADC-IP-address(NSIP)/nitro/v1/config/gslbservice_lbmonitor_binding

HTTP Method:PUT

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Content-Type:application/json

Request Payload:

```
{

"gslbservice_lbmonitor_binding":{

<b>"servicename":<String_value>,

</b>"monitor_name":<String_value>,

"monstate":<String_value>,

"weight":<Double_value>

}}
```

Response:

HTTP Status Code on Success: 201 Created HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error

delete

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/gslbservice_lbmonitor_binding/servicename_value<String>

Query-parameters:

args

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/gslbservice_lbmonitor_binding/servicename_value<String>?**args=monitor_name:<String_value>**

HTTP Method:DELETE

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue>

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error

get

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/gslbservice_lbmonitor_binding/servicename_value<String>

Query-parameters:

filter

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/gslbservice_lbmonitor_binding/servicename_value<String>?**filter=property-name1:property-value1,property-name2:property-value2**

Use this query-parameter to get the filtered set of gslbservice_lbmonitor_binding resources configured on Citrix ADC. Filtering can be done on any of the properties of the resource.

pagination

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/gslbservice_lbmonitor_binding/servicename_value<String>?**pagesize=#no;pageno=#no**

Use this query-parameter to get the gslbservice_lbmonitor_binding resources in chunks.

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "gslbservice_lbmonitor_binding": [ {

"servicename":<String_value>,

"weight":<Double_value>,

"monitor_name":<String_value>,

"monstate":<String_value>,

"monstatparam2":<Integer_value>,

"monstatcode":<Integer_value>,

"lastresponse":<String_value>,

"failedprobes":<Double_value>,

"monstatparam3":<Integer_value>,

"monitor_state":<String_value>,

"monitortotalprobes":<Double_value>,

"monstatparam1":<Integer_value>,

"responsetime":<Double_value>,

"monitortotalfailedprobes":<Double_value>,

"monitorcurrentfailedprobes":<Double_value>,

"totalfailedprobes":<Double_value>

}]}
```

get (all)

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/gslbservice_lbmonitor_binding

Query-parameters:

bulkbindings

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/gslbservice_lbmonitor_binding?**bulkbindings=yes**

NITRO allows you to fetch bindings in bulk.

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "gslbservice_lbmonitor_binding": [ {

"servicename":<String_value>,

"weight":<Double_value>,

"monitor_name":<String_value>,

"monstate":<String_value>,

"monstatparam2":<Integer_value>,

"monstatcode":<Integer_value>,
```

```
"lastresponse":<String_value>,

"failedprobes":<Double_value>,

"monstatparam3":<Integer_value>,

"monitor_state":<String_value>,

"monitortotalprobes":<Double_value>,

"monstatparam1":<Integer_value>,

"responsetime":<Double_value>,

"monitortotalfailedprobes":<Double_value>,

"monitorcurrentfailedprobes":<Double_value>,

"totalfailedprobes":<Double_value>

}]}
```

count

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/gslbservice_lbmonitor_binding/servicename_value<String>?**count=yes**

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{"gslbservice_lbmonitor_binding": [ { "__count": "#no"} ] }
```

gslbsite

Configuration for GSLB site resource.

Properties

(click to see Operations)

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
sitename	<String>	Read-write	Name for the GSLB site. Must begin with an ASCII alphanumeric or underscore (_) character, and must contain only ASCII alphanumeric, underscore, hash (#), period (.), space, colon (:), at (@), equals (=), and hyphen (-) characters. Cannot be changed after the virtual server is created. CLI Users: If the name includes one or more spaces, enclose the name in double or single quotation marks (for example, "my

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
			gslbsite" or 'my gslbsite'). Minimum length = 1
sitetype	<String>	Read-write	Type of site to create. If the type is not specified, the appliance automatically detects and sets the type on the basis of the IP address being assigned to the site. If the specified site IP address is owned by the appliance (for example, a MIP address or SNIP address), the site is a local site. Otherwise, it is a remote site. Default value: NONE Possible values = REMOTE, LOCAL
siteipaddress	<String>	Read-write	IP address for the GSLB site. The GSLB site uses this IP address to communicate with other GSLB sites. For a local site, use any IP address that is owned by the appliance (for example, a SNIP or MIP address, or the

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
			IP address of the ADNS service). Minimum length = 1
publicip	<String>	Read-write	Public IP address for the local site. Required only if the appliance is deployed in a private address space and the site has a public IP address hosted on an external firewall or a NAT device. Minimum length = 1
metricexchange	<String>	Read-write	Exchange metrics with other sites. Metrics are exchanged by using Metric Exchange Protocol (MEP). The appliances in the GSLB setup exchange health information once every second. If you disable metrics exchange, you can use only static load balancing methods (such as round robin, static proximity, or the hash-based methods), and if you disable metrics exchange when a

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
			dynamic load balancing method (such as least connection) is in operation, the appliance falls back to round robin. Also, if you disable metrics exchange, you must use a monitor to determine the state of GSLB services. Otherwise, the service is marked as DOWN. Default value: ENABLED Possible values = ENABLED, DISABLED
nwmetricexchange	<String>	Read-write	Exchange, with other GSLB sites, network metrics such as round-trip time (RTT), learned from communications with various local DNS (LDNS) servers used by clients. RTT information is used in the dynamic RTT load balancing method, and is

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
			exchanged every 5 seconds. Default value: ENABLED Possible values = ENABLED, DISABLED
sessionexchange	<String>	Read-write	Exchange persistent session entries with other GSLB sites every five seconds. Default value: ENABLED Possible values = ENABLED, DISABLED
triggermonitor	<String>	Read-write	Specify the conditions under which the GSLB service must be monitored by a monitor, if one is bound. Available settings function as follows: * ALWAYS - Monitor the GSLB service at all times. * MEPDOWN - Monitor the GSLB service only when the exchange of metrics through the Metrics Exchange

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
			Protocol (MEP) is disabled. MEPDOWN_SVCDOWN - Monitor the service in either of the following situations: * The exchange of metrics through MEP is disabled. * The exchange of metrics through MEP is enabled but the status of the service, learned through metrics exchange, is DOWN. Default value: ALWAYS Possible values = ALWAYS, MEPDOWN, MEPDOWN_SVCDOWN
parentsite	<String>	Read-write	Parent site of the GSLB site, in a parent-child topology.
clip	<String>	Read-write	Cluster IP address. Specify this parameter to connect

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
			to the remote cluster site for GSLB auto-sync. Note: The cluster IP address is defined when creating the cluster.
publicclip	<String>	Read-write	IP address to be used to globally access the remote cluster when it is deployed behind a NAT. It can be same as the normal cluster IP address.
naptrreplacementsuffix	<String>	Read-write	The naptr replacement suffix configured here will be used to construct the naptr replacement field in NAPTR record. Minimum length = 1
backupparentlist	<String[]>	Read-write	The list of backup gslb sites configured in preferred order. Need to be parent gsb sites. Default value: "None"
status	<String>	Read-only	Current metric exchange status. Possible values = ACTIVE, INACTIVE, DOWN

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
persistencemepstatus	<String>	Read-only	Network metric and persistence exchange MEP connection status. Possible values = ACTIVE, INACTIVE, DOWN
version	<Double>	Read-only	will be true if the remote site's version is ncore compatible with the local site.(>= 9.2).
curbackupparentip	<String>	Read-only	Current active backup parent IP address since the configured is DOWN.
__count	<Double>	Read-only	count parameter

Operations

(click to see Properties)

ADD| DELETE| UPDATE| UNSET| GET (ALL)| GET| COUNT

Some options that you can use for each operations:

- **Getting warnings in response:**NITRO allows you to get warnings in an operation by specifying the "warning" query parameter as "yes". For example, to get warnings while connecting to the Citrix ADC appliance, the URL is as follows:

  http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/login?warning=yes

  If any, the warnings are displayed in the response payload with the HTTP code "209 X-NITRO-WARNING".
- **Authenticated access for individual NITRO operations:**NITRO allows you to logon to the Citrix ADC appliance to perform individual operations. You can use this option instead of

creating a NITRO session (using the login object) and then using that session to perform all operations,

To do this, you must specify the username and password in the request header of the NITRO request as follows:

X-NITRO-USER:<username>

X-NITRO-PASS:<password>

Note: In such cases, make sure that the request header DOES not include the following:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue>

`*Note:`*

Mandatory parameters are marked in redand placeholder content is marked in <green>.

add

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/gslbsite

HTTP Method:POST

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Content-Type:application/json

Request Payload:

```
{"gslbsite":{

<b>"sitename":<String_value>,

</b>"sitetype":<String_value>,

<b>"siteipaddress":<String_value>,

</b>"publicip":<String_value>,

"metricexchange":<String_value>,

"nwmetricexchange":<String_value>,
```

```
"sessionexchange":<String_value>,

"triggermonitor":<String_value>,

"parentsite":<String_value>,

"clip":<String_value>,

"publicclip":<String_value>,

"naptrreplacementsuffix":<String_value>,

"backupparentlist":<String[]_value>

}}
```

Response:

HTTP Status Code on Success: 201 Created HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error

delete

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/gslbsite/sitename_value<String>

HTTP Method:DELETE

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue>

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error

update

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/gslbsite

HTTP Method:PUT

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Content-Type:application/json

Request Payload:

```
{"gslbsite":{

<b>"sitename":<String_value>,

</b>"metricexchange":<String_value>,

"nwmetricexchange":<String_value>,

"sessionexchange":<String_value>,

"triggermonitor":<String_value>,

"naptrreplacementsuffix":<String_value>,

"backupparentlist":<String[]_value>

}}
```

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error

unset

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/gslbsite?**action=unset**

HTTP Method:POST

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Content-Type:application/json

Request Payload:

```
{"gslbsite":{

<b>"sitename":<String_value>,

</b>"metricexchange":true,

"nwmetricexchange":true,

"sessionexchange":true,

"triggermonitor":true,

"naptrreplacementsuffix":true,

"backupparentlist":true

}}
```

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error

get (all)

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/gslbsite

Query-parameters:

attrs

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/gslbsite?**attrs=property-name1,property-name2**

Use this query parameter to specify the resource details that you want to retrieve.

filter

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/gslbsite?**filter=property-name1:property-val1,property-name2:property-val2**

Use this query-parameter to get the filtered set of gslbsite resources configured on Citrix ADC. Filtering can be done on any of the properties of the resource.

view

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/gslbsite?**view=summary**

Note: By default, the retrieved results are displayed in detail view (?view=detail).

pagination

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/gslbsite?**pagesize=#no;pageno=#no**

Use this query-parameter to get the gslbsite resources in chunks.

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "gslbsite": [ {

"sitename":<String_value>,

"sitetype":<String_value>,

"siteipaddress":<String_value>,

"publicip":<String_value>,

"metricexchange":<String_value>,
```

```
"status":<String_value>,

"persistencemepstatus":<String_value>,

"nwmetricexchange":<String_value>,

"sessionexchange":<String_value>,

"triggermonitor":<String_value>,

"parentsite":<String_value>,

"version":<Double_value>,

"clip":<String_value>,

"publicclip":<String_value>,

"naptrreplacementsuffix":<String_value>,

"backupparentlist":<String[]_value>,

"curbackupparentip":<String_value>

}]}
```

get

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/gslbsite/sitename_value<String>

Query-parameters:

attrs

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/gslbsite/sitename_value<String>?**attrs=property-name1,property-name2**

Use this query parameter to specify the resource details that you want to retrieve.

view

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/gslbsite/sitename_value<String>?**view=summary**

Note: By default, the retrieved results are displayed in detail view (?view=detail).

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "gslbsite": [ {

"sitename":<String_value>,

"sitetype":<String_value>,

"siteipaddress":<String_value>,

"publicip":<String_value>,

"metricexchange":<String_value>,

"status":<String_value>,

"persistencemepstatus":<String_value>,

"nwmetricexchange":<String_value>,

"sessionexchange":<String_value>,

"triggermonitor":<String_value>,
```

```
"parentsite":<String_value>,

"version":<Double_value>,

"clip":<String_value>,

"publicclip":<String_value>,

"naptrreplacementsuffix":<String_value>,

"backupparentlist":<String[]_value>,

"curbackupparentip":<String_value>

}]}
```

count

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/gslbsite?**count=yes**

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "gslbsite": [ { "__count": "#no"} ] }
```

gslbsite_binding

Binding object which returns the resources bound to gslbsite.

Properties

(click to see Operations)

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
sitename	<String>	Read-write	Name of the GSLB site. If you specify a site name, details of all the site's constituent services are also displayed. Minimum length = 1
gslbsite_gslbservicegroup_binding	<gslbsite_gslbservicegroup_binding[]>	Read-only	gslbservicegroup that can be bound to gslbsite.
gslbsite_gslbservice_binding	<gslbsite_gslbservice_binding[]>	Read-only	gslbservice that can be bound to gslbsite.
gslbsite_gslbservicegroupmember_binding	<gslbsite_gslbservicegroupmember_binding[]>	Read-only	gslbservicegroupmember that can be bound to gslbsite.

Operations

(click to see Properties)

GET| GET (ALL)

Some options that you can use for each operations:

- **Getting warnings in response:**NITRO allows you to get warnings in an operation by specifying the "warning" query parameter as "yes". For example, to get warnings while connecting to the Citrix ADC appliance, the URL is as follows:

  http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/login?warning=yes

  If any, the warnings are displayed in the response payload with the HTTP code "209 X-NITRO-WARNING".

- **Authenticated access for individual NITRO operations:**NITRO allows you to logon to the Citrix ADC appliance to perform individual operations. You can use this option instead of creating a NITRO session (using the login object) and then using that session to perform all operations,

  To do this, you must specify the username and password in the request header of the NITRO request as follows:

  X-NITRO-USER:<username>

  X-NITRO-PASS:<password>

  Note: In such cases, make sure that the request header DOES not include the following:

  Cookie:NITRO_AUTH_TOKEN=<tokenvalue>

`*Note:`*

Mandatory parameters are marked in redand placeholder content is marked in <green>.

get

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/gslbsite_binding/sitename_value<String>

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "gslbsite_binding": [ {
```

```
"sitename":<String_value>,

"gslbsite_gslbservicegroupmember_binding":<gslbsite_gslbservicegroupmember
_binding[]_value>,

"gslbsite_gslbservicegroup_binding":<gslbsite_gslbservicegroup_binding[]_v
alue>,

"gslbsite_gslbservice_binding":<gslbsite_gslbservice_binding[]_value>

}]}
```

get (all)

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/gslbsite_binding

Query-parameters:

bulkbindings

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/gslbsite_binding?**bulkbindings=yes**

NITRO allows you to fetch bindings in bulk.

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "gslbsite_binding": [ {
```

```
"sitename":<String_value>,

"gslbsite_gslbservicegroupmember_binding":<gslbsite_gslbservicegroupmember
_binding[]_value>,

"gslbsite_gslbservicegroup_binding":<gslbsite_gslbservicegroup_binding[]_v
alue>,

"gslbsite_gslbservice_binding":<gslbsite_gslbservice_binding[]_value>

}]}
```

gslbsite_gslbservice_binding

Binding object showing the gslbservice that can be bound to gslbsite.

Properties

(click to see Operations)

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
servicename	<String>	Read-write	Service name.
sitename	<String>	Read-write	Name of the GSLB site. If you specify a site name, details of all the site's constituent services are also displayed. Minimum length = 1
cnameentry	<String>	Read-only	The cname of the gslb service. Minimum length = 1
state	<String>	Read-only	State of the gslb service. Possible values = UP, DOWN, UNKNOWN, BUSY,

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
			OUT OF SERVICE, GOING OUT OF SERVICE, DOWN WHEN GOING OUT OF SERVICE, NS_EMPTY_STR, Unknown, DISABLED
servicetype	<String>	Read-only	Service type. Possible values = HTTP, FTP, TCP, UDP, SSL, SSL_BRIDGE, SSL_TCP, DTLS, NNTP, DNS, DHCPRA, ANY, SIP_UDP, SIP_TCP, SIP_SSL, DNS_TCP, RTSP, PUSH, SSL_PUSH, RADIUS, RDP, MYSQL, MSSQL, DIAMETER, SSL_DIAMETER, TFTP, ORACLE, SMPP, SYSLOGTCP, SYSLOGUDP, FIX, SSL_FIX, PROXY, USER_TCP, USER_SSL_TCP, QUIC, IPFIX, LOGSTREAM

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
port	<Integer>	Read-only	Port number of the gslb service. Range 1 - 65535 * in CLI is represented as 65535 in NITRO API
ipaddress	<String>	Read-only	IP Address of the gslb service.
__count	<Double>	Read-only	count parameter

Operations

(click to see Properties)

GET| GET (ALL)| COUNT

Some options that you can use for each operations:

- **Getting warnings in response:**NITRO allows you to get warnings in an operation by specifying the "warning" query parameter as "yes". For example, to get warnings while connecting to the Citrix ADC appliance, the URL is as follows:

  http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/login?warning=yes

  If any, the warnings are displayed in the response payload with the HTTP code "209 X-NITRO-WARNING".

- **Authenticated access for individual NITRO operations:**NITRO allows you to logon to the Citrix ADC appliance to perform individual operations. You can use this option instead of creating a NITRO session (using the login object) and then using that session to perform all operations,

  To do this, you must specify the username and password in the request header of the NITRO request as follows:

  X-NITRO-USER:<username>

  X-NITRO-PASS:<password>

Note: In such cases, make sure that the request header DOES not include the following:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue>

`*Note:`*

Mandatory parameters are marked in redand placeholder content is marked in <green>.

get

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/gslbsite_gslbservice_binding/sitename_value<String>

Query-parameters:

filter

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/gslbsite_gslbservice_binding/sitename_value<String>?**filter=property-name1:property-value1,property-name2:property-value2**

Use this query-parameter to get the filtered set of gslbsite_gslbservice_binding resources configured on Citrix ADC. Filtering can be done on any of the properties of the resource.

pagination

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/gslbsite_gslbservice_binding/sitename_value<String>?**pagesize=#no;pageno=#no**

Use this query-parameter to get the gslbsite_gslbservice_binding resources in chunks.

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "gslbsite_gslbservice_binding": [ {

"servicename":<String_value>,

"sitename":<String_value>,

"cnameentry":<String_value>,

"state":<String_value>,

"servicetype":<String_value>,

"port":<Integer_value>,

"ipaddress":<String_value>

}]}
```

get (all)

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/gslbsite_gslbservice_binding

Query-parameters:

bulkbindings

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/gslbsite_gslbservice_binding?**bulkbindings=yes**

NITRO allows you to fetch bindings in bulk.

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "gslbsite_gslbservice_binding": [ {

"servicename":<String_value>,

"sitename":<String_value>,

"cnameentry":<String_value>,

"state":<String_value>,

"servicetype":<String_value>,

"port":<Integer_value>,

"ipaddress":<String_value>

}]}
```

count

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/gslbsite_gslbservice_binding/sitename_value<String>?**count=yes**

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{"gslbsite_gslbservice_binding": [ { "__count": "#no"} ] }
```

gslbsyncstatus

Configuration for sync status resource.

Properties

(click to see Operations)

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
summary	<Boolean>	Read-write	sync status summary to be displayed in one line (Success/ Failure), in case of Failure stating reason for failure.
response	<String>	Read-only	gslb sync status as text blob.

Operations

(click to see Properties)

GET (ALL)

Some options that you can use for each operations:

- **Getting warnings in response:**NITRO allows you to get warnings in an operation by specifying the "warning" query parameter as "yes". For example, to get warnings while connecting to the Citrix ADC appliance, the URL is as follows:

  http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/login?warning=yes

  If any, the warnings are displayed in the response payload with the HTTP code "209 X-NITRO-WARNING".

- **Authenticated access for individual NITRO operations:**NITRO allows you to logon to the Citrix ADC appliance to perform individual operations. You can use this option instead of creating a NITRO session (using the login object) and then using that session to perform all operations,

  To do this, you must specify the username and password in the request header of the NITRO request as follows:

  X-NITRO-USER:<username>

  X-NITRO-PASS:<password>

  Note: In such cases, make sure that the request header DOES not include the following:

  Cookie:NITRO_AUTH_TOKEN=<tokenvalue>

`*Note:`*

Mandatory parameters are marked in redand placeholder content is marked in <green>.

get (all)

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/gslbsyncstatus

Query-parameters:

args

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/gslbsyncstatus?
args=summary:<Boolean_value>

Use this query-parameter to get gslbsyncstatus resources based on additional properties.

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "gslbsyncstatus": [ {

summary:<Boolean_value>"response":<String_value>

}]}
```

gslbvserver

Configuration for Global Server Load Balancing Virtual Server resource.

Properties

(click to see Operations)

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
name	<String>	Read-write	Name for the GSLB virtual server. Must begin with an ASCII alphanumeric or underscore (_) character, and must contain only ASCII alphanumeric, underscore, hash (#), period (.), space, colon (:), at (@), equals (=), and

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
			hyphen (-) characters. Can be changed after the virtual server is created. CLI Users: If the name includes one or more spaces, enclose the name in double or single quotation marks (for example, "my vserver" or 'my vserver'). Minimum length = 1
servicetype	<String>	Read-write	Protocol used by services bound to the virtual server. Possible values = HTTP, FTP, TCP, UDP, SSL, SSL_BRIDGE, SSL_TCP, NNTP, ANY, SIP_UDP, SIP_TCP, SIP_SSL, RADIUS, RDP, RTSP, MYSQL, MSSQL, ORACLE
iptype	<String>	Read-write	The IP type for this GSLB vserver. Default value: IPV4 Possible values = IPV4, IPV6

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
dnsrecordtype	<String>	Read-write	DNS record type to associate with the GSLB virtual server's domain name. Default value: A Possible values = A, AAAA, CNAME, NAPTR
lbmethod	<String>	Read-write	Load balancing method for the GSLB virtual server. Default value: LEASTCONNECTION Possible values = ROUNDROBIN, LEASTCONNECTION, LEASTRESPONSETIME, SOURCEIPHASH, LEASTBANDWIDTH, LEASTPACKETS, STATICPROXIMITY, RTT, CUSTOMLOAD
backupsessiontimeout	<Double>	Read-write	A non zero value enables the feature whose minimum value is 2 minutes. The feature can be disabled by setting the value to zero. The created session

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
			is in effect for a specific client per domain. Minimum value = 0 Maximum value = 1440
backuplbmethod	<String>	Read-write	Backup load balancing method. Becomes operational if the primary load balancing method fails or cannot be used. Valid only if the primary method is based on either round-trip time (RTT) or static proximity. Possible values = ROUNDROBIN, LEASTCONNECTION, LEASTRESPONSETIME, SOURCEIPHASH, LEASTBANDWIDTH, LEASTPACKETS, STATICPROXIMITY, RTT, CUSTOMLOAD
netmask	<String>	Read-write	IPv4 network mask for use in the SOURCEIPHASH load balancing method. Minimum length = 1

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
v6netmasklen	<Double>	Read-write	Number of bits to consider, in an IPv6 source IP address, for creating the hash that is required by the SOURCEIPHASH load balancing method. Default value: 128 Minimum value = 1 Maximum value = 128
tolerance	<Double>	Read-write	Site selection tolerance, in milliseconds, for implementing the RTT load balancing method. If a site's RTT deviates from the lowest RTT by more than the specified tolerance, the site is not considered when the Citrix ADC makes a GSLB decision. The appliance implements the round robin method of global server load balancing between sites whose RTT values are within the specified tolerance. If the tolerance is 0 (zero), the appliance always sends clients

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
			the IP address of the site with the lowest RTT. Minimum value = 0 Maximum value = 100
persistencetype	<String>	Read-write	Use source IP address based persistence for the virtual server. After the load balancing method selects a service for the first packet, the IP address received in response to the DNS query is used for subsequent requests from the same client. Possible values = SOURCEIP, NONE
persistenceid	<Double>	Read-write	The persistence ID for the GSLB virtual server. The ID is a positive integer that enables GSLB sites to identify the GSLB virtual server, and is required if source IP address based or spill over based persistence is enabled on the virtual server.

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
			Minimum value = 0 Maximum value = 65535
persistmask	<String>	Read-write	The optional IPv4 network mask applied to IPv4 addresses to establish source IP address based persistence. Minimum length = 1
v6persistmasklen	<Double>	Read-write	Number of bits to consider in an IPv6 source IP address when creating source IP address based persistence sessions. Default value: 128 Minimum value = 1 Maximum value = 128
timeout	<Double>	Read-write	Idle time, in minutes, after which a persistence entry is cleared. Default value: 2 Minimum value = 2 Maximum value = 1440
edr	<String>	Read-write	Send clients an empty DNS

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
			response when the GSLB virtual server is DOWN. Default value: DISABLED Possible values = ENABLED, DISABLED
ecs	<String>	Read-write	If enabled, respond with EDNS Client Subnet (ECS) option in the response for a DNS query with ECS. The ECS address will be used for persistence and spillover persistence (if enabled) instead of the LDNS address. Persistence mask is ignored if ECS is enabled. Default value: DISABLED Possible values = ENABLED, DISABLED
ecsaddrvalidation	<String>	Read-write	Validate if ECS address is a private or unroutable address and in such cases, use the LDNS IP. Default value: DISABLED

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
			Possible values = ENABLED, DISABLED
mir	<String>	Read-write	Include multiple IP addresses in the DNS responses sent to clients. Default value: DISABLED Possible values = ENABLED, DISABLED
disableprimaryondown	<String>	Read-write	Continue to direct traffic to the backup chain even after the primary GSLB virtual server returns to the UP state. Used when spillover is configured for the virtual server. Default value: DISABLED Possible values = ENABLED, DISABLED
dynamicweight	<String>	Read-write	Specify if the appliance should consider the service count, service weights, or ignore both when using weight-based load balancing methods. The state of the

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
			number of services bound to the virtual server help the appliance to select the service. Default value: DISABLED Possible values = SERVICECOUNT, SERVICEWEIGHT, DISABLED
state	<String>	Read-write	State of the GSLB virtual server. Default value: ENABLED Possible values = ENABLED, DISABLED
considereffectivestate	<String>	Read-write	If the primary state of all bound GSLB services is DOWN, consider the effective states of all the GSLB services, obtained through the Metrics Exchange Protocol (MEP), when determining the state of the GSLB virtual server. To consider the effective state, set the parameter to STATE_ONLY. To disregard the effective state, set

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
			the parameter to NONE. The effective state of a GSLB service is the ability of the corresponding virtual server to serve traffic. The effective state of the load balancing virtual server, which is transferred to the GSLB service, is UP even if only one virtual server in the backup chain of virtual servers is in the UP state. Default value: NONE Possible values = NONE, STATE_ONLY
comment	<String>	Read-write	Any comments that you might want to associate with the GSLB virtual server.
somethod	<String>	Read-write	Type of threshold that, when exceeded, triggers spillover. Available settings function as follows:

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
			* CONNECTION - Spillover occurs when the number of client connections exceeds the threshold. * DYNAMICCONNECTION - Spillover occurs when the number of client connections at the GSLB virtual server exceeds the sum of the maximum client (Max Clients) settings for bound GSLB services. Do not specify a spillover threshold for this setting, because the threshold is implied by the Max Clients settings of the bound GSLB services. * BANDWIDTH - Spillover occurs when the bandwidth consumed by the GSLB virtual server's incoming and outgoing traffic

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
			exceeds the threshold. * HEALTH - Spillover occurs when the percentage of weights of the GSLB services that are UP drops below the threshold. For example, if services gslbSvc1, gslbSvc2, and gslbSvc3 are bound to a virtual server, with weights 1, 2, and 3, and the spillover threshold is 50%, spillover occurs if gslbSvc1 and gslbSvc3 or gslbSvc2 and gslbSvc3 transition to DOWN. * NONE - Spillover does not occur. Possible values = CONNECTION, DYNAMICCONNECTION, BANDWIDTH, HEALTH, NONE
sopersistence	<String>	Read-write	If spillover occurs, maintain source IP address based

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
			persistence for both primary and backup GSLB virtual servers. Default value: DISABLED Possible values = ENABLED, DISABLED
sopersistencetimeout	<Double>	Read-write	Timeout for spillover persistence, in minutes. Default value: 2 Minimum value = 2 Maximum value = 1440
sothreshold	<Double>	Read-write	Threshold at which spillover occurs. Specify an integer for the CONNECTION spillover method, a bandwidth value in kilobits per second for the BANDWIDTH method (do not enter the units), or a percentage for the HEALTH method (do not enter the percentage symbol). Minimum value = 1 Maximum value = 4294967287
sobackupaction	<String>	Read-write	Action to be performed if spillover

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
			is to take effect, but no backup chain to spillover is usable or exists. Possible values = DROP, ACCEPT, REDIRECT
appflowlog	<String>	Read-write	Enable logging appflow flow information. Default value: ENABLED Possible values = ENABLED, DISABLED
backupvserver	<String>	Read-write	Name of the backup GSLB virtual server to which the appliance should to forward requests if the status of the primary GSLB virtual server is down or exceeds its spillover threshold. Minimum length = 1
servicename	<String>	Read-write	Name of the GSLB service for which to change the weight. Minimum length = 1
weight	<Double>	Read-write	Weight to assign to the GSLB service. Minimum value = 1

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
			Maximum value = 100
domainname	<String>	Read-write	Domain name for which to change the time to live (TTL) and/or backup service IP address. Minimum length = 1
ttl	<Double>	Read-write	Time to live (TTL) for the domain. Minimum value = 1
backupip	<String>	Read-write	The IP address of the backup service for the specified domain name. Used when all the services bound to the domain are down, or when the backup chain of virtual servers is down. Minimum length = 1
cookie_domain	<String>	Read-write	The cookie domain for the GSLB site. Used when inserting the GSLB site cookie in the HTTP response. Minimum length = 1
cookietimeout	<Double>	Read-write	Timeout, in minutes, for the GSLB site cookie. Minimum value = 0

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
			Maximum value = 1440
sitedomainttl	<Double>	Read-write	TTL, in seconds, for all internally created site domains (created when a site prefix is configured on a GSLB service) that are associated with this virtual server. Minimum value = 1
newname	<String>	Read-write	New name for the GSLB virtual server. Minimum length = 1
curstate	<String>	Read-only	State of the gslb vserver. Possible values = UP, DOWN, UNKNOWN, BUSY, OUT OF SERVICE, GOING OUT OF SERVICE, DOWN WHEN GOING OUT OF SERVICE, NS_EMPTY_STR, Unknown, DISABLED
status	<Integer>	Read-only	Current status of the gslb vserver. During the initial phase if the configured lb method is not round robin , the vserver will adopt round robin to

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
			distribute traffic for a predefined number of requests.
lbrrreason	<Integer>	Read-only	Reason why a vserver is in RR. The following are the reasons: 1 - MEP is DOWN (GSLB) 2 - LB method has changed 3 - Bound service's state changed to UP 4 - A new service is bound 5 - Startup RR factor has changed 6 - LB feature is enabled 7 - Load monitor is not active on a service 8 - Vserver is Enabled 9 - SSL feature is Enabled 10 - All bound services have reached threshold. Using effective state

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
			to load balance (GSLB) 11 - Primary state of bound services are not UP. Using effective state to load balance (GSLB) 12 - No LB decision can be made as all bound services have either reached threshold or are not UP (GSLB) 13 - All load monitors are active .
iscname	<String>	Read-only	is cname feature set on vserver. Possible values = ENABLED, DISABLED
sitepersistence	<String>	Read-only	Type of Site Persistence set. Possible values = ConnectionProxy, HTTPRedirect, NONE
totalservices	<Double>	Read-only	Total number of services bound to the vserver.

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
activeservices	<Double>	Read-only	Total number of active services bound to the vserver.
statechangetimesec	<String>	Read-only	Time when last state change happened. Seconds part.
statechangetimemsec	<Double>	Read-only	Time at which last state change happened. Milliseconds part.
tickssincelaststatechange	<Double>	Read-only	Time in 10 millisecond ticks since the last state change.
health	<Double>	Read-only	Health of vserver based on percentage of weights of active svcs/all svcs. This does not consider administratively disabled svcs.
policyname	<String>	Read-only	Name of the policy bound to the GSLB vserver.
priority	<Double>	Read-only	Priority. Minimum value = 1 Maximum value = 2147483647
gotopriorityexpression	<String>	Read-only	Expression specifying the priority of the next policy which will get evaluated if the

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
			current policy rule evaluates to TRUE. o If gotoPriorityExpression is not present or if it is equal to END then the policy bank evaluation ends here o Else if the gotoPriorityExpression is equal to NEXT then the next policy in the priority order is evaluated. o Else gotoPriorityExpression is evaluated. The result of gotoPriorityExpression (which has to be a number) is processed as follows: - An UNDEF event is triggered if . gotoPriorityExpression cannot be evaluated . gotoPriorityExpression evaluates to

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
			number which is smaller than the maximum priority in the policy bank but is not same as any policy's priority . gotoPriorityExpression evaluates to a priority that is smaller than the current policy's priority - If the gotoPriorityExpression evaluates to the priority of the current policy then the next policy in the priority order is evaluated. - If the gotoPriorityExpression evaluates to the priority of a policy further ahead in the list then that policy will be evaluated next. This field is applicable only to rewrite and responder policies.

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
type	<String>	Read-only	The bindpoint to which the policy is bound. Possible values = REQUEST, RESPONSE
vsvrbindsvcip	<String>	Read-only	used for showing the ip of bound entities.
vsvrbindsvcport	<Integer>	Read-only	used for showing ports of bound entities. Range 1 - 65535 * in CLI is represented as 65535 in NITRO API
servername	<String>	Read-only	This field is used to display server name in case of GSLB servicegroup binding to GSLB vserver .
nodefaultbindings	<String>	Read-only	to determine if the configuration will have default ssl CIPHER and ECC curve bindings. Default value: NO Possible values = YES, NO
__count	<Double>	Read-only	count parameter

Operations

(click to see Properties)

ADD| DELETE| UPDATE| UNSET| ENABLE| DISABLE| RENAME| GET (ALL)| GET| COUNT

Some options that you can use for each operations:

- **Getting warnings in response:**NITRO allows you to get warnings in an operation by specifying the "warning" query parameter as "yes". For example, to get warnings while connecting to the Citrix ADC appliance, the URL is as follows:

  http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/login?warning=yes

  If any, the warnings are displayed in the response payload with the HTTP code "209 X-NITRO-WARNING".

- **Authenticated access for individual NITRO operations:**NITRO allows you to logon to the Citrix ADC appliance to perform individual operations. You can use this option instead of creating a NITRO session (using the login object) and then using that session to perform all operations,

  To do this, you must specify the username and password in the request header of the NITRO request as follows:

  X-NITRO-USER:<username>

  X-NITRO-PASS:<password>

  Note: In such cases, make sure that the request header DOES not include the following:

  Cookie:NITRO_AUTH_TOKEN=<tokenvalue>

`*Note:`*

Mandatory parameters are marked in redand placeholder content is marked in <green>.

add

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/gslbvserver

HTTP Method:POST

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Content-Type:application/json

Request Payload:

```
{"gslbvserver":{

<b>"name":<String_value>,

</b><b>"servicetype":<String_value>,

</b>"iptype":<String_value>,

"dnsrecordtype":<String_value>,

"lbmethod":<String_value>,

"backupsessiontimeout":<Double_value>,

"backuplbmethod":<String_value>,

"netmask":<String_value>,

"v6netmasklen":<Double_value>,

"tolerance":<Double_value>,

"persistencetype":<String_value>,

"persistenceid":<Double_value>,

"persistmask":<String_value>,

"v6persistmasklen":<Double_value>,

"timeout":<Double_value>,

"edr":<String_value>,

"ecs":<String_value>,

"ecsaddrvalidation":<String_value>,

"mir":<String_value>,
```

```
"disableprimaryondown":<String_value>,

"dynamicweight":<String_value>,

"state":<String_value>,

"considereffectivestate":<String_value>,

"comment":<String_value>,

"somethod":<String_value>,

"sopersistence":<String_value>,

"sopersistencetimeout":<Double_value>,

"sothreshold":<Double_value>,

"sobackupaction":<String_value>,

"appflowlog":<String_value>

}}
```

Response:

HTTP Status Code on Success: 201 Created HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error

delete

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/gslbvserver/name_value<String>

HTTP Method:DELETE

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue>

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error

update

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/gslbvserver

HTTP Method:PUT

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Content-Type:application/json

Request Payload:

```
{"gslbvserver":{

<b>"name":<String_value>,

</b>"iptype":<String_value>,

"dnsrecordtype":<String_value>,

"backupvserver":<String_value>,

"backupsessiontimeout":<Double_value>,

"lbmethod":<String_value>,

"backuplbmethod":<String_value>,

"netmask":<String_value>,

"v6netmasklen":<Double_value>,

"tolerance":<Double_value>,

"persistencetype":<String_value>,

"persistenceid":<Double_value>,
```

```
"persistmask":<String_value>,

"v6persistmasklen":<Double_value>,

"timeout":<Double_value>,

"edr":<String_value>,

"ecs":<String_value>,

"ecsaddrvalidation":<String_value>,

"mir":<String_value>,

"disableprimaryondown":<String_value>,

"dynamicweight":<String_value>,

"considereffectivestate":<String_value>,

"somethod":<String_value>,

"sopersistence":<String_value>,

"sopersistencetimeout":<Double_value>,

"sothreshold":<Double_value>,

"sobackupaction":<String_value>,

"servicename":<String_value>,

"weight":<Double_value>,

"domainname":<String_value>,

"ttl":<Double_value>,

"backupip":<String_value>,

"cookie_domain":<String_value>,

"cookietimeout":<Double_value>,
```

```
"sitedomainttl":<Double_value>,

"comment":<String_value>,

"appflowlog":<String_value>

}}
```

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error

unset

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/gslbvserver?**action=unset**

HTTP Method:POST

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Content-Type:application/json

Request Payload:

```
{"gslbvserver":{

<b>"name":<String_value>,

</b>"backupvserver":true,

"sothreshold":true,

"iptype":true,

"dnsrecordtype":true,

"backupsessiontimeout":true,
```

```
"lbmethod":true,

"backuplbmethod":true,

"netmask":true,

"v6netmasklen":true,

"tolerance":true,

"persistencetype":true,

"persistenceid":true,

"persistmask":true,

"v6persistmasklen":true,

"timeout":true,

"edr":true,

"ecs":true,

"ecsaddrvalidation":true,

"mir":true,

"disableprimaryondown":true,

"dynamicweight":true,

"considereffectivestate":true,

"somethod":true,

"sopersistence":true,

"sopersistencetimeout":true,

"sobackupaction":true,

"servicename":true,
```

```
"weight":true,

"comment":true,

"appflowlog":true

}}
```

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error

enable

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/gslbvserver?**action=enable**

HTTP Method:POST

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Content-Type:application/json

Request Payload:

```
{"gslbvserver":{

<b>"name":<String_value>

</b>}}
```

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error

disable

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/gslbvserver?**action=disable**

HTTP Method:POST

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Content-Type:application/json

Request Payload:

```
{"gslbvserver":{

<b>"name":<String_value>

</b>}}
```

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error

rename

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/gslbvserver?**action=rename**

HTTP Method:POST

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Content-Type:application/json

Request Payload:

```
{"gslbvserver":{

<b>"name":<String_value>,
```

```
</b><b>"newname":<String_value>

</b>}}
```

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error

get (all)

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/gslbvserver

Query-parameters:

attrs

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/gslbvserver?**attrs=property-name1,property-name2**

Use this query parameter to specify the resource details that you want to retrieve.

filter

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/gslbvserver?**filter=property-name1:property-val1,property-name2:property-val2**

Use this query-parameter to get the filtered set of gslbvserver resources configured on Citrix ADC. Filtering can be done on any of the properties of the resource.

view

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/gslbvserver?**view=summary**

Note: By default, the retrieved results are displayed in detail view (?view=detail).

pagination

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/gslbvserver?**pagesize=#no;pageno=#no**

Use this query-parameter to get the gslbvserver resources in chunks.

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "gslbvserver": [ {

"name":<String_value>,

"servicetype":<String_value>,

"state":<String_value>,

"iptype":<String_value>,

"dnsrecordtype":<String_value>,

"persistencetype":<String_value>,

"persistenceid":<Double_value>,

"lbmethod":<String_value>,

"backuplbmethod":<String_value>,

"tolerance":<Double_value>,

"timeout":<Double_value>,

"curstate":<String_value>,

"netmask":<String_value>,
```

```
"v6netmasklen":<Double_value>,

"persistmask":<String_value>,

"v6persistmasklen":<Double_value>,

"servicename":<String_value>,

"weight":<Double_value>,

"domainname":<String_value>,

"ttl":<Double_value>,

"backupip":<String_value>,

"cookie_domain":<String_value>,

"cookietimeout":<Double_value>,

"sitedomainttl":<Double_value>,

"status":<Integer_value>,

"lbrrreason":<Integer_value>,

"backupvserver":<String_value>,

"backupsessiontimeout":<Double_value>,

"edr":<String_value>,

"ecs":<String_value>,

"ecsaddrvalidation":<String_value>,

"mir":<String_value>,

"disableprimaryondown":<String_value>,

"dynamicweight":<String_value>,

"iscname":<String_value>,
```

```
"sitepersistence":<String_value>,

"considereffectivestate":<String_value>,

"totalservices":<Double_value>,

"activeservices":<Double_value>,

"statechangetimesec":<String_value>,

"statechangetimemsec":<Double_value>,

"tickssincelaststatechange":<Double_value>,

"comment":<String_value>,

"sopersistencetimeout":<Double_value>,

"somethod":<String_value>,

"sobackupaction":<String_value>,

"sopersistence":<String_value>,

"sothreshold":<Double_value>,

"health":<Double_value>,

"appflowlog":<String_value>,

"policyname":<String_value>,

"priority":<Double_value>,

"gotopriorityexpression":<String_value>,

"type":<String_value>,

"vsvrbindsvcip":<String_value>,

"vsvrbindsvcport":<Integer_value>,

"servername":<String_value>,
```

```
"nodefaultbindings":<String_value>

}]}
```

get

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/gslbvserver/name_value<String>

Query-parameters:

attrs

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/gslbvserver/name_value<String>?**attrs=property-name1,property-name2**

Use this query parameter to specify the resource details that you want to retrieve.

view

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/gslbvserver/name_value<String>?**view=summary**

Note: By default, the retrieved results are displayed in detail view (?view=detail).

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "gslbvserver": [ {
```

```
"name":<String_value>,

"servicetype":<String_value>,

"state":<String_value>,

"iptype":<String_value>,

"dnsrecordtype":<String_value>,

"persistencetype":<String_value>,

"persistenceid":<Double_value>,

"lbmethod":<String_value>,

"backuplbmethod":<String_value>,

"tolerance":<Double_value>,

"timeout":<Double_value>,

"curstate":<String_value>,

"netmask":<String_value>,

"v6netmasklen":<Double_value>,

"persistmask":<String_value>,

"v6persistmasklen":<Double_value>,

"servicename":<String_value>,

"weight":<Double_value>,

"domainname":<String_value>,

"ttl":<Double_value>,

"backupip":<String_value>,

"cookie_domain":<String_value>,
```

```
"cookietimeout":<Double_value>,

"sitedomainttl":<Double_value>,

"status":<Integer_value>,

"lbrrreason":<Integer_value>,

"backupvserver":<String_value>,

"backupsessiontimeout":<Double_value>,

"edr":<String_value>,

"ecs":<String_value>,

"ecsaddrvalidation":<String_value>,

"mir":<String_value>,

"disableprimaryondown":<String_value>,

"dynamicweight":<String_value>,

"iscname":<String_value>,

"sitepersistence":<String_value>,

"considereffectivestate":<String_value>,

"totalservices":<Double_value>,

"activeservices":<Double_value>,

"statechangetimesec":<String_value>,

"statechangetimemsec":<Double_value>,

"tickssincelaststatechange":<Double_value>,

"comment":<String_value>,

"sopersistencetimeout":<Double_value>,
```

```
"somethod":<String_value>,

"sobackupaction":<String_value>,

"sopersistence":<String_value>,

"sothreshold":<Double_value>,

"health":<Double_value>,

"appflowlog":<String_value>,

"policyname":<String_value>,

"priority":<Double_value>,

"gotopriorityexpression":<String_value>,

"type":<String_value>,

"vsvrbindsvcip":<String_value>,

"vsvrbindsvcport":<Integer_value>,

"servername":<String_value>,

"nodefaultbindings":<String_value>

}]}
```

count

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/gslbvserver?**count=yes**

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "gslbvserver": [ { "__count": "#no"} ] }
```

gslbvserver_binding

Binding object which returns the resources bound to gslbvserver.

Properties

(click to see Operations)

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
name	<String>	Read-write	Name of the GSLB virtual server. Minimum length = 1
gslbvserver_gslbservicegroup_binding	<gslbvserver_gslbservicegroup_binding[]>	Read-only	gslbservicegroup that can be bound to gslbvserver.
gslbvserver_gslbservicegroupmember_binding	<gslbvserver_gslbservicegroupmember_binding[]>	Read-only	gslbservicegroupmember that can be bound to gslbvserver.
gslbvserver_spillover policy_binding	<gslbvserver_spillover policy_binding[]>	Read-only	spilloverpolicy that can be bound to gslbvserver.
gslbvserver_gslbservice_binding	<gslbvserver_gslbservice_binding[]>	Read-only	gslbservice that can be bound to gslbvserver.

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
gslbvserver_domain_binding	<gslbvserver_domain_binding[]>	Read-only	domain that can be bound to gslbvserver.

Operations

(click to see Properties)

GET| GET (ALL)

Some options that you can use for each operations:

- **Getting warnings in response:**NITRO allows you to get warnings in an operation by specifying the "warning" query parameter as "yes". For example, to get warnings while connecting to the Citrix ADC appliance, the URL is as follows:

  http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/login?warning=yes

  If any, the warnings are displayed in the response payload with the HTTP code "209 X-NITRO-WARNING".

- **Authenticated access for individual NITRO operations:**NITRO allows you to logon to the Citrix ADC appliance to perform individual operations. You can use this option instead of creating a NITRO session (using the login object) and then using that session to perform all operations,

  To do this, you must specify the username and password in the request header of the NITRO request as follows:

  X-NITRO-USER:<username>

  X-NITRO-PASS:<password>

  Note: In such cases, make sure that the request header DOES not include the following:

  Cookie:NITRO_AUTH_TOKEN=<tokenvalue>

`*Note:`*

Mandatory parameters are marked in redand placeholder content is marked in <green>.

get

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/gslbvserver_binding/name_value<String>

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "gslbvserver_binding": [ {

"name":<String_value>,

"gslbvserver_gslbservice_binding":<gslbvserver_gslbservice_binding[]_value
>,

"gslbvserver_spilloverpolicy_binding":<gslbvserver_spilloverpolicy_binding
[]_value>,

"gslbvserver_gslbservicegroupmember_binding":<gslbvserver_gslbservicegroup
member_binding[]_value>,

"gslbvserver_domain_binding":<gslbvserver_domain_binding[]_value>,

"gslbvserver_gslbservicegroup_binding":<gslbvserver_gslbservicegroup_bindi
ng[]_value>

}]}
```

get (all)

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/gslbvserver_binding

Query-parameters:

bulkbindings

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/gslbvserver_binding?**bulkbindings=yes**

NITRO allows you to fetch bindings in bulk.

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "gslbvserver_binding": [ {

"name":<String_value>,

"gslbvserver_gslbservice_binding":<gslbvserver_gslbservice_binding[]_value
>,

"gslbvserver_spilloverpolicy_binding":<gslbvserver_spilloverpolicy_binding
[]_value>,

"gslbvserver_gslbservicegroupmember_binding":<gslbvserver_gslbservicegroup
member_binding[]_value>,

"gslbvserver_domain_binding":<gslbvserver_domain_binding[]_value>,
```

```
"gslbvserver_gslbservicegroup_binding":<gslbvserver_gslbservicegroup_binding[]_value>

}]}
```

gslbvserver_domain_binding

Binding object showing the domain that can be bound to gslbvserver.

Properties

(click to see Operations)

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
backupipflag	<Boolean>	Read-write	The IP address of the backup service for the specified domain name. Used when all the services bound to the domain are down, or when the backup chain of virtual servers is down.
cookietimeout	<Double>	Read-write	Timeout, in minutes, for the GSLB site cookie. Minimum value = 0 Maximum value = 1440
backupip	<String>	Read-write	The IP address of the backup service for the specified domain name. Used when all the services bound to the domain are down, or when

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
			the backup chain of virtual servers is down. Minimum length = 1
name	<String>	Read-write	Name of the virtual server on which to perform the binding operation. Minimum length = 1
ttl	<Double>	Read-write	Time to live (TTL) for the domain. Minimum value = 1
domainname	<String>	Read-write	Domain name for which to change the time to live (TTL) and/or backup service IP address. Minimum length = 1
sitedomainttl	<Double>	Read-write	TTL, in seconds, for all internally created site domains (created when a site prefix is configured on a GSLB service) that are associated with this virtual server. Minimum value = 1
cookie_domainflag	<Boolean>	Read-write	The cookie domain for the GSLB site. Used when inserting the GSLB site cookie in the HTTP response.

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
cookie_domain	<String>	Read-write	The cookie domain for the GSLB site. Used when inserting the GSLB site cookie in the HTTP response. Minimum length = 1
__count	<Double>	Read-write	count parameter

Operations

(click to see Properties)

ADD:| DELETE:| GET| GET (ALL)| COUNT

Some options that you can use for each operations:

- **Getting warnings in response:**NITRO allows you to get warnings in an operation by specifying the "warning" query parameter as "yes". For example, to get warnings while connecting to the Citrix ADC appliance, the URL is as follows:

  http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/login?warning=yes

  If any, the warnings are displayed in the response payload with the HTTP code "209 X-NITRO-WARNING".

- **Authenticated access for individual NITRO operations:**NITRO allows you to logon to the Citrix ADC appliance to perform individual operations. You can use this option instead of creating a NITRO session (using the login object) and then using that session to perform all operations,

  To do this, you must specify the username and password in the request header of the NITRO request as follows:

  X-NITRO-USER:<username>

  X-NITRO-PASS:<password>

  Note: In such cases, make sure that the request header DOES not include the following:

  Cookie:NITRO_AUTH_TOKEN=<tokenvalue>

`*Note:`*

Mandatory parameters are marked in redand placeholder content is marked in <green>.

add

URL:http://<Citrix-ADC-IP-address(NSIP)/nitro/v1/config/gslbvserver_domain_binding

HTTP Method:PUT

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Content-Type:application/json

Request Payload:

```
{
"gslbvserver_domain_binding":{
<b>"name":<String_value>,
</b>"domainname":<String_value>,
"ttl":<Double_value>,
"backupip":<String_value>,
"cookie_domain":<String_value>,
"cookietimeout":<Double_value>,
"sitedomainttl":<Double_value>
}}
```

Response:

HTTP Status Code on Success: 201 Created HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error

delete

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/gslbvserver_domain_binding/name_value<String>

Query-parameters:

args

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/gslbvserver_domain_binding/name_value<String>?**args=domainname:<String_value>,backupipflag:<Boolean_value>,cookie_domainflag:<Boolean_value>**

HTTP Method:DELETE

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue>

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error

get

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/gslbvserver_domain_binding/name_value<String>

Query-parameters:

filter

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/gslbvserver_domain_binding/name_value<String>?**filter=property-name1:property-value1,property-name2:property-value2**

Use this query-parameter to get the filtered set of gslbvserver_domain_binding resources configured on Citrix ADC. Filtering can be done on any of the properties of the resource.

pagination

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/gslbvserver_domain_binding/name_value<String>?**pagesize=#no;pageno=#no**

Use this query-parameter to get the gslbvserver_domain_binding resources in chunks.

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "gslbvserver_domain_binding": [ {

"backupipflag":<Boolean_value>,

"cookietimeout":<Double_value>,

"backupip":<String_value>,

"name":<String_value>,

"ttl":<Double_value>,

"domainname":<String_value>,

"sitedomainttl":<Double_value>,

"cookie_domainflag":<Boolean_value>,

"cookie_domain":<String_value>

}]}
```

get (all)

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/gslbvserver_domain_binding

Query-parameters:

bulkbindings

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/gslbvserver_domain_binding?**bulkbindings=yes**

NITRO allows you to fetch bindings in bulk.

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "gslbvserver_domain_binding": [ {

"backupipflag":<Boolean_value>,

"cookietimeout":<Double_value>,

"backupip":<String_value>,

"name":<String_value>,

"ttl":<Double_value>,

"domainname":<String_value>,

"sitedomainttl":<Double_value>,

"cookie_domainflag":<Boolean_value>,
```

```
"cookie_domain":<String_value>

}]}
```

count

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/gslbvserver_domain_binding/name_value<String>?**count=yes**

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{"gslbvserver_domain_binding": [ { "__count": "#no"} ] }
```

gslbvserver_gslbservice_binding

Binding object showing the gslbservice that can be bound to gslbvserver.

Properties

(click to see Operations)

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
weight	<Double>	Read-write	Weight to assign to the GSLB service.

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
			Minimum value = 1 Maximum value = 100
name	<String>	Read-write	Name of the virtual server on which to perform the binding operation. Minimum length = 1
servicename	<String>	Read-write	Name of the GSLB service for which to change the weight. Minimum length = 1
domainname	<String>	Read-write	Domain name for which to change the time to live (TTL) and/or backup service IP address. Minimum length = 1
cnameentry	<String>	Read-only	The cname of the gslb service.
svcsitepersistence	<String>	Read-only	Type of Site Persistence set on the bound service. Possible values = ConnectionProxy, HTTPRedirect, NONE
gslbboundsvctype	<String>	Read-only	Protocol used by services bound to the GSLBvirtual server. Possible values = HTTP, FTP, TCP,

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
			UDP, SSL, SSL_BRIDGE, SSL_TCP, NNTP, ANY, SIP_UDP, SIP_TCP, SIP_SSL, RADIUS, RDP, RTSP, MYSQL, MSSQL, ORACLE
preferredlocation	<String>	Read-only	The target site to be returned in the DNS response when a policy is successfully evaluated against the incoming DNS request. Target site is specified in dotted notation with up to 6 qualifiers. Wildcard `*' is accepted as a valid qualifier token.
dynamicconfwt	<Double>	Read-only	Weight obtained by the virtue of bound service count or weight.
cumulativeweight	<Double>	Read-only	Cumulative weight is the weight of GSLB service considering both its configured weight and dynamic weight. It is equal to product of dynamic weight and configured weight of the gslb service .

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
gslbthreshold	<Integer>	Read-only	Indicates if gslb svc has reached threshold.
sitepersistcookie	<String>	Read-only	This field is introduced for displaying the cookie in cluster setup. Minimum length = 1
port	<Integer>	Read-only	Port number. Range 1 - 65535 * in CLI is represented as 65535 in NITRO API
iscname	<String>	Read-only	is cname feature set on vserver. Possible values = ENABLED, DISABLED
curstate	<String>	Read-only	State of the gslb vserver. Possible values = UP, DOWN, UNKNOWN, BUSY, OUT OF SERVICE, GOING OUT OF SERVICE, DOWN WHEN GOING OUT OF SERVICE, NS_EMPTY_STR, Unknown, DISABLED

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
svreffgslbstate	<String>	Read-only	Effective state of the gslb svc. Possible values = UP, DOWN, UNKNOWN, BUSY, OUT OF SERVICE, GOING OUT OF SERVICE, DOWN WHEN GOING OUT OF SERVICE, NS_EMPTY_STR, Unknown, DISABLED
thresholdvalue	<Integer>	Read-only	Tells whether threshold exceeded for this service participating in CUSTOMLB.
ipaddress	<String>	Read-only	IP address.
__count	<Double>	Read-only	count parameter

Operations

(click to see Properties)

ADD:| DELETE:| GET| GET (ALL)| COUNT

Some options that you can use for each operations:

- **Getting warnings in response:**NITRO allows you to get warnings in an operation by specifying the "warning" query parameter as "yes". For example, to get warnings while connecting to the Citrix ADC appliance, the URL is as follows:

  http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/login?warning=yes

  If any, the warnings are displayed in the response payload with the HTTP code "209 X-NITRO-WARNING".

- **Authenticated access for individual NITRO operations:**NITRO allows you to logon to the Citrix ADC appliance to perform individual operations. You can use this option instead of creating a NITRO session (using the login object) and then using that session to perform all operations,

  To do this, you must specify the username and password in the request header of the NITRO request as follows:

  X-NITRO-USER:<username>

  X-NITRO-PASS:<password>

  Note: In such cases, make sure that the request header DOES not include the following:

  Cookie:NITRO_AUTH_TOKEN=<tokenvalue>

`*Note:`*

Mandatory parameters are marked in redand placeholder content is marked in <green>.

add

URL:http://<Citrix-ADC-IP-address(NSIP)/nitro/v1/config/gslbvserver_gslbservice_binding

HTTP Method:PUT

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Content-Type:application/json

Request Payload:

```
{

"gslbvserver_gslbservice_binding":{

<b>"name":<String_value>,

</b>"servicename":<String_value>,

"weight":<Double_value>,

"domainname":<String_value>
```

```
}}
```

Response:

HTTP Status Code on Success: 201 Created HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error

delete

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/gslbvserver_gslbservice_binding/name_value<String>

Query-parameters:

args

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/gslbvserver_gslbservice_binding/name_value<String>?**args=servicename:<String_value>,domainname:<String_value>**

HTTP Method:DELETE

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue>

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error

get

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/gslbvserver_gslbservice_binding/name_value<String>

Query-parameters:

filter

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/gslbvserver_gslbservice_binding/name_value<String>?**filter=property-name1:property-value1,property-name2:property-value2**

Use this query-parameter to get the filtered set of gslbvserver_gslbservice_binding resources configured on Citrix ADC. Filtering can be done on any of the properties of the resource.

pagination

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/gslbvserver_gslbservice_binding/name_value<String>?**pagesize=#no;pageno=#no**

Use this query-parameter to get the gslbvserver_gslbservice_binding resources in chunks.

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "gslbvserver_gslbservice_binding": [ {

"weight":<Double_value>,

"name":<String_value>,

"servicename":<String_value>,

"domainname":<String_value>,

"cnameentry":<String_value>,

"svcsitepersistence":<String_value>,

"gslbboundsvctype":<String_value>,
```

```
"preferredlocation":<String_value>,

"dynamicconfwt":<Double_value>,

"cumulativeweight":<Double_value>,

"gslbthreshold":<Integer_value>,

"sitepersistcookie":<String_value>,

"port":<Integer_value>,

"iscname":<String_value>,

"curstate":<String_value>,

"svreffgslbstate":<String_value>,

"thresholdvalue":<Integer_value>,

"ipaddress":<String_value>

}]}
```

get (all)

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/gslbvserver_gslbservice_binding

Query-parameters:

bulkbindings

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/gslbvserver_gslbservice_binding?**bulkbindings=yes**

NITRO allows you to fetch bindings in bulk.

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "gslbvserver_gslbservice_binding": [ {

"weight":<Double_value>,

"name":<String_value>,

"servicename":<String_value>,

"domainname":<String_value>,

"cnameentry":<String_value>,

"svcsitepersistence":<String_value>,

"gslbboundsvctype":<String_value>,

"preferredlocation":<String_value>,

"dynamicconfwt":<Double_value>,

"cumulativeweight":<Double_value>,

"gslbthreshold":<Integer_value>,

"sitepersistcookie":<String_value>,

"port":<Integer_value>,

"iscname":<String_value>,

"curstate":<String_value>,
```

```
"svreffgslbstate":<String_value>,

"thresholdvalue":<Integer_value>,

"ipaddress":<String_value>

}]}
```

count

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/gslbvserver_gslbservice_binding/name_value<String>?**count=yes**

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{"gslbvserver_gslbservice_binding": [ { "__count": "#no"} ] }
```

gslbvserver_spilloverpolicy_binding

Binding object showing the spilloverpolicy that can be bound to gslbvserver.

Properties

(click to see Operations)

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
priority	<Double>	Read-write	Priority. Minimum value = 1 Maximum value = 2147483647
gotopriorityexpression	<String>	Read-write	Expression specifying the priority of the next policy which will get evaluated if the current policy rule evaluates to TRUE. o If gotoPriorityExpression is not present or if it is equal to END then the policy bank evaluation ends here o Else if the gotoPriorityExpression is equal to NEXT then the next policy in the priority order is evaluated. o Else gotoPriorityExpression is evaluated. The result of gotoPriorityExpression (which has to be a number) is processed as follows: - An UNDEF event is triggered if . gotoPriorityExpression cannot be evaluated . gotoPriorityExpression evaluates to number which is smaller than the

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
			maximum priority in the policy bank but is not same as any policy's priority . gotoPriorityExpression evaluates to a priority that is smaller than the current policy's priority - If the gotoPriorityExpression evaluates to the priority of the current policy then the next policy in the priority order is evaluated. - If the gotoPriorityExpression evaluates to the priority of a policy further ahead in the list then that policy will be evaluated next. This field is applicable only to rewrite and responder policies.
policyname	<String>	Read-write	Name of the policy bound to the GSLB vserver.
name	<String>	Read-write	Name of the virtual server on which to perform the binding operation. Minimum length = 1
type	<String>	Read-write	The bindpoint to which the policy is bound.

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
			Possible values = REQUEST, RESPONSE
__count	<Double>	Read-write	count parameter

Operations

(click to see Properties)

ADD:| DELETE:| GET| GET (ALL)| COUNT

Some options that you can use for each operations:

- **Getting warnings in response:**NITRO allows you to get warnings in an operation by specifying the "warning" query parameter as "yes". For example, to get warnings while connecting to the Citrix ADC appliance, the URL is as follows:

  http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/login?warning=yes

  If any, the warnings are displayed in the response payload with the HTTP code "209 X-NITRO-WARNING".

- **Authenticated access for individual NITRO operations:**NITRO allows you to logon to the Citrix ADC appliance to perform individual operations. You can use this option instead of creating a NITRO session (using the login object) and then using that session to perform all operations,

  To do this, you must specify the username and password in the request header of the NITRO request as follows:

  X-NITRO-USER:<username>

  X-NITRO-PASS:<password>

  Note: In such cases, make sure that the request header DOES not include the following:

  Cookie:NITRO_AUTH_TOKEN=<tokenvalue>

`*Note:`*

Mandatory parameters are marked in redand placeholder content is marked in <green>.

add

URL:http://<Citrix-ADC-IP-address(NSIP)/nitro/v1/config/gslbvserver_spilloverpolicy_binding

HTTP Method:PUT

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Content-Type:application/json

Request Payload:

```
{

"gslbvserver_spilloverpolicy_binding":{

<b>"name":<String_value>,

</b>"policyname":<String_value>,

"priority":<Double_value>,

"gotopriorityexpression":<String_value>,

"type":<String_value>

}}
```

Response:

HTTP Status Code on Success: 201 Created HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error

delete

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/gslbvserver_spilloverpolicy_binding/name_value<String>

Query-parameters:

args

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/gslbvserver_spilloverpolicy_binding/name_value<String>?**args=policyname:<String_value>**

HTTP Method:DELETE

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue>

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error

get

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/gslbvserver_spilloverpolicy_binding/name_value<String>

Query-parameters:

filter

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/gslbvserver_spilloverpolicy_binding/name_value<String>?**filter=property-name1:property-value1,property-name2:property-value2**

Use this query-parameter to get the filtered set of gslbvserver_spilloverpolicy_binding resources configured on Citrix ADC. Filtering can be done on any of the properties of the resource.

pagination

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/gslbvserver_spilloverpolicy_binding/name_value<String>?**pagesize=#no;pageno=#no**

Use this query-parameter to get the gslbvserver_spilloverpolicy_binding resources in chunks.

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "gslbvserver_spilloverpolicy_binding": [ {

"priority":<Double_value>,

"gotopriorityexpression":<String_value>,

"policyname":<String_value>,

"name":<String_value>,

"type":<String_value>

}]}
```

get (all)

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/gslbvserver_spilloverpolicy_binding

Query-parameters:

bulkbindings

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/gslbvserver_spilloverpolicy_binding?**bulkbindings=yes**

NITRO allows you to fetch bindings in bulk.

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "gslbvserver_spilloverpolicy_binding": [ {

"priority":<Double_value>,

"gotopriorityexpression":<String_value>,

"policyname":<String_value>,

"name":<String_value>,

"type":<String_value>

}]}
```

count

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/gslbvserver_spilloverpolicy_binding/name_value<String>?**count=yes**

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{"gslbvserver_spilloverpolicy_binding": [ { "__count": "#no"} ] }
```

gslbservicegroup

Configuration for GSLB service group resource.

Properties

(click to see Operations)

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
servicegroupname	<String>	Read-write	Name of the GSLB service group. Must begin with an ASCII alphabetic or underscore (_) character, and must contain only ASCII alphanumeric, underscore, hash (#), period (.), space, colon (:), at (@), equals (=), and hyphen (-) characters. Can be changed after the name is created. Minimum length = 1
servicetype	<String>	Read-write	Protocol used to exchange data with the GSLB service. Possible values = HTTP, FTP, TCP,

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
			UDP, SSL, SSL_BRIDGE, SSL_TCP, NNTP, ANY, SIP_UDP, SIP_TCP, SIP_SSL, RADIUS, RDP, RTSP, MYSQL, MSSQL, ORACLE
maxclient	<Double>	Read-write	Maximum number of simultaneous open connections for the GSLB service group. Minimum value = 0 Maximum value = 4294967294
cip	<String>	Read-write	Insert the Client IP header in requests forwarded to the GSLB service. Possible values = ENABLED, DISABLED
cipheader	<String>	Read-write	Name of the HTTP header whose value must be set to the IP address of the client. Used with the Client IP parameter. If client IP insertion is enabled, and the client IP header is not specified, the value of Client IP Header parameter or the value set by the set ns config

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
			command is used as client's IP header name. Minimum length = 1
healthmonitor	<String>	Read-write	Monitor the health of this GSLB service.Available settings function are as follows: YES - Send probes to check the health of the GSLB service. NO - Do not send probes to check the health of the GSLB service. With the NO option, the appliance shows the service as UP at all times. Default value: YES Possible values = YES, NO
clttimeout	<Double>	Read-write	Time, in seconds, after which to terminate an idle client connection. Minimum value = 0 Maximum value = 31536000

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
svrtimeout	<Double>	Read-write	Time, in seconds, after which to terminate an idle server connection. Minimum value = 0 Maximum value = 31536000
maxbandwidth	<Double>	Read-write	Maximum bandwidth, in Kbps, allocated for all the services in the GSLB service group. Minimum value = 0 Maximum value = 4294967287
monthreshold	<Double>	Read-write	Minimum sum of weights of the monitors that are bound to this GSLB service. Used to determine whether to mark a GSLB service as UP or DOWN. Minimum value = 0 Maximum value = 65535
state	<String>	Read-write	Initial state of the GSLB service group. Default value: ENABLED Possible values = ENABLED, DISABLED

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
downstateflush	<String>	Read-write	Flush all active transactions associated with all the services in the GSLB service group whose state transitions from UP to DOWN. Do not enable this option for applications that must complete their transactions. Default value: ENABLED Possible values = ENABLED, DISABLED
comment	<String>	Read-write	Any information about the GSLB service group.
appflowlog	<String>	Read-write	Enable logging of AppFlow information for the specified GSLB service group. Default value: ENABLED Possible values = ENABLED, DISABLED
autoscale	<String>	Read-write	Auto scale option for a GSLB servicegroup. Default value: DISABLED

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
			Possible values = DISABLED, DNS
sitename	<String>	Read-write	Name of the GSLB site to which the service group belongs. Minimum length = 1
sitepersistence	<String>	Read-write	Use cookie-based site persistence. Applicable only to HTTP and SSL non-autoscale enabled GSLB servicegroups. Possible values = ConnectionProxy, HTTPRedirect, NONE
servername	<String>	Read-write	Name of the server to which to bind the service group. Minimum length = 1
port	<Integer>	Read-write	Server port number. Range 1 - 65535 * in CLI is represented as 65535 in NITRO API
weight	<Double>	Read-write	Weight to assign to the servers in the service group. Specifies the capacity of the servers relative to the other servers in

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
			the load balancing configuration. The higher the weight, the higher the percentage of requests sent to the service. Minimum value = 1 Maximum value = 100
hashid	<Double>	Read-write	The hash identifier for the service. This must be unique for each service. This parameter is used by hash based load balancing methods. Minimum value = 1
publicip	<String>	Read-write	The public IP address that a NAT device translates to the GSLB service's private IP address. Optional. Minimum length = 1
publicport	<Integer>	Read-write	The public port associated with the GSLB service's public IP address. The port is mapped to the service's private port number. Applicable to the local GSLB service. Optional. Minimum value = 1

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
siteprefix	<String>	Read-write	The site's prefix string. When the GSLB service group is bound to a GSLB virtual server, a GSLB site domain is generated internally for each bound serviceitem-domain pair by concatenating the site prefix of the service item and the name of the domain. If the special string NONE is specified, the site-prefix string is unset. When implementing HTTP redirect site persistence, the Citrix ADC redirects GSLB requests to GSLB services by using their site domains.
monitor_name_svc	<String>	Read-write	Name of the monitor bound to the GSLB service group. Used to assign a weight to the monitor. Minimum length = 1
dup_weight	<Double>	Read-write	weight of the monitor that is bound to GSLB servicegroup. Minimum value = 1

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
delay	<Double>	Read-write	The time allowed (in seconds) for a graceful shutdown. During this period, new connections or requests will continue to be sent to this service for clients who already have a persistent session on the system. Connections or requests from fresh or new clients who do not yet have a persistence sessions on the system will not be sent to the service. Instead, they will be load balanced among other available services. After the delay time expires, no new requests or connections will be sent to the service.
graceful	<String>	Read-write	Wait for all existing connections to the service to terminate before shutting down the service. Default value: NO Possible values = YES, NO

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
includemembers	<Boolean>	Read-write	Display the members of the listed GSLB service groups in addition to their settings. Can be specified when no service group name is provided in the command. In that case, the details displayed for each service group are identical to the details displayed when a service group name is provided, except that bound monitors are not displayed.
newname	<String>	Read-write	New name for the GSLB service group. Minimum length = 1
numofconnections	<Integer>	Read-only	This will tell the number of client side connections are still open.
serviceconftype	<Boolean>	Read-only	The configuration type of the GSLB service group.
value	<String>	Read-only	SSL Status. Possible values = Certkey not bound, SSL feature disabled

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
svrstate	<String>	Read-only	The state of the GSLB service. Possible values = UP, DOWN, UNKNOWN, BUSY, OUT OF SERVICE, GOING OUT OF SERVICE, DOWN WHEN GOING OUT OF SERVICE, NS_EMPTY_STR, Unknown, DISABLED
ip	<String>	Read-only	IP Address.
monstatcode	<Integer>	Read-only	The code indicating the monitor response.
monstatparam1	<Integer>	Read-only	First parameter for use with message code.
monstatparam2	<Integer>	Read-only	Second parameter for use with message code.
monstatparam3	<Integer>	Read-only	Third parameter for use with message code.
statechangetimemsec	<Double>	Read-only	Time when last state change occurred. Milliseconds part.
stateupdatereason	<Double>	Read-only	Checks state update reason on the secondary node.

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
clmonowner	<Double>	Read-only	Tells the mon owner of the service.
clmonview	<Double>	Read-only	Tells the view id of the monitoring owner.
groupcount	<Double>	Read-only	Servicegroup Count.
serviceipstr	<String>	Read-only	This field has been intorduced to show the dbs services ip.
servicegroupeffective state	<String>	Read-only	Indicates the effective GSLB servicegroup state based on the state of the bound service items.If all services are UP the effective state is UP, if all are DOWN its DOWN,if all are OFS its OFS.If atleast one serviceis UP and rest are either DOWN or OFS, the effective state is PARTIAL-UP.If atleast one bound service is DOWN and rest are OFS the effective state is PARTIAL DOWN. Possible values = UP, DOWN, OUT OF SERVICE, PARTIAL-UP, PARTIAL-DOWN

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
gslb	<String>	Read-only	. Default value: GSLB Possible values = REMOTE, LOCAL
svreffgslbstate	<String>	Read-only	Effective state of the gslb svc. Possible values = UP, DOWN, UNKNOWN, BUSY, OUT OF SERVICE, GOING OUT OF SERVICE, DOWN WHEN GOING OUT OF SERVICE, NS_EMPTY_STR, Unknown, DISABLED
nodefaultbindings	<String>	Read-only	to determine if the configuration is from stylebooks. Default value: NO Possible values = YES, NO
__count	<Double>	Read-only	count parameter

Operations

(click to see Properties)

ADD| DELETE| UPDATE| UNSET| ENABLE| DISABLE| RENAME| GET (ALL)| GET| COUNT

Some options that you can use for each operations:

- **Getting warnings in response:**NITRO allows you to get warnings in an operation by specifying the "warning" query parameter as "yes". For example, to get warnings while connecting to the Citrix ADC appliance, the URL is as follows:

  http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/login?warning=yes

  If any, the warnings are displayed in the response payload with the HTTP code "209 X-NITRO-WARNING".

- **Authenticated access for individual NITRO operations:**NITRO allows you to logon to the Citrix ADC appliance to perform individual operations. You can use this option instead of creating a NITRO session (using the login object) and then using that session to perform all operations,

  To do this, you must specify the username and password in the request header of the NITRO request as follows:

  X-NITRO-USER:<username>

  X-NITRO-PASS:<password>

  Note: In such cases, make sure that the request header DOES not include the following:

  Cookie:NITRO_AUTH_TOKEN=<tokenvalue>

`*Note:`*

Mandatory parameters are marked in redand placeholder content is marked in <green>.

add

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/gslbservicegroup

HTTP Method:POST

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Content-Type:application/json

Request Payload:

```
{"gslbservicegroup":{
```

```
<b>"servicegroupname":<String_value>,

</b><b>"servicetype":<String_value>,

</b>"maxclient":<Double_value>,

"cip":<String_value>,

"cipheader":<String_value>,

"healthmonitor":<String_value>,

"clttimeout":<Double_value>,

"svrtimeout":<Double_value>,

"maxbandwidth":<Double_value>,

"monthreshold":<Double_value>,

"state":<String_value>,

"downstateflush":<String_value>,

"comment":<String_value>,

"appflowlog":<String_value>,

"autoscale":<String_value>,

<b>"sitename":<String_value>,

</b>"sitepersistence":<String_value>

}}
```

Response:

HTTP Status Code on Success: 201 Created HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error

delete

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/gslbservicegroup/servicegroupname_value<String>

HTTP Method:DELETE

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue>

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error

update

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/gslbservicegroup

HTTP Method:PUT

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Content-Type:application/json

Request Payload:

```
{"gslbservicegroup":{

<b>"servicegroupname":<String_value>,

</b>"servername":<String_value>,

"port":<Integer_value>,

"weight":<Double_value>,

"hashid":<Double_value>,

"publicip":<String_value>,

"publicport":<Integer_value>,
```

```
"siteprefix":<String_value>,

"monitor_name_svc":<String_value>,

"dup_weight":<Double_value>,

"maxclient":<Double_value>,

"healthmonitor":<String_value>,

"cip":<String_value>,

"cipheader":<String_value>,

"clttimeout":<Double_value>,

"svrtimeout":<Double_value>,

"maxbandwidth":<Double_value>,

"monthreshold":<Double_value>,

"downstateflush":<String_value>,

"comment":<String_value>,

"appflowlog":<String_value>,

"sitepersistence":<String_value>

}}
```

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error

unset

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/gslbservicegroup?**action=unset**

HTTP Method:POST

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Content-Type:application/json

Request Payload:

```
{"gslbservicegroup":{

<b>"servicegroupname":<String_value>,

</b>"servername":true,

"port":true,

"weight":true,

"hashid":true,

"publicip":true,

"publicport":true,

"siteprefix":true,

"maxclient":true,

"cip":true,

"clttimeout":true,

"svrtimeout":true,

"maxbandwidth":true,

"monthreshold":true,

"appflowlog":true,

"sitepersistence":true,
```

```
"monitor_name_svc":true,

"dup_weight":true,

"healthmonitor":true,

"cipheader":true,

"downstateflush":true,

"comment":true

}}
```

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error

enable

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/gslbservicegroup?**action=enable**

HTTP Method:POST

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Content-Type:application/json

Request Payload:

```
{"gslbservicegroup":{

<b>"servicegroupname":<String_value>,

</b>"servername":<String_value>,

"port":<Integer_value>
```

```
}}
```

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error

disable

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/gslbservicegroup?**action=disable**

HTTP Method:POST

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Content-Type:application/json

Request Payload:

```
{"gslbservicegroup":{

<b>"servicegroupname":<String_value>,

</b>"servername":<String_value>,

"port":<Integer_value>,

"delay":<Double_value>,

"graceful":<String_value>

}}
```

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error

rename

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/gslbservicegroup?**action=rename**

HTTP Method:POST

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Content-Type:application/json

Request Payload:

```
{"gslbservicegroup":{

<b>"servicegroupname":<String_value>,

</b><b>"newname":<String_value>

</b>}}
```

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error

get (all)

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/gslbservicegroup

Query-parameters:

args

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/gslbservicegroup?**args=servicegroupname:<String_value>,includemembers:<Boolean_value>**

Use this query-parameter to get gslbservicegroup resources based on additional properties.

attrs

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/gslbservicegroup?**attrs=property-name1,property-name2**

Use this query parameter to specify the resource details that you want to retrieve.

filter

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/gslbservicegroup?**filter=property-name1:property-val1,property-name2:property-val2**

Use this query-parameter to get the filtered set of gslbservicegroup resources configured on Citrix ADC. Filtering can be done on any of the properties of the resource.

view

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/gslbservicegroup?**view=summary**

Note: By default, the retrieved results are displayed in detail view (?view=detail).

pagination

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/gslbservicegroup?**pagesize=#no;pageno=#no**

Use this query-parameter to get the gslbservicegroup resources in chunks.

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "gslbservicegroup": [ {
```

```
servicegroupname:<String_value>,includemembers:<Boolean_value>"numofconnec
tions":<Integer_value>,

"servicetype":<String_value>,

"port":<Integer_value>,

"serviceconftype":<Boolean_value>,

"value":<String_value>,

"maxclient":<Double_value>,

"cip":<String_value>,

"cipheader":<String_value>,

"clttimeout":<Double_value>,

"svrtimeout":<Double_value>,

"maxbandwidth":<Double_value>,

"state":<String_value>,

"svrstate":<String_value>,

"delay":<Double_value>,

"ip":<String_value>,

"servername":<String_value>,

"monthreshold":<Double_value>,

"weight":<Double_value>,

"monstatcode":<Integer_value>,

"monstatparam1":<Integer_value>,

"monstatparam2":<Integer_value>,
```

```
"monstatparam3":<Integer_value>,

"downstateflush":<String_value>,

"statechangetimemsec":<Double_value>,

"stateupdatereason":<Double_value>,

"clmonowner":<Double_value>,

"clmonview":<Double_value>,

"groupcount":<Double_value>,

"comment":<String_value>,

"hashid":<Double_value>,

"graceful":<String_value>,

"healthmonitor":<String_value>,

"appflowlog":<String_value>,

"autoscale":<String_value>,

"serviceipstr":<String_value>,

"servicegroupeffectivestate":<String_value>,

"sitename":<String_value>,

"gslb":<String_value>,

"publicip":<String_value>,

"publicport":<Integer_value>,

"svreffgslbstate":<String_value>,

"sitepersistence":<String_value>,

"siteprefix":<String_value>,
```

```
"nodefaultbindings":<String_value>

}]}
```

get

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/gslbservicegroup/servicegroupname_value<String>

Query-parameters:

attrs

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/gslbservicegroup/servicegroupname_value<String>?**attrs=property-name1,property-name2**

Use this query parameter to specify the resource details that you want to retrieve.

view

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/gslbservicegroup/servicegroupname_value<String>?**view=summary**

Note: By default, the retrieved results are displayed in detail view (?view=detail).

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "gslbservicegroup": [ {

servicegroupname:<String_value>,includemembers:<Boolean_value>"numofconnec
tions":<Integer_value>,

"servicetype":<String_value>,

"port":<Integer_value>,

"serviceconftype":<Boolean_value>,

"value":<String_value>,

"maxclient":<Double_value>,

"cip":<String_value>,

"cipheader":<String_value>,

"clttimeout":<Double_value>,

"svrtimeout":<Double_value>,

"maxbandwidth":<Double_value>,

"state":<String_value>,

"svrstate":<String_value>,

"delay":<Double_value>,

"ip":<String_value>,

"servername":<String_value>,

"monthreshold":<Double_value>,

"weight":<Double_value>,

"monstatcode":<Integer_value>,
```

```
"monstatparam1":<Integer_value>,

"monstatparam2":<Integer_value>,

"monstatparam3":<Integer_value>,

"downstateflush":<String_value>,

"statechangetimemsec":<Double_value>,

"stateupdatereason":<Double_value>,

"clmonowner":<Double_value>,

"clmonview":<Double_value>,

"groupcount":<Double_value>,

"comment":<String_value>,

"hashid":<Double_value>,

"graceful":<String_value>,

"healthmonitor":<String_value>,

"appflowlog":<String_value>,

"autoscale":<String_value>,

"serviceipstr":<String_value>,

"servicegroupeffectivestate":<String_value>,

"sitename":<String_value>,

"gslb":<String_value>,

"publicip":<String_value>,

"publicport":<Integer_value>,

"svreffgslbstate":<String_value>,
```

```
"sitepersistence":<String_value>,

"siteprefix":<String_value>,

"nodefaultbindings":<String_value>

}]}
```

count

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/gslbservicegroup?**count=yes**

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "gslbservicegroup": [ { "__count": "#no"} ] }
```

gslbservicegroup_binding

Binding object which returns the resources bound to gslbservicegroup.

Properties

(click to see Operations)

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
servicegroupname	<String>	Read-write	Name of the GSLB service group. Minimum length = 1
gslbservicegroup_servicegroupentitymonbindings_binding	<gslbservicegroup_servicegroupentitymonbindings_binding[]>	Read-only	servicegroupentitymonbindings that can be bound to gslbservicegroup.
gslbservicegroup_lbmonitor_binding	<gslbservicegroup_lbmonitor_binding[]>	Read-only	lbmonitor that can be bound to gslbservicegroup.
gslbservicegroup_gslbservicegroupmember_binding	<gslbservicegroup_gslbservicegroupmember_binding[]>	Read-only	gslbservicegroupmember that can be bound to gslbservicegroup.

Operations

(click to see Properties)

GET| GET (ALL)

Some options that you can use for each operations:

- **Getting warnings in response:**NITRO allows you to get warnings in an operation by specifying the "warning" query parameter as "yes". For example, to get warnings while connecting to the Citrix ADC appliance, the URL is as follows:

  http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/login?warning=yes

  If any, the warnings are displayed in the response payload with the HTTP code "209 X-NITRO-WARNING".

- **Authenticated access for individual NITRO operations:**NITRO allows you to logon to the Citrix ADC appliance to perform individual operations. You can use this option instead of creating a NITRO session (using the login object) and then using that session to perform all operations,

  To do this, you must specify the username and password in the request header of the NITRO request as follows:

X-NITRO-USER:<username>

X-NITRO-PASS:<password>

Note: In such cases, make sure that the request header DOES not include the following:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue>

`*Note:`*

Mandatory parameters are marked in redand placeholder content is marked in <green>.

get

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/gslbservicegroup_binding/servicegroupname_value<String>

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "gslbservicegroup_binding": [ {

"servicegroupname":<String_value>,

"gslbservicegroup_servicegroupentitymonbindings_binding":<gslbservicegroup
_servicegroupentitymonbindings_binding[]_value>,

"gslbservicegroup_lbmonitor_binding":<gslbservicegroup_lbmonitor_binding[]
_value>,
```

```
"gslbservicegroup_gslbservicegroupmember_binding":<gslbservicegroup_gslbse
rvicegroupmember_binding[]_value>

}]}
```

get (all)

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/gslbservicegroup_binding

Query-parameters:

bulkbindings

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/gslbservicegroup_binding?**bulkbindings=yes**

NITRO allows you to fetch bindings in bulk.

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "gslbservicegroup_binding": [ {

"servicegroupname":<String_value>,

"gslbservicegroup_servicegroupentitymonbindings_binding":<gslbservicegroup
_servicegroupentitymonbindings_binding[]_value>,

"gslbservicegroup_lbmonitor_binding":<gslbservicegroup_lbmonitor_binding[]
```

```
_value>,

"gslbservicegroup_gslbservicegroupmember_binding":<gslbservicegroup_gslbse
rvicegroupmember_binding[]_value>

}]}
```

gslbservicegroup_gslbservicegroupmember_binding

Binding object showing the gslbservicegroupmember that can be bound to gslbservicegroup.

Properties

(click to see Operations)

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
state	<String>	Read-write	Initial state of the GSLB service group. Default value: ENABLED Possible values = ENABLED, DISABLED
publicport	<Integer>	Read-write	The public port associated with the GSLB service's public IP address. The port is mapped to the service's private port number. Applicable to the local GSLB service. Optional. Minimum value = 1
weight	<Double>	Read-write	Weight to assign to the servers in the service group. Specifies the

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
			capacity of the servers relative to the other servers in the load balancing configuration. The higher the weight, the higher the percentage of requests sent to the service. Minimum value = 1 Maximum value = 100
port	<Integer>	Read-write	Server port number. Range 1 - 65535 * in CLI is represented as 65535 in NITRO API
servername	<String>	Read-write	Name of the server to which to bind the service group. Minimum length = 1
siteprefix	<String>	Read-write	The site's prefix string. When the GSLB service group is bound to a GSLB virtual server, a GSLB site domain is generated internally for each bound serviceitem-domain pair by concatenating the site prefix of the service item and the

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
			name of the domain. If the special string NONE is specified, the site-prefix string is unset. When implementing HTTP redirect site persistence, the Citrix ADC redirects GSLB requests to GSLB services by using their site domains.
ip	<String>	Read-write	IP Address.
servicegroupname	<String>	Read-write	Name of the GSLB service group. Minimum length = 1
hashid	<Double>	Read-write	The hash identifier for the service. This must be unique for each service. This parameter is used by hash based load balancing methods. Minimum value = 1
publicip	<String>	Read-write	The public IP address that a NAT device translates to the GSLB service's private IP address. Optional. Minimum length = 1
threshold	<String>	Read-only	. Possible values = ABOVE, BELOW

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
delay	<Double>	Read-only	The time allowed (in seconds) for a graceful shutdown. During this period, new connections or requests will continue to be sent to this service for clients who already have a persistent session on the system. Connections or requests from fresh or new clients who do not yet have a persistence sessions on the system will not be sent to the service. Instead, they will be load balanced among other available services. After the delay time expires, no new requests or connections will be sent to the service.
statechangetimesec	<String>	Read-only	Time when last state change occurred. Seconds part.
preferredlocation	<String>	Read-only	Prefered location.
svrstate	<String>	Read-only	The state of the GSLB service. Possible values = UP, DOWN, UNKNOWN, BUSY,

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
			OUT OF SERVICE, GOING OUT OF SERVICE, DOWN WHEN GOING OUT OF SERVICE, NS_EMPTY_STR, Unknown, DISABLED
gslbthreshold	<Integer>	Read-only	Indicates if gslb svc has reached threshold.
tickssincelaststatechange	<Double>	Read-only	Time in 10 millisecond ticks since the last state change.
graceful	<String>	Read-only	Wait for all existing connections to the service to terminate before shutting down the service. Default value: NO Possible values = YES, NO
__count	<Double>	Read-only	count parameter

Operations

(click to see Properties)

ADD:| DELETE:| GET| GET (ALL)| COUNT

Some options that you can use for each operations:

- **Getting warnings in response:**NITRO allows you to get warnings in an operation by specifying the "warning" query parameter as "yes". For example, to get warnings while connecting to the Citrix ADC appliance, the URL is as follows:

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/login?warning=yes

If any, the warnings are displayed in the response payload with the HTTP code "209 X-NITRO-WARNING".

- **Authenticated access for individual NITRO operations:**NITRO allows you to logon to the Citrix ADC appliance to perform individual operations. You can use this option instead of creating a NITRO session (using the login object) and then using that session to perform all operations,

  To do this, you must specify the username and password in the request header of the NITRO request as follows:

  X-NITRO-USER:<username>

  X-NITRO-PASS:<password>

  Note: In such cases, make sure that the request header DOES not include the following:

  Cookie:NITRO_AUTH_TOKEN=<tokenvalue>

`*Note:`*

Mandatory parameters are marked in redand placeholder content is marked in <green>.

add

URL:http://<Citrix-ADC-IP-address(NSIP)/nitro/v1/config/gslbservicegroup_gslbservicegroupmember_binding

HTTP Method:PUT

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Content-Type:application/json

Request Payload:

```
{

"gslbservicegroup_gslbservicegroupmember_binding":{

<b>"servicegroupname":<String_value>,
```

```
</b>"ip":<String_value>,

"servername":<String_value>,

"port":<Integer_value>,

"weight":<Double_value>,

"state":<String_value>,

"hashid":<Double_value>,

"publicip":<String_value>,

"publicport":<Integer_value>,

"siteprefix":<String_value>

}}
```

Response:

HTTP Status Code on Success: 201 Created HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error

delete

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/gslbservicegroup_gslbservicegroupmember_binding/servicegroupname_value<String>

Query-parameters:

args

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/gslbservicegroup_gslbservicegroupmember_binding/servicegroupname_value<String>?**args=ip:<String_value>,servername:<String_value>,port:<Integer_value>**

HTTP Method:DELETE

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue>

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error

get

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/gslbservicegroup_gslbservicegroupmember_binding/servicegroupname_value<String>

Query-parameters:

filter

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/gslbservicegroup_gslbservicegroupmember_binding/servicegroupname_value<String>?**filter=property-name1:property-value1,property-name2:property-value2**

Use this query-parameter to get the filtered set of gslbservicegroup_gslbservicegroupmember_binding resources configured on Citrix ADC. Filtering can be done on any of the properties of the resource.

pagination

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/gslbservicegroup_gslbservicegroupmember_binding/servicegroupname_value<String>?**pagesize=#no;pageno=#no**

Use this query-parameter to get the gslbservicegroup_gslbservicegroupmember_binding resources in chunks.

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "gslbservicegroup_gslbservicegroupmember_binding": [ {

"state":<String_value>,

"publicport":<Integer_value>,

"weight":<Double_value>,

"port":<Integer_value>,

"servername":<String_value>,

"siteprefix":<String_value>,

"ip":<String_value>,

"servicegroupname":<String_value>,

"hashid":<Double_value>,

"publicip":<String_value>,

"threshold":<String_value>,

"delay":<Double_value>,

"statechangetimesec":<String_value>,

"preferredlocation":<String_value>,

"svrstate":<String_value>,

"gslbthreshold":<Integer_value>,

"tickssincelaststatechange":<Double_value>,

"graceful":<String_value>
```

```
}]}
```

get (all)

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/gslbservicegroup_gslbservicegroupmember_binding

Query-parameters:

bulkbindings

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/gslbservicegroup_gslbservicegroupmember_binding?**bulkbindings=yes**

NITRO allows you to fetch bindings in bulk.

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "gslbservicegroup_gslbservicegroupmember_binding": [ {

"state":<String_value>,

"publicport":<Integer_value>,

"weight":<Double_value>,

"port":<Integer_value>,
```

```
"servername":<String_value>,

"siteprefix":<String_value>,

"ip":<String_value>,

"servicegroupname":<String_value>,

"hashid":<Double_value>,

"publicip":<String_value>,

"threshold":<String_value>,

"delay":<Double_value>,

"statechangetimesec":<String_value>,

"preferredlocation":<String_value>,

"svrstate":<String_value>,

"gslbthreshold":<Integer_value>,

"tickssincelaststatechange":<Double_value>,

"graceful":<String_value>

}]}
```

count

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/gslbservicegroup_gslbservicegroupmember_binding/servicegroupname_value<String>?**count=yes**

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{"gslbservicegroup_gslbservicegroupmember_binding": [ { "__count":
"#no"} ] }
```

gslbservicegroup_lbmonitor_binding

Binding object showing the lbmonitor that can be bound to gslbservicegroup.

Properties

(click to see Operations)

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
state	<String>	Read-write	Initial state of the service after binding. Default value: ENABLED Possible values = ENABLED, DISABLED
publicport	<Integer>	Read-write	The public port associated with the GSLB service's public IP address. The port is mapped to the service's private port number. Applicable to the

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
			local GSLB service. Optional.
weight	<Double>	Read-write	Weight to assign to the servers in the service group. Specifies the capacity of the servers relative to the other servers in the load balancing configuration. The higher the weight, the higher the percentage of requests sent to the service. Minimum value = 1 Maximum value = 100
port	<Integer>	Read-write	Port number of the GSLB service. Each service must have a unique port number. Range 1 - 65535 * in CLI is represented as 65535 in NITRO API
siteprefix	<String>	Read-write	The site's prefix string. When the GSLB service group is bound to a GSLB virtual server, a GSLB site domain is generated internally for each bound

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
			serviceitem-domain pair by concatenating the site prefix of the service item and the name of the domain. If the special string NONE is specified, the site-prefix string is unset. When implementing HTTP redirect site persistence, the Citrix ADC redirects GSLB requests to GSLB services by using their site domains.
passive	<Boolean>	Read-write	Indicates if load monitor is passive. A passive load monitor does not remove service from LB decision when threshold is breached.
monitor_name	<String>	Read-write	Monitor name.
monstate	<String>	Read-write	Monitor state. Possible values = ENABLED, DISABLED
servicegroupname	<String>	Read-write	Name of the GSLB service group. Minimum length = 1
hashid	<Double>	Read-write	Unique numerical identifier used by

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
			hash based load balancing methods to identify a service. Minimum value = 1
publicip	<String>	Read-write	The public IP address that a NAT device translates to the GSLB service's private IP address. Optional.
monweight	<Double>	Read-only	weight of the monitor that is bound to GSLB servicegroup.
__count	<Double>	Read-only	count parameter

Operations

(click to see Properties)

ADD:| DELETE:| GET| GET (ALL)| COUNT

Some options that you can use for each operations:

- **Getting warnings in response:**NITRO allows you to get warnings in an operation by specifying the "warning" query parameter as "yes". For example, to get warnings while connecting to the Citrix ADC appliance, the URL is as follows:

  http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/login?warning=yes

  If any, the warnings are displayed in the response payload with the HTTP code "209 X-NITRO-WARNING".

- **Authenticated access for individual NITRO operations:**NITRO allows you to logon to the Citrix ADC appliance to perform individual operations. You can use this option instead of creating a NITRO session (using the login object) and then using that session to perform all operations,

  To do this, you must specify the username and password in the request header of the NITRO request as follows:

X-NITRO-USER:<username>

X-NITRO-PASS:<password>

Note: In such cases, make sure that the request header DOES not include the following:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue>

`*Note:`*

Mandatory parameters are marked in redand placeholder content is marked in <green>.

add

URL:http://<Citrix-ADC-IP-address(NSIP)/nitro/v1/config/gslbservicegroup_lbmonitor_binding

HTTP Method:PUT

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Content-Type:application/json

Request Payload:

```
{
"gslbservicegroup_lbmonitor_binding":{
<b>"servicegroupname":<String_value>,
</b>"port":<Integer_value>,
"monitor_name":<String_value>,
"monstate":<String_value>,
"passive":<Boolean_value>,
"weight":<Double_value>,
"state":<String_value>,
"hashid":<Double_value>,
```

```
"publicip":<String_value>,

"publicport":<Integer_value>,

"siteprefix":<String_value>

}}
```

Response:

HTTP Status Code on Success: 201 Created HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error

delete

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/gslbservicegroup_lbmonitor_binding/servicegroupname_value<String>

Query-parameters:

args

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/gslbservicegroup_lbmonitor_binding/servicegroupname_value<String>?**args=port:<Integer_value>,monitor_name:<String_value>**

HTTP Method:DELETE

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue>

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error

get

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/gslbservicegroup_lbmonitor_binding/servicegroupname_value<String>

Query-parameters:

filter

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/gslbservicegroup_lbmonitor_binding/servicegroupname_value<String>?**filter=property-name1:property-value1,property-name2:property-value2**

Use this query-parameter to get the filtered set of gslbservicegroup_lbmonitor_binding resources configured on Citrix ADC. Filtering can be done on any of the properties of the resource.

pagination

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/gslbservicegroup_lbmonitor_binding/servicegroupname_value<String>?**pagesize=#no;pageno=#no**

Use this query-parameter to get the gslbservicegroup_lbmonitor_binding resources in chunks.

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "gslbservicegroup_lbmonitor_binding": [ {

"state":<String_value>,

"publicport":<Integer_value>,

"weight":<Double_value>,
```

```
"port":<Integer_value>,

"siteprefix":<String_value>,

"passive":<Boolean_value>,

"monitor_name":<String_value>,

"monstate":<String_value>,

"servicegroupname":<String_value>,

"hashid":<Double_value>,

"publicip":<String_value>,

"monweight":<Double_value>

}]}
```

get (all)

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/gslbservicegroup_lbmonitor_binding

Query-parameters:

bulkbindings

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/gslbservicegroup_lbmonitor_binding?**bulkbindings=yes**

NITRO allows you to fetch bindings in bulk.

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "gslbservicegroup_lbmonitor_binding": [ {

"state":<String_value>,

"publicport":<Integer_value>,

"weight":<Double_value>,

"port":<Integer_value>,

"siteprefix":<String_value>,

"passive":<Boolean_value>,

"monitor_name":<String_value>,

"monstate":<String_value>,

"servicegroupname":<String_value>,

"hashid":<Double_value>,

"publicip":<String_value>,

"monweight":<Double_value>

}]}
```

count

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/gslbservicegroup_lbmonitor_binding/servicegroupname_value<String>?**count=yes**

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{"gslbservicegroup_lbmonitor_binding": [ { "__count": "#no"} ] }
```

gslbservicegroup_servicegroupentitymonbindings_binding

Binding object showing the servicegroupentitymonbindings that can be bound to gslbservicegroup.

Properties

(click to see Operations)

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
servicegroupentname2	<String>	Read-write	.
state	<String>	Read-write	Initial state of the service after binding. Default value: ENABLED Possible values = ENABLED, DISABLED

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
publicport	<Integer>	Read-write	The public port associated with the GSLB service's public IP address. The port is mapped to the service's private port number. Applicable to the local GSLB service. Optional.
weight	<Double>	Read-write	When used along with monitor name, it specifies the weight of the monitor binding. When used along with servername ; port pair, it specifies the weight of this GSLB service . Default value: 1 Minimum value = 1 Maximum value = 100
port	<Integer>	Read-write	Port number of the GSLB service. Each service must have a unique port number. Range 1 - 65535 * in CLI is represented as 65535 in NITRO API

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
siteprefix	<String>	Read-write	The site's prefix string. When the GSLB service group is bound to a GSLB virtual server, a GSLB site domain is generated internally for each bound serviceitem-domain pair by concatenating the site prefix of the service item and the name of the domain. If the special string NONE is specified, the site-prefix string is unset. When implementing HTTP redirect site persistence, the Citrix ADC redirects GSLB requests to GSLB services by using their site domains.
passive	<Boolean>	Read-write	Indicates if load monitor is passive. A passive load monitor does not remove service from LB decision when threshold is breached.
monitor_name	<String>	Read-write	Monitor name.
servicegroupname	<String>	Read-write	Name of the GSLB service group.

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
			Minimum length = 1
hashid	<Double>	Read-write	Unique numerical identifier used by hash based load balancing methods to identify a service. Minimum value = 1
publicip	<String>	Read-write	The public IP address that a NAT device translates to the GSLB service's private IP address. Optional.
lastresponse	<String>	Read-only	The string form of monstatcode.
monitor_state	<String>	Read-only	The running state of the monitor on this service. Possible values = UP, DOWN, UNKNOWN, BUSY, OUT OF SERVICE, GOING OUT OF SERVICE, DOWN WHEN GOING OUT OF SERVICE, NS_EMPTY_STR, Unknown, DISABLED
monitortotalprobes	<Double>	Read-only	Total number of probes sent to monitor this service.

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
monitortotalfailedprobes	<Double>	Read-only	Total number of failed probes.
monitorcurrentfailedprobes	<Double>	Read-only	Total number of currently failed probes.
__count	<Double>	Read-only	count parameter

Operations

(click to see Properties)

GET| GET (ALL)| COUNT

Some options that you can use for each operations:

- **Getting warnings in response:**NITRO allows you to get warnings in an operation by specifying the "warning" query parameter as "yes". For example, to get warnings while connecting to the Citrix ADC appliance, the URL is as follows:

  http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/login?warning=yes

  If any, the warnings are displayed in the response payload with the HTTP code "209 X-NITRO-WARNING".

- **Authenticated access for individual NITRO operations:**NITRO allows you to logon to the Citrix ADC appliance to perform individual operations. You can use this option instead of creating a NITRO session (using the login object) and then using that session to perform all operations,

  To do this, you must specify the username and password in the request header of the NITRO request as follows:

  X-NITRO-USER:<username>

  X-NITRO-PASS:<password>

  Note: In such cases, make sure that the request header DOES not include the following:

  Cookie:NITRO_AUTH_TOKEN=<tokenvalue>

`*Note:`*

Mandatory parameters are marked in redand placeholder content is marked in <green>.

get

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/gslbservicegroup_servicegroupentitymonbindings_binding/servicegroupname_value<String>

Query-parameters:

filter

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/gslbservicegroup_servicegroupentitymonbindings_binding/servicegroupname_value<String>?**filter=property-name1:property-value1,property-name2:property-value2**

Use this query-parameter to get the filtered set of gslbservicegroup_servicegroupentitymonbindings_binding resources configured on Citrix ADC. Filtering can be done on any of the properties of the resource.

pagination

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/gslbservicegroup_servicegroupentitymonbindings_binding/servicegroupname_value<String>?**pagesize=#no;pageno=#no**

Use this query-parameter to get the gslbservicegroup_servicegroupentitymonbindings_binding resources in chunks.

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "gslbservicegroup_servicegroupentitymonbindings_binding": [ {
"servicegroupentname2":<String_value>,
"state":<String_value>,
"publicport":<Integer_value>,
"weight":<Double_value>,
"port":<Integer_value>,
"siteprefix":<String_value>,
"passive":<Boolean_value>,
"monitor_name":<String_value>,
"servicegroupname":<String_value>,
"hashid":<Double_value>,
"publicip":<String_value>,
"lastresponse":<String_value>,
"monitor_state":<String_value>,
"monitortotalprobes":<Double_value>,
"monitortotalfailedprobes":<Double_value>,
"monitorcurrentfailedprobes":<Double_value>
}]}
```

get (all)

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/gslbservicegroup_servicegroupentitymonbindings_binding

Query-parameters:

bulkbindings

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/gslbservicegroup_servicegroupentitymonbindings_binding?**bulkbindings=yes**

NITRO allows you to fetch bindings in bulk.

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "gslbservicegroup_servicegroupentitymonbindings_binding": [ {

"servicegroupentname2":<String_value>,

"state":<String_value>,

"publicport":<Integer_value>,

"weight":<Double_value>,

"port":<Integer_value>,

"siteprefix":<String_value>,

"passive":<Boolean_value>,

"monitor_name":<String_value>,
```

```
"servicegroupname":<String_value>,

"hashid":<Double_value>,

"publicip":<String_value>,

"lastresponse":<String_value>,

"monitor_state":<String_value>,

"monitortotalprobes":<Double_value>,

"monitortotalfailedprobes":<Double_value>,

"monitorcurrentfailedprobes":<Double_value>

}]}
```

count

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/gslbservicegroup_servicegroupentitymonbindings_binding/servicegroupname_value<String>?**count=yes**

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{"gslbservicegroup_servicegroupentitymonbindings_binding": [ { "__count":
"#no"} ] }
```


gslbsite_gslbservicegroup_binding

Binding object showing the gslbservicegroup that can be bound to gslbsite.

Properties

(click to see Operations)

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
sitename	<String>	Read-write	Name of the GSLB site. If you specify a site name, details of all the site's constituent services are also displayed. Minimum length = 1
servicegroupname	<String>	Read-write	Servicegroup name.
servicetype	<String>	Read-only	Service type. Possible values = HTTP, FTP, TCP, UDP, SSL, SSL_BRIDGE, SSL_TCP, DTLS, NNTP, DNS, DHCPRA, ANY, SIP_UDP, SIP_TCP, SIP_SSL, DNS_TCP, RTSP, PUSH, SSL_PUSH, RADIUS, RDP, MYSQL, MSSQL, DIAMETER,

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
			SSL_DIAMETER, TFTP, ORACLE, SMPP, SYSLOGTCP, SYSLOGUDP, FIX, SSL_FIX, PROXY, USER_TCP, USER_SSL_TCP, QUIC, IPFIX, LOGSTREAM
__count	<Double>	Read-only	count parameter

Operations

(click to see Properties)

GET| GET (ALL)| COUNT

Some options that you can use for each operations:

- **Getting warnings in response:**NITRO allows you to get warnings in an operation by specifying the "warning" query parameter as "yes". For example, to get warnings while connecting to the Citrix ADC appliance, the URL is as follows:

  http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/login?warning=yes

  If any, the warnings are displayed in the response payload with the HTTP code "209 X-NITRO-WARNING".

- **Authenticated access for individual NITRO operations:**NITRO allows you to logon to the Citrix ADC appliance to perform individual operations. You can use this option instead of creating a NITRO session (using the login object) and then using that session to perform all operations,

  To do this, you must specify the username and password in the request header of the NITRO request as follows:

  X-NITRO-USER:<username>

  X-NITRO-PASS:<password>

Note: In such cases, make sure that the request header DOES not include the following:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue>

`*Note:`*

Mandatory parameters are marked in redand placeholder content is marked in <green>.

get

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/gslbsite_gslbservicegroup_binding/sitename_value<String>

Query-parameters:

filter

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/gslbsite_gslbservicegroup_binding/sitename_value<String>?**filter=property-name1:property-value1,property-name2:property-value2**

Use this query-parameter to get the filtered set of gslbsite_gslbservicegroup_binding resources configured on Citrix ADC. Filtering can be done on any of the properties of the resource.

pagination

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/gslbsite_gslbservicegroup_binding/sitename_value<String>?**pagesize=#no;pageno=#no**

Use this query-parameter to get the gslbsite_gslbservicegroup_binding resources in chunks.

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "gslbsite_gslbservicegroup_binding": [ {

"sitename":<String_value>,

"servicegroupname":<String_value>,

"servicetype":<String_value>

}]}
```

get (all)

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/gslbsite_gslbservicegroup_binding

Query-parameters:

bulkbindings

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/gslbsite_gslbservicegroup_binding?**bulkbindings=yes**

NITRO allows you to fetch bindings in bulk.

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "gslbsite_gslbservicegroup_binding": [ {

"sitename":<String_value>,

"servicegroupname":<String_value>,

"servicetype":<String_value>

}]}
```

count

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/gslbsite_gslbservicegroup_binding/sitename_value<String>?**count=yes**

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{"gslbsite_gslbservicegroup_binding": [ { "__count": "#no"} ] }
```

gslbsite_gslbservicegroupmember_binding

Binding object showing the gslbservicegroupmember that can be bound to gslbsite.

Properties

(click to see Operations)

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
sitename	<String>	Read-write	Name of the GSLB site. If you specify a site name, details of all the site's constituent services are also displayed. Minimum length = 1
servicegroupname	<String>	Read-write	Servicegroup name.
state	<String>	Read-only	State of the gslb service. Possible values = UP, DOWN, UNKNOWN, BUSY, OUT OF SERVICE, GOING OUT OF SERVICE, DOWN WHEN GOING OUT OF SERVICE, NS_EMPTY_STR, Unknown, DISABLED
servicetype	<String>	Read-only	Service type. Possible values = HTTP, FTP, TCP, UDP, SSL, SSL_BRIDGE, SSL_TCP, DTLS, NNTP, DNS, DHCPRA, ANY, SIP_UDP, SIP_TCP, SIP_SSL,

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
			DNS_TCP, RTSP, PUSH, SSL_PUSH, RADIUS, RDP, MYSQL, MSSQL, DIAMETER, SSL_DIAMETER, TFTP, ORACLE, SMPP, SYSLOGTCP, SYSLOGUDP, FIX, SSL_FIX, PROXY, USER_TCP, USER_SSL_TCP, QUIC, IPFIX, LOGSTREAM
port	<Integer>	Read-only	Port number of the gslb service. Range 1 - 65535 * in CLI is represented as 65535 in NITRO API
ipaddress	<String>	Read-only	IP Address of the gslb service.
__count	<Double>	Read-only	count parameter

Operations

(click to see Properties)

GET| GET (ALL)| COUNT

Some options that you can use for each operations:

- **Getting warnings in response:**NITRO allows you to get warnings in an operation by specifying the "warning" query parameter as "yes". For example, to get warnings while connecting to the Citrix ADC appliance, the URL is as follows:

  http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/login?warning=yes

  If any, the warnings are displayed in the response payload with the HTTP code "209 X-NITRO-WARNING".

- **Authenticated access for individual NITRO operations:**NITRO allows you to logon to the Citrix ADC appliance to perform individual operations. You can use this option instead of creating a NITRO session (using the login object) and then using that session to perform all operations,

  To do this, you must specify the username and password in the request header of the NITRO request as follows:

  X-NITRO-USER:<username>

  X-NITRO-PASS:<password>

  Note: In such cases, make sure that the request header DOES not include the following:

  Cookie:NITRO_AUTH_TOKEN=<tokenvalue>

`*Note:`*

Mandatory parameters are marked in redand placeholder content is marked in <green>.

get

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/gslbsite_gslbservicegroupmember_binding/sitename_value<String>

Query-parameters:

filter

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/gslbsite_gslbservicegroupmember_binding/sitename_value<String>?**filter=property-name1:property-value1,property-name2:property-value2**

Use this query-parameter to get the filtered set of gslbsite_gslbservicegroupmember_binding resources configured on Citrix ADC. Filtering can be done on any of the properties of the resource.

pagination

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/gslbsite_gslbservicegroupmember_binding/sitename_value<String>?**pagesize=#no;pageno=#no**

Use this query-parameter to get the gslbsite_gslbservicegroupmember_binding resources in chunks.

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "gslbsite_gslbservicegroupmember_binding": [ {

"sitename":<String_value>,

"servicegroupname":<String_value>,

"state":<String_value>,

"servicetype":<String_value>,

"port":<Integer_value>,

"ipaddress":<String_value>

}]}
```

get (all)

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/gslbsite_gslbservicegroupmember_binding

Query-parameters:

bulkbindings

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/gslbsite_gslbservicegroupmember_binding?**bulkbindings=yes**

NITRO allows you to fetch bindings in bulk.

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "gslbsite_gslbservicegroupmember_binding": [ {

"sitename":<String_value>,

"servicegroupname":<String_value>,

"state":<String_value>,

"servicetype":<String_value>,

"port":<Integer_value>,

"ipaddress":<String_value>
```

```
}]}
```

count

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/gslbsite_gslbservicegroupmember_binding/sitename_value<String>?**count=yes**

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{"gslbsite_gslbservicegroupmember_binding": [ { "__count": "#no"} ] }
```

gslbvserver_gslbservicegroup_binding

Binding object showing the gslbservicegroup that can be bound to gslbvserver.

Properties

(click to see Operations)

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
name	<String>	Read-write	Name of the virtual server on which to perform the binding operation.

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
			Minimum length = 1
servicegroupname	<String>	Read-write	The GSLB service group name bound to the selected GSLB virtual server.
__count	<Double>	Read-write	count parameter

Operations

(click to see Properties)

ADD:| DELETE:| GET| GET (ALL)| COUNT

Some options that you can use for each operations:

- **Getting warnings in response:**NITRO allows you to get warnings in an operation by specifying the "warning" query parameter as "yes". For example, to get warnings while connecting to the Citrix ADC appliance, the URL is as follows:

  http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/login?warning=yes

  If any, the warnings are displayed in the response payload with the HTTP code "209 X-NITRO-WARNING".

- **Authenticated access for individual NITRO operations:**NITRO allows you to logon to the Citrix ADC appliance to perform individual operations. You can use this option instead of creating a NITRO session (using the login object) and then using that session to perform all operations,

  To do this, you must specify the username and password in the request header of the NITRO request as follows:

  X-NITRO-USER:<username>

  X-NITRO-PASS:<password>

  Note: In such cases, make sure that the request header DOES not include the following:

  Cookie:NITRO_AUTH_TOKEN=<tokenvalue>

`*Note:`*

Mandatory parameters are marked in redand placeholder content is marked in <green>.

add

URL:http://<Citrix-ADC-IP-address(NSIP)/nitro/v1/config/gslbvserver_gslbservicegroup_binding

HTTP Method:PUT

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Content-Type:application/json

Request Payload:

```
{

"gslbvserver_gslbservicegroup_binding":{

<b>"name":<String_value>,

</b>"servicegroupname":<String_value>

}}
```

Response:

HTTP Status Code on Success: 201 Created HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error

delete

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/gslbvserver_gslbservicegroup_binding/name_value<String>

Query-parameters:

args

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/gslbvserver_gslbservicegroup_binding/name_value<String>?**args=servicegroupname:<String_value>**

HTTP Method:DELETE

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue>

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error

get

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/gslbvserver_gslbservicegroup_binding/name_value<String>

Query-parameters:

filter

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/gslbvserver_gslbservicegroup_binding/name_value<String>?**filter=property-name1:property-value1,property-name2:property-value2**

Use this query-parameter to get the filtered set of gslbvserver_gslbservicegroup_binding resources configured on Citrix ADC. Filtering can be done on any of the properties of the resource.

pagination

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/gslbvserver_gslbservicegroup_binding/name_value<String>?**pagesize=#no;pageno=#no**

Use this query-parameter to get the gslbvserver_gslbservicegroup_binding resources in chunks.

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "gslbvserver_gslbservicegroup_binding": [ {

"name":<String_value>,

"servicegroupname":<String_value>

}]}
```

get (all)

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/gslbvserver_gslbservicegroup_binding

Query-parameters:

bulkbindings

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/gslbvserver_gslbservicegroup_binding?**bulkbindings=yes**

NITRO allows you to fetch bindings in bulk.

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "gslbvserver_gslbservicegroup_binding": [ {

"name":<String_value>,

"servicegroupname":<String_value>

}]}
```

count

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/gslbvserver_gslbservicegroup_binding/name_value<String>?**count=yes**

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{"gslbvserver_gslbservicegroup_binding": [ { "__count": "#no"} ] }
```

gslbvserver_gslbservicegroupmember_binding

Binding object showing the gslbservicegroupmember that can be bound to gslbvserver.

Properties

(click to see Operations)

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
weight	<Double>	Read-write	Weight to assign to the GSLB service. Minimum value = 1 Maximum value = 100
name	<String>	Read-write	Name of the virtual server on which to perform the binding operation. Minimum length = 1
servicegroupname	<String>	Read-write	The GSLB service group name bound to the selected GSLB virtual server.
svcsitepersistence	<String>	Read-only	Type of Site Persistence set on the bound service. Possible values = ConnectionProxy, HTTPRedirect, NONE
preferredlocation	<String>	Read-only	The target site to be returned in the DNS response when a policy is successfully evaluated against the incoming DNS request. Target site is specified in dotted notation with up to 6 qualifiers. Wildcard

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
			`*' is accepted as a valid qualifier token.
servicetype	<String>	Read-only	Protocol used by services bound to the virtual server. Possible values = HTTP, FTP, TCP, UDP, SSL, SSL_BRIDGE, SSL_TCP, NNTP, ANY, SIP_UDP, SIP_TCP, SIP_SSL, RADIUS, RDP, RTSP, MYSQL, MSSQL, ORACLE
gslbthreshold	<Integer>	Read-only	Indicates if gslb svc has reached threshold.
sitepersistcookie	<String>	Read-only	This field is introduced for displaying the cookie in cluster setup. Minimum length = 1
port	<Integer>	Read-only	Port number. Range 1 - 65535 * in CLI is represented as 65535 in NITRO API
curstate	<String>	Read-only	State of the gslb vserver. Possible values = UP, DOWN,

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
			UNKNOWN, BUSY, OUT OF SERVICE, GOING OUT OF SERVICE, DOWN WHEN GOING OUT OF SERVICE, NS_EMPTY_STR, Unknown, DISABLED
svreffgslbstate	<String>	Read-only	Effective state of the gslb svc. Possible values = UP, DOWN, UNKNOWN, BUSY, OUT OF SERVICE, GOING OUT OF SERVICE, DOWN WHEN GOING OUT OF SERVICE, NS_EMPTY_STR, Unknown, DISABLED
dynamicweight	<String>	Read-only	Specify if the appliance should consider the service count, service weights, or ignore both when using weight-based load balancing methods. The state of the number of services bound to the virtual server help the appliance to select the service.

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
			Default value: DISABLED Possible values = SERVICECOUNT, SERVICEWEIGHT, DISABLED
thresholdvalue	<Integer>	Read-only	Tells whether threshold exceeded for this service participating in CUSTOMLB.
ipaddress	<String>	Read-only	IP address.
__count	<Double>	Read-only	count parameter

Operations

(click to see Properties)

GET| GET (ALL)| COUNT

Some options that you can use for each operations:

- **Getting warnings in response:**NITRO allows you to get warnings in an operation by specifying the "warning" query parameter as "yes". For example, to get warnings while connecting to the Citrix ADC appliance, the URL is as follows:

  http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/login?warning=yes

  If any, the warnings are displayed in the response payload with the HTTP code "209 X-NITRO-WARNING".

- **Authenticated access for individual NITRO operations:**NITRO allows you to logon to the Citrix ADC appliance to perform individual operations. You can use this option instead of creating a NITRO session (using the login object) and then using that session to perform all operations,

  To do this, you must specify the username and password in the request header of the NITRO request as follows:

X-NITRO-USER:<username>

X-NITRO-PASS:<password>

Note: In such cases, make sure that the request header DOES not include the following:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue>

`*Note:`*

Mandatory parameters are marked in redand placeholder content is marked in <green>.

get

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/gslbvserver_gslbservicegroupmember_binding/name_value<String>

Query-parameters:

filter

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/gslbvserver_gslbservicegroupmember_binding/name_value<String>?**filter=property-name1:property-value1,property-name2:property-value2**

Use this query-parameter to get the filtered set of gslbvserver_gslbservicegroupmember_binding resources configured on Citrix ADC. Filtering can be done on any of the properties of the resource.

pagination

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/gslbvserver_gslbservicegroupmember_binding/name_value<String>?**pagesize=#no;pageno=#no**

Use this query-parameter to get the gslbvserver_gslbservicegroupmember_binding resources in chunks.

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "gslbvserver_gslbservicegroupmember_binding": [ {

"weight":<Double_value>,

"name":<String_value>,

"servicegroupname":<String_value>,

"svcsitepersistence":<String_value>,

"preferredlocation":<String_value>,

"servicetype":<String_value>,

"gslbthreshold":<Integer_value>,

"sitepersistcookie":<String_value>,

"port":<Integer_value>,

"curstate":<String_value>,

"svreffgslbstate":<String_value>,

"dynamicweight":<String_value>,

"thresholdvalue":<Integer_value>,

"ipaddress":<String_value>

}]}
```

get (all)

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/gslbvserver_gslbservicegroupmember_binding

Query-parameters:

bulkbindings

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/gslbvserver_gslbservicegroupmember_binding?**bulkbindings=yes**

NITRO allows you to fetch bindings in bulk.

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "gslbvserver_gslbservicegroupmember_binding": [ {

"weight":<Double_value>,

"name":<String_value>,

"servicegroupname":<String_value>,

"svcsitepersistence":<String_value>,

"preferredlocation":<String_value>,

"servicetype":<String_value>,
```

```
"gslbthreshold":<Integer_value>,

"sitepersistcookie":<String_value>,

"port":<Integer_value>,

"curstate":<String_value>,

"svreffgslbstate":<String_value>,

"dynamicweight":<String_value>,

"thresholdvalue":<Integer_value>,

"ipaddress":<String_value>

}]}
```

count

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/gslbvserver_gslbservicegroupmember_binding/name_value<String>?**count=yes**

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{"gslbvserver_gslbservicegroupmember_binding": [ { "__count": "#no"} ] }
```

gslbdomain_gslbservicegroup_binding

Binding object showing the gslbservicegroup that can be bound to gslbdomain.

Properties

(click to see Operations)

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
servicegroupname	<String>	Read-write	The GSLB service group name bound to the selected GSLB virtual server.
name	<String>	Read-write	Name of the Domain. Minimum length = 1
__count	<Double>	Read-write	count parameter

Operations

(click to see Properties)

GET| GET (ALL)| COUNT

Some options that you can use for each operations:

- **Getting warnings in response:**NITRO allows you to get warnings in an operation by specifying the "warning" query parameter as "yes". For example, to get warnings while connecting to the Citrix ADC appliance, the URL is as follows:

  http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/login?warning=yes

  If any, the warnings are displayed in the response payload with the HTTP code "209 X-NITRO-WARNING".
- **Authenticated access for individual NITRO operations:**NITRO allows you to logon to the Citrix ADC appliance to perform individual operations. You can use this option instead of creating a NITRO session (using the login object) and then using that session to perform all operations,

To do this, you must specify the username and password in the request header of the NITRO request as follows:

X-NITRO-USER:<username>

X-NITRO-PASS:<password>

Note: In such cases, make sure that the request header DOES not include the following:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue>

`*Note:`*

Mandatory parameters are marked in redand placeholder content is marked in <green>.

get

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/gslbdomain_gslbservicegroup_binding/name_value<String>

Query-parameters:

filter

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/gslbdomain_gslbservicegroup_binding/name_value<String>?**filter=property-name1:property-value1,property-name2:property-value2**

Use this query-parameter to get the filtered set of gslbdomain_gslbservicegroup_binding resources configured on Citrix ADC. Filtering can be done on any of the properties of the resource.

pagination

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/gslbdomain_gslbservicegroup_binding/name_value<String>?**pagesize=#no;pageno=#no**

Use this query-parameter to get the gslbdomain_gslbservicegroup_binding resources in chunks.

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "gslbdomain_gslbservicegroup_binding": [ {

"servicegroupname":<String_value>,

"name":<String_value>

}]}
```

get (all)

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/gslbdomain_gslbservicegroup_binding

Query-parameters:

bulkbindings

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/gslbdomain_gslbservicegroup_binding?**bulkbindings=yes**

NITRO allows you to fetch bindings in bulk.

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "gslbdomain_gslbservicegroup_binding": [ {

"servicegroupname":<String_value>,

"name":<String_value>

}]}
```

count

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/gslbdomain_gslbservicegroup_binding/name_value<String>?**count=yes**

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{"gslbdomain_gslbservicegroup_binding": [ { "__count": "#no"} ] }
```

gslbdomain_gslbservicegroupmember_binding

Binding object showing the gslbservicegroupmember that can be bound to gslbdomain.

Properties

(click to see Operations)

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
servicegroupname	<String>	Read-write	The GSLB service group name bound to the selected GSLB virtual server.
name	<String>	Read-write	Name of the Domain. Minimum length = 1
gslbthreshold	<Integer>	Read-only	The threshold value of the service.
port	<Integer>	Read-only	Port Number. Minimum value = 1 Range 1 - 65535 * in CLI is represented as 65535 in NITRO API
svreffgslbstate	<String>	Read-only	GSLB server state. Possible values = UP, DOWN, UNKNOWN, BUSY, OUT OF SERVICE, GOING OUT OF SERVICE, DOWN WHEN GOING OUT OF SERVICE, NS_EMPTY_STR,

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
			Unknown, DISABLED
weight	<Double>	Read-only	weight assigned. Minimum value = 1 Maximum value = 100
servicetype	<String>	Read-only	The type GSLB service. Possible values = HTTP, FTP, TCP, UDP, SSL, SSL_BRIDGE, SSL_TCP, NNTP, ANY, SIP_UDP, SIP_TCP, SIP_SSL, RADIUS, RDP, RTSP, MYSQL, MSSQL, ORACLE
ipaddress	<String>	Read-only	The Ip address of the service.
__count	<Double>	Read-only	count parameter

Operations

(click to see Properties)

GET| GET (ALL)| COUNT

Some options that you can use for each operations:

- **Getting warnings in response:**NITRO allows you to get warnings in an operation by specifying the "warning" query parameter as "yes". For example, to get warnings while connecting to the Citrix ADC appliance, the URL is as follows:

  http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/login?warning=yes

If any, the warnings are displayed in the response payload with the HTTP code "209 X-NITRO-WARNING".

- **Authenticated access for individual NITRO operations:**NITRO allows you to logon to the Citrix ADC appliance to perform individual operations. You can use this option instead of creating a NITRO session (using the login object) and then using that session to perform all operations,

  To do this, you must specify the username and password in the request header of the NITRO request as follows:

  X-NITRO-USER:<username>

  X-NITRO-PASS:<password>

  Note: In such cases, make sure that the request header DOES not include the following:

  Cookie:NITRO_AUTH_TOKEN=<tokenvalue>

`*Note:`*

Mandatory parameters are marked in redand placeholder content is marked in <green>.

get

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/gslbdomain_gslbservicegroupmember_binding/name_value<String>

Query-parameters:

filter

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/gslbdomain_gslbservicegroupmember_binding/name_value<String>?**filter=property-name1:property-value1,property-name2:property-value2**

Use this query-parameter to get the filtered set of gslbdomain_gslbservicegroupmember_binding resources configured on Citrix ADC. Filtering can be done on any of the properties of the resource.

pagination

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/gslbdomain_gslbservicegroupmember_binding/name_value<String>?**pagesize=#no;pageno=#no**

Use this query-parameter to get the gslbdomain_gslbservicegroupmember_binding resources in chunks.

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "gslbdomain_gslbservicegroupmember_binding": [ {

"servicegroupname":<String_value>,

"name":<String_value>,

"gslbthreshold":<Integer_value>,

"port":<Integer_value>,

"svreffgslbstate":<String_value>,

"weight":<Double_value>,

"servicetype":<String_value>,

"ipaddress":<String_value>

}]}
```

get (all)

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/gslbdomain_gslbservicegroupmember_binding

Query-parameters:

bulkbindings

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/gslbdomain_gslbservicegroupmember_binding?**bulkbindings=yes**

NITRO allows you to fetch bindings in bulk.

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "gslbdomain_gslbservicegroupmember_binding": [ {

"servicegroupname":<String_value>,

"name":<String_value>,

"gslbthreshold":<Integer_value>,

"port":<Integer_value>,

"svreffgslbstate":<String_value>,

"weight":<Double_value>,
```

```
"servicetype":<String_value>,

"ipaddress":<String_value>

}]}
```

count

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/gslbdomain_gslbservicegroupmember_binding/name_value<String>?**count=yes**

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{"gslbdomain_gslbservicegroupmember_binding": [ { "__count": "#no"} ] }
```

High-Availability

High Availability

High-availability commands.

NAME	DESCRIPTION
hafailover	Configuration for failover
hafiles	Configuration for files
hanode	Configuration for node
hanode_binding	Binding object showing the resources that can be bound to hanode
hanode_ci_binding	Binding object showing the ci that can be bound to hanode
hanode_fis_binding	Binding object showing the fis that can be bound to hanode
hanode_partialfailureinterfaces_binding	Binding object showing the partialfailureinterfaces that can be bound to hanode
hanode_routemonitor6_binding	Binding object showing the routemonitor6 that can be bound to hanode
hanode_routemonitor_binding	Binding object showing the routemonitor that can be bound to hanode
hasync	Configuration for sync

hafailover

Configuration for failover resource.

Properties

(click to see Operations)

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
force	<Boolean>	Read-write	Force a failover without prompting for confirmation.

Operations

(click to see Properties)

FORCE

Some options that you can use for each operations:

- **Getting warnings in response:**NITRO allows you to get warnings in an operation by specifying the "warning" query parameter as "yes". For example, to get warnings while connecting to the Citrix ADC appliance, the URL is as follows:

  http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/login?warning=yes

  If any, the warnings are displayed in the response payload with the HTTP code "209 X-NITRO-WARNING".

- **Authenticated access for individual NITRO operations:**NITRO allows you to logon to the Citrix ADC appliance to perform individual operations. You can use this option instead of creating a NITRO session (using the login object) and then using that session to perform all operations,

  To do this, you must specify the username and password in the request header of the NITRO request as follows:

  X-NITRO-USER:<username>

  X-NITRO-PASS:<password>

  Note: In such cases, make sure that the request header DOES not include the following:

  Cookie:NITRO_AUTH_TOKEN=<tokenvalue>

`*Note:`*

Mandatory parameters are marked in redand placeholder content is marked in <green>.

Force

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/hafailover?**action=Force**

HTTP Method:POST

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Content-Type:application/json

Request Payload:

```
{"hafailover":{

"force":<Boolean_value>

}}
```

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error

hafiles

Configuration for files resource.

Properties

(click to see Operations)

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
mode	<String[]>	Read-write	Specify one of the following modes of synchronization. * all - Synchronize files related to system configuration, Access Gateway bookmarks, SSL certificates, SSL CRL lists, HTML injection scripts, and Application

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
			Firewall XML objects. * bookmarks - Synchronize all Access Gateway bookmarks. * ssl - Synchronize all certificates, keys, and CRLs for the SSL feature. * htmlinjection. Synchronize all scripts configured for the HTML injection feature. * imports. Synchronize all XML objects (for example, WSDLs, schemas, error pages) configured for the application firewall. * misc - Synchronize all license files and the rc.conf file. * all_plus_misc - Synchronize files related to system configuration, Access Gateway

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
			bookmarks, SSL certificates, SSL CRL lists, HTML injection scripts, application firewall XML objects, licenses, and the rc.conf file. Possible values = all, bookmarks, ssl, htmlinjection, imports, misc, dns, krb, AAA, app_catalog, all_plus_misc, all_minus_misc

Operations

(click to see Properties)

SYNC

Some options that you can use for each operations:

- **Getting warnings in response:**NITRO allows you to get warnings in an operation by specifying the "warning" query parameter as "yes". For example, to get warnings while connecting to the Citrix ADC appliance, the URL is as follows:

  http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/login?warning=yes

  If any, the warnings are displayed in the response payload with the HTTP code "209 X-NITRO-WARNING".

- **Authenticated access for individual NITRO operations:**NITRO allows you to logon to the Citrix ADC appliance to perform individual operations. You can use this option instead of creating a NITRO session (using the login object) and then using that session to perform all operations,

To do this, you must specify the username and password in the request header of the NITRO request as follows:

X-NITRO-USER:<username>

X-NITRO-PASS:<password>

Note: In such cases, make sure that the request header DOES not include the following:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue>

`*Note:`*

Mandatory parameters are marked in redand placeholder content is marked in <green>.

sync

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/hafiles?**action=sync**

HTTP Method:POST

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Content-Type:application/json

Request Payload:

```
{"hafiles":{

"mode":<String[]_value>

}}
```

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error

hanode

Configuration for node resource.

Properties

(click to see Operations)

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
id	<Double>	Read-write	Number that uniquely identifies the node. For self node, it will always be 0. Peer node values can range from 1-64. Minimum value = 1 Maximum value = 64
ipaddress	<String>	Read-write	The NSIP or NSIP6 address of the node to be added for an HA configuration. This setting is neither propagated nor synchronized. Minimum length = 1
inc	<String>	Read-write	This option is required if the HA nodes reside on different networks. When this mode is enabled, the following independent network entities and configurations are neither propagated nor synced to the other node: MIPs, SNIPs, VLANs, routes (except LLB routes), route

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
			monitors, RNAT rules (except any RNAT rule with a VIP as the NAT IP), and dynamic routing configurations. They are maintained independently on each node. Default value: DISABLED Possible values = ENABLED, DISABLED
hastatus	<String>	Read-write	The HA status of the node. The HA status STAYSECONDARY is used to force the secondary device stay as secondary independent of the state of the Primary device. For example, in an existing HA setup, the Primary node has to be upgraded and this process would take few seconds. During the upgradation, it is possible that the Primary node may suffer from a downtime for a few seconds. However, the Secondary should not take over as the Primary node. Thus, the Secondary

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
			node should remain as Secondary even if there is a failure in the Primary node. STAYPRIMARY configuration keeps the node in primary state in case if it is healthy, even if the peer node was the primary node initially. If the node with STAYPRIMARY setting (and no peer node) is added to a primary node (which has this node as the peer) then this node takes over as the new primary and the older node becomes secondary. ENABLED state means normal HA operation without any constraints/ preferences. DISABLED state disables the normal HA operation of the node. Possible values = ENABLED, STAYSECONDARY,

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
			DISABLED, STAYPRIMARY
hasync	<String>	Read-write	Automatically maintain synchronization by duplicating the configuration of the primary node on the secondary node. This setting is not propagated. Automatic synchronization requires that this setting be enabled (the default) on the current secondary node. Synchronization uses TCP port 3010. Default value: ENABLED Possible values = ENABLED, DISABLED
haprop	<String>	Read-write	Automatically propagate all commands from the primary to the secondary node, except the following: * All HA configuration related commands. For example, add ha node, set ha

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
			node, and bind ha node. * All Interface related commands. For example, set interface and unset interface. * All channels related commands. For example, add channel, set channel, and bind channel. The propagated command is executed on the secondary node before it is executed on the primary. If command propagation fails, or if command execution fails on the secondary, the primary node executes the command and logs an error. Command propagation uses port 3010. Note: After enabling propagation, run force

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
			synchronization on either node. Default value: ENABLED Possible values = ENABLED, DISABLED
hellointerval	<Double>	Read-write	Interval, in milliseconds, between heartbeat messages sent to the peer node. The heartbeat messages are UDP packets sent to port 3003 of the peer node. Default value: 200 Minimum value = 200 Maximum value = 1000
deadinterval	<Double>	Read-write	Number of seconds after which a peer node is marked DOWN if heartbeat messages are not received from the peer node. Default value: 3 Minimum value = 3 Maximum value = 60

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
failsafe	<String>	Read-write	Keep one node primary if both nodes fail the health check, so that a partially available node can back up data and handle traffic. This mode is set independently on each node. Default value: OFF Possible values = ON, OFF
maxflips	<Double>	Read-write	Max number of flips allowed before becoming sticky primary. Default value: 0
maxfliptime	<Double>	Read-write	Interval after which flipping of node states can again start. Default value: 0
syncvlan	<Double>	Read-write	Vlan on which HA related communication is sent. This include sync, propagation , connection mirroring , LB persistency config sync, persistent session sync and session state sync. However HA

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
			heartbeats can go all interfaces. Minimum value = 1 Maximum value = 4094
name	<String>	Read-only	Node Name.
flags	<Double>	Read-only	The flags for this entry.
state	<String>	Read-only	HA Master State. Possible values = Primary, Secondary, UNKNOWN
enaifaces	<String>	Read-only	Enabled interfaces.
disifaces	<String>	Read-only	Disabled interfaces.
hamonifaces	<String>	Read-only	HAMON ON interfaces.
haheartbeatifaces	<String>	Read-only	HAHEARTBEAT OFF interfaces.
pfifaces	<String>	Read-only	Interfaces causing Partial Failure.
ifaces	<String>	Read-only	Interfaces on which non-multicast is not seen.
netmask	<String>	Read-only	The netmask.
ssl2	<String>	Read-only	SSL card status. Possible values = DOWN, UP, NOT PRESENT, UNKNOWN

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
masterstatetime	<Double>	Read-only	Time elapsed in current master state.
routemonitor	<String>	Read-only	The IP address (IPv4 or IPv6).
curflips	<Double>	Read-only	Keeps track of number of flips that have happened till now in current interval.
completedfliptime	<Double>	Read-only	To inform user whether flip time is elapsed or not.
routemonitorstate	<String>	Read-only	State for route monitor. Possible values = UP, DOWN
hasyncfailurereason	<String>	Read-only	Displays the reason for HA SYNC Failure. Possible values =
__count	<Double>	Read-only	count parameter

Operations

(click to see Properties)

ADD| DELETE| UPDATE| UNSET| GET (ALL)| GET| COUNT

Some options that you can use for each operations:

- **Getting warnings in response:**NITRO allows you to get warnings in an operation by specifying the "warning" query parameter as "yes". For example, to get warnings while connecting to the Citrix ADC appliance, the URL is as follows:

  http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/login?warning=yes

If any, the warnings are displayed in the response payload with the HTTP code "209 X-NITRO-WARNING".

- **Authenticated access for individual NITRO operations:**NITRO allows you to logon to the Citrix ADC appliance to perform individual operations. You can use this option instead of creating a NITRO session (using the login object) and then using that session to perform all operations,

  To do this, you must specify the username and password in the request header of the NITRO request as follows:

  X-NITRO-USER:<username>

  X-NITRO-PASS:<password>

  Note: In such cases, make sure that the request header DOES not include the following:

  Cookie:NITRO_AUTH_TOKEN=<tokenvalue>

`*Note:`*

Mandatory parameters are marked in redand placeholder content is marked in <green>.

add

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/hanode

HTTP Method:POST

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Content-Type:application/json

Request Payload:

```
{"hanode":{

<b>"id":<Double_value>,

</b><b>"ipaddress":<String_value>,

</b>"inc":<String_value>
```

```
}}
```

Response:

HTTP Status Code on Success: 201 Created HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error

delete

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/hanode/id_value<Double>

HTTP Method:DELETE

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue>

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error

update

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/hanode

HTTP Method:PUT

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Content-Type:application/json

Request Payload:

```
{"hanode":{

"id":<Double_value>,

"hastatus":<String_value>,
```

```
"hasync":<String_value>,

"haprop":<String_value>,

"hellointerval":<Double_value>,

"deadinterval":<Double_value>,

"failsafe":<String_value>,

"maxflips":<Double_value>,

"maxfliptime":<Double_value>,

"syncvlan":<Double_value>

}}
```

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error

unset

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/hanode?**action=unset**

HTTP Method:POST

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Content-Type:application/json

Request Payload:

```
{"hanode":{

"id":<Double_value>,
```

```
"hastatus":true,

"hasync":true,

"haprop":true,

"hellointerval":true,

"deadinterval":true,

"failsafe":true,

"maxflips":true,

"maxfliptime":true,

"syncvlan":true

}}
```

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error

get (all)

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/hanode

Query-parameters:

attrs

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/hanode?**attrs=property-name1,property-name2**

Use this query parameter to specify the resource details that you want to retrieve.

filter

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/hanode?**filter=property-name1:property-val1,property-name2:property-val2**

Use this query-parameter to get the filtered set of hanode resources configured on Citrix ADC. Filtering can be done on any of the properties of the resource.

view

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/hanode?**view=summary**

Note: By default, the retrieved results are displayed in detail view (?view=detail).

pagination

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/hanode?**pagesize=#no;pageno=#no**

Use this query-parameter to get the hanode resources in chunks.

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "hanode": [ {

"id":<Double_value>,

"name":<String_value>,

"ipaddress":<String_value>,

"flags":<Double_value>,
```

```
"hastatus":<String_value>,

"state":<String_value>,

"hasync":<String_value>,

"haprop":<String_value>,

"enaifaces":<String_value>,

"disifaces":<String_value>,

"hamonifaces":<String_value>,

"haheartbeatifaces":<String_value>,

"pfifaces":<String_value>,

"ifaces":<String_value>,

"netmask":<String_value>,

"inc":<String_value>,

"ssl2":<String_value>,

"hellointerval":<Double_value>,

"deadinterval":<Double_value>,

"masterstatetime":<Double_value>,

"failsafe":<String_value>,

"routemonitor":<String_value>,

"maxflips":<Double_value>,

"maxfliptime":<Double_value>,

"curflips":<Double_value>,

"completedfliptime":<Double_value>,
```

```
"syncvlan":<Double_value>,

"routemonitorstate":<String_value>,

"hasyncfailurereason":<String_value>

}]}
```

get

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/hanode/id_value<Double>

Query-parameters:

attrs

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/hanode/id_value<Double>?**attrs=property-name1,property-name2**

Use this query parameter to specify the resource details that you want to retrieve.

view

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/hanode/id_value<Double>?**view=summary**

Note: By default, the retrieved results are displayed in detail view (?view=detail).

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "hanode": [ {
"id":<Double_value>,
"name":<String_value>,
"ipaddress":<String_value>,
"flags":<Double_value>,
"hastatus":<String_value>,
"state":<String_value>,
"hasync":<String_value>,
"haprop":<String_value>,
"enaifaces":<String_value>,
"disifaces":<String_value>,
"hamonifaces":<String_value>,
"haheartbeatifaces":<String_value>,
"pfifaces":<String_value>,
"ifaces":<String_value>,
"netmask":<String_value>,
"inc":<String_value>,
"ssl2":<String_value>,
"hellointerval":<Double_value>,
"deadinterval":<Double_value>,
"masterstatetime":<Double_value>,
```

```
"failsafe":<String_value>,

"routemonitor":<String_value>,

"maxflips":<Double_value>,

"maxfliptime":<Double_value>,

"curflips":<Double_value>,

"completedfliptime":<Double_value>,

"syncvlan":<Double_value>,

"routemonitorstate":<String_value>,

"hasyncfailurereason":<String_value>

}]}
```

count

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/hanode?**count=yes**

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "hanode": [ { "__count": "#no"} ] }
```

hanode_binding

Binding object which returns the resources bound to hanode.

Properties

(click to see Operations)

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
id	<Double>	Read-write	ID of the node whose HA settings you want to display. (The ID of the local node is always 0.). Minimum value = 0 Maximum value = 64
hanode_partialfailure interfaces_binding	<hanode_partialfailur einterfaces_binding[] >	Read-only	partialfailureinterface s that can be bound to hanode.
hanode_routemonito r6_binding	<hanode_routemonit or6_binding[]>	Read-only	routemonitor6 that can be bound to hanode.
hanode_ci_binding	<hanode_ci_binding[ ]>	Read-only	ci that can be bound to hanode.
hanode_routemonito r_binding	<hanode_routemonit or_binding[]>	Read-only	routemonitor that can be bound to hanode.
hanode_fis_binding	<hanode_fis_binding []>	Read-only	fis that can be bound to hanode.

Operations

(click to see Properties)

GET| GET (ALL)

Some options that you can use for each operations:

- **Getting warnings in response:**NITRO allows you to get warnings in an operation by specifying the "warning" query parameter as "yes". For example, to get warnings while connecting to the Citrix ADC appliance, the URL is as follows:

  http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/login?warning=yes

  If any, the warnings are displayed in the response payload with the HTTP code "209 X-NITRO-WARNING".

- **Authenticated access for individual NITRO operations:**NITRO allows you to logon to the Citrix ADC appliance to perform individual operations. You can use this option instead of creating a NITRO session (using the login object) and then using that session to perform all operations,

  To do this, you must specify the username and password in the request header of the NITRO request as follows:

  X-NITRO-USER:<username>

  X-NITRO-PASS:<password>

  Note: In such cases, make sure that the request header DOES not include the following:

  Cookie:NITRO_AUTH_TOKEN=<tokenvalue>

`*Note:`*

Mandatory parameters are marked in redand placeholder content is marked in <green>.

get

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/hanode_binding/id_value<Double>

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "hanode_binding": [ {

"id":<Double_value>,

"hanode_routemonitor_binding":<hanode_routemonitor_binding[]_value>,

"hanode_partialfailureinterfaces_binding":<hanode_partialfailureinterfaces
_binding[]_value>,

"hanode_fis_binding":<hanode_fis_binding[]_value>,

"hanode_routemonitor6_binding":<hanode_routemonitor6_binding[]_value>,

"hanode_ci_binding":<hanode_ci_binding[]_value>

}]}
```

get (all)

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/hanode_binding

Query-parameters:

bulkbindings

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/hanode_binding?**bulkbindings=yes**

NITRO allows you to fetch bindings in bulk.

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "hanode_binding": [ {

"id":<Double_value>,

"hanode_routemonitor_binding":<hanode_routemonitor_binding[]_value>,

"hanode_partialfailureinterfaces_binding":<hanode_partialfailureinterfaces
_binding[]_value>,

"hanode_fis_binding":<hanode_fis_binding[]_value>,

"hanode_routemonitor6_binding":<hanode_routemonitor6_binding[]_value>,

"hanode_ci_binding":<hanode_ci_binding[]_value>

}]}
```

hanode_ci_binding

Binding object showing the ci that can be bound to hanode.

Properties

(click to see Operations)

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
enaifaces	<String>	Read-write	Enabled interfaces.
routemonitor	<String>	Read-write	A route that you want the Citrix ADC to monitor in its internal routing table. You can specify an IPv4 address or network, or an IPv6 address or network prefix. If you specify an IPv4 network address or IPv6 network prefix, the appliance monitors any route that matches the network or prefix. Minimum length = 1
id	<Double>	Read-write	Number that uniquely identifies the local node. The ID of the local node is always 0. Minimum value = 0 Maximum value = 64
__count	<Double>	Read-write	count parameter

Operations

(click to see Properties)

GET| GET (ALL)| COUNT

Some options that you can use for each operations:

- **Getting warnings in response:**NITRO allows you to get warnings in an operation by specifying the "warning" query parameter as "yes". For example, to get warnings while connecting to the Citrix ADC appliance, the URL is as follows:

  http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/login?warning=yes

  If any, the warnings are displayed in the response payload with the HTTP code "209 X-NITRO-WARNING".

- **Authenticated access for individual NITRO operations:**NITRO allows you to logon to the Citrix ADC appliance to perform individual operations. You can use this option instead of creating a NITRO session (using the login object) and then using that session to perform all operations,

  To do this, you must specify the username and password in the request header of the NITRO request as follows:

  X-NITRO-USER:<username>

  X-NITRO-PASS:<password>

  Note: In such cases, make sure that the request header DOES not include the following:

  Cookie:NITRO_AUTH_TOKEN=<tokenvalue>

`*Note:`*

Mandatory parameters are marked in redand placeholder content is marked in <green>.

get

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/hanode_ci_binding/id_value<Double>

Query-parameters:

filter

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/hanode_ci_binding/id_value<Double>?**filter=property-name1:property-value1,property-name2:property-value2**

Use this query-parameter to get the filtered set of hanode_ci_binding resources configured on Citrix ADC. Filtering can be done on any of the properties of the resource.

pagination

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/hanode_ci_binding/id_value<Double>?**pagesize=#no;pageno=#no**

Use this query-parameter to get the hanode_ci_binding resources in chunks.

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "hanode_ci_binding": [ {

"enaifaces":<String_value>,

"routemonitor":<String_value>,

"id":<Double_value>

}]}
```

get (all)

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/hanode_ci_binding

Query-parameters:

bulkbindings

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/hanode_ci_binding?**bulkbindings=yes**

NITRO allows you to fetch bindings in bulk.

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "hanode_ci_binding": [ {

"enaifaces":<String_value>,

"routemonitor":<String_value>,

"id":<Double_value>

}]}
```

count

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/hanode_ci_binding/id_value<Double>?**count=yes**

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{"hanode_ci_binding": [ { "__count": "#no"} ] }
```

hanode_fis_binding

Binding object showing the fis that can be bound to hanode.

Properties

(click to see Operations)

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
name	<String>	Read-write	Node Name.
routemonitor	<String>	Read-write	A route that you want the Citrix ADC to monitor in its internal routing table. You can specify an IPv4 address or network, or an IPv6 address or network prefix. If you specify an IPv4 network address or IPv6 network prefix, the appliance monitors any route that matches the network or prefix. Minimum length = 1

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
id	<Double>	Read-write	Number that uniquely identifies the local node. The ID of the local node is always 0. Minimum value = 0 Maximum value = 64
enaifaces	<String>	Read-only	Enabled interfaces.
__count	<Double>	Read-only	count parameter

Operations

(click to see Properties)

GET| GET (ALL)| COUNT

Some options that you can use for each operations:

- **Getting warnings in response:**NITRO allows you to get warnings in an operation by specifying the "warning" query parameter as "yes". For example, to get warnings while connecting to the Citrix ADC appliance, the URL is as follows:

  http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/login?warning=yes

  If any, the warnings are displayed in the response payload with the HTTP code "209 X-NITRO-WARNING".

- **Authenticated access for individual NITRO operations:**NITRO allows you to logon to the Citrix ADC appliance to perform individual operations. You can use this option instead of creating a NITRO session (using the login object) and then using that session to perform all operations,

  To do this, you must specify the username and password in the request header of the NITRO request as follows:

  X-NITRO-USER:<username>

  X-NITRO-PASS:<password>

Note: In such cases, make sure that the request header DOES not include the following:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue>

`*Note:`*

Mandatory parameters are marked in redand placeholder content is marked in <green>.

get

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/hanode_fis_binding/id_value<Double>

Query-parameters:

filter

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/hanode_fis_binding/id_value<Double>?**filter=property-name1:property-value1,property-name2:property-value2**

Use this query-parameter to get the filtered set of hanode_fis_binding resources configured on Citrix ADC. Filtering can be done on any of the properties of the resource.

pagination

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/hanode_fis_binding/id_value<Double>?**pagesize=#no;pageno=#no**

Use this query-parameter to get the hanode_fis_binding resources in chunks.

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "hanode_fis_binding": [ {

"name":<String_value>,

"routemonitor":<String_value>,

"id":<Double_value>,

"enaifaces":<String_value>

}]}
```

get (all)

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/hanode_fis_binding

Query-parameters:

bulkbindings

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/hanode_fis_binding?**bulkbindings=yes**

NITRO allows you to fetch bindings in bulk.

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "hanode_fis_binding": [ {

"name":<String_value>,

"routemonitor":<String_value>,

"id":<Double_value>,

"enaifaces":<String_value>

}]}
```

count

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/hanode_fis_binding/id_value<Double>?**count=yes**

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{"hanode_fis_binding": [ { "__count": "#no"} ] }
```

hanode_partialfailureinterfaces_binding

Binding object showing the partialfailureinterfaces that can be bound to hanode.

Properties

(click to see Operations)

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
pfifaces	<String>	Read-write	Interfaces causing Partial Failure.
routemonitor	<String>	Read-write	A route that you want the Citrix ADC to monitor in its internal routing table. You can specify an IPv4 address or network, or an IPv6 address or network prefix. If you specify an IPv4 network address or IPv6 network prefix, the appliance monitors any route that matches the network or prefix. Minimum length = 1
id	<Double>	Read-write	Number that uniquely identifies the local node. The ID of the local node is always 0. Minimum value = 0 Maximum value = 64
__count	<Double>	Read-write	count parameter

Operations

(click to see Properties)

GET| GET (ALL)| COUNT

Some options that you can use for each operations:

- **Getting warnings in response:**NITRO allows you to get warnings in an operation by specifying the "warning" query parameter as "yes". For example, to get warnings while connecting to the Citrix ADC appliance, the URL is as follows:

  http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/login?warning=yes

  If any, the warnings are displayed in the response payload with the HTTP code "209 X-NITRO-WARNING".

- **Authenticated access for individual NITRO operations:**NITRO allows you to logon to the Citrix ADC appliance to perform individual operations. You can use this option instead of creating a NITRO session (using the login object) and then using that session to perform all operations,

  To do this, you must specify the username and password in the request header of the NITRO request as follows:

  X-NITRO-USER:<username>

  X-NITRO-PASS:<password>

  Note: In such cases, make sure that the request header DOES not include the following:

  Cookie:NITRO_AUTH_TOKEN=<tokenvalue>

`*Note:`*

Mandatory parameters are marked in redand placeholder content is marked in <green>.

get

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/hanode_partialfailureinterfaces_binding/id_value<Double>

Query-parameters:

filter

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/hanode_partialfailureinterfaces_binding/id_value<Double>?**filter=property-name1:property-value1,property-name2:property-value2**

Use this query-parameter to get the filtered set of hanode_partialfailureinterfaces_binding resources configured on Citrix ADC. Filtering can be done on any of the properties of the resource.

pagination

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/hanode_partialfailureinterfaces_binding/id_value<Double>?**pagesize=#no;pageno=#no**

Use this query-parameter to get the hanode_partialfailureinterfaces_binding resources in chunks.

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "hanode_partialfailureinterfaces_binding": [ {

"pfifaces":<String_value>,

"routemonitor":<String_value>,

"id":<Double_value>

}]}
```

get (all)

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/hanode_partialfailureinterfaces_binding

Query-parameters:

bulkbindings

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/hanode_partialfailureinterfaces_binding?**bulkbindings=yes**

NITRO allows you to fetch bindings in bulk.

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "hanode_partialfailureinterfaces_binding": [ {

"pfifaces":<String_value>,

"routemonitor":<String_value>,

"id":<Double_value>

}]}
```

count

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/hanode_partialfailureinterfaces_binding/id_value<Double>?**count=yes**

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{"hanode_partialfailureinterfaces_binding": [ { "__count": "#no"} ] }
```

hanode_routemonitor6_binding

Binding object showing the routemonitor6 that can be bound to hanode.

Properties

(click to see Operations)

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
routemonitor	<String>	Read-write	The IP address (IPv4 or IPv6).
id	<Double>	Read-write	Number that uniquely identifies the local node. The ID of the local node is always 0.

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
			Minimum value = 0 Maximum value = 64
netmask	<String>	Read-write	The netmask.
routemonitorstate	<String>	Read-only	State for route monitor. Possible values = UP, DOWN
flags	<Double>	Read-only	The flags for this entry.
__count	<Double>	Read-only	count parameter

Operations

(click to see Properties)

ADD:| DELETE:| GET| GET (ALL)| COUNT

Some options that you can use for each operations:

- **Getting warnings in response:**NITRO allows you to get warnings in an operation by specifying the "warning" query parameter as "yes". For example, to get warnings while connecting to the Citrix ADC appliance, the URL is as follows:

  http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/login?warning=yes

  If any, the warnings are displayed in the response payload with the HTTP code "209 X-NITRO-WARNING".

- **Authenticated access for individual NITRO operations:**NITRO allows you to logon to the Citrix ADC appliance to perform individual operations. You can use this option instead of creating a NITRO session (using the login object) and then using that session to perform all operations,

  To do this, you must specify the username and password in the request header of the NITRO request as follows:

X-NITRO-USER:<username>

X-NITRO-PASS:<password>

Note: In such cases, make sure that the request header DOES not include the following:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue>

`*Note:`*

Mandatory parameters are marked in redand placeholder content is marked in <green>.

add

URL:http://<Citrix-ADC-IP-address(NSIP)/nitro/v1/config/hanode_routemonitor6_binding

HTTP Method:PUT

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Content-Type:application/json

Request Payload:

```
{

"hanode_routemonitor6_binding":{

"id":<Double_value>,

<b>"routemonitor":<String_value>,

</b>"netmask":<String_value>

}}
```

Response:

HTTP Status Code on Success: 201 Created HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error

delete

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/hanode_routemonitor6_binding/id_value<Double>

Query-parameters:

args

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/hanode_routemonitor6_binding/id_value<Double>?**args=routemonitor:<String_value>,netmask:<String_value>**

HTTP Method:DELETE

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue>

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error

get

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/hanode_routemonitor6_binding/id_value<Double>

Query-parameters:

filter

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/hanode_routemonitor6_binding/id_value<Double>?**filter=property-name1:property-value1,property-name2:property-value2**

Use this query-parameter to get the filtered set of hanode_routemonitor6_binding resources configured on Citrix ADC. Filtering can be done on any of the properties of the resource.

pagination

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/hanode_routemonitor6_binding/id_value<Double>?**pagesize=#no;pageno=#no**

Use this query-parameter to get the hanode_routemonitor6_binding resources in chunks.

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "hanode_routemonitor6_binding": [ {

"routemonitor":<String_value>,

"id":<Double_value>,

"netmask":<String_value>,

"routemonitorstate":<String_value>,

"flags":<Double_value>

}]}
```

get (all)

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/hanode_routemonitor6_binding

Query-parameters:

bulkbindings

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/hanode_routemonitor6_binding?**bulkbindings=yes**

NITRO allows you to fetch bindings in bulk.

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "hanode_routemonitor6_binding": [ {

"routemonitor":<String_value>,

"id":<Double_value>,

"netmask":<String_value>,

"routemonitorstate":<String_value>,

"flags":<Double_value>

}]}
```

count

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/hanode_routemonitor6_binding/id_value<Double>?**count=yes**

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{"hanode_routemonitor6_binding": [ { "__count": "#no"} ] }
```

hanode_routemonitor_binding

Binding object showing the routemonitor that can be bound to hanode.

Properties

(click to see Operations)

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
routemonitor	<String>	Read-write	The IP address (IPv4 or IPv6).
id	<Double>	Read-write	Number that uniquely identifies the local node. The ID of the local node is always 0. Minimum value = 0 Maximum value = 64
netmask	<String>	Read-write	The netmask.

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
routemonitorstate	<String>	Read-only	State for route monitor. Possible values = UP, DOWN
flags	<Double>	Read-only	The flags for this entry.
__count	<Double>	Read-only	count parameter

Operations

(click to see Properties)

ADD:| DELETE:| GET| GET (ALL)| COUNT

Some options that you can use for each operations:

- **Getting warnings in response:**NITRO allows you to get warnings in an operation by specifying the "warning" query parameter as "yes". For example, to get warnings while connecting to the Citrix ADC appliance, the URL is as follows:

  http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/login?warning=yes

  If any, the warnings are displayed in the response payload with the HTTP code "209 X-NITRO-WARNING".

- **Authenticated access for individual NITRO operations:**NITRO allows you to logon to the Citrix ADC appliance to perform individual operations. You can use this option instead of creating a NITRO session (using the login object) and then using that session to perform all operations,

  To do this, you must specify the username and password in the request header of the NITRO request as follows:

  X-NITRO-USER:<username>

  X-NITRO-PASS:<password>

  Note: In such cases, make sure that the request header DOES not include the following:

  Cookie:NITRO_AUTH_TOKEN=<tokenvalue>

`*Note:` *

Mandatory parameters are marked in redand placeholder content is marked in <green>.

add

URL:http://<Citrix-ADC-IP-address(NSIP)/nitro/v1/config/hanode_routemonitor_binding

HTTP Method:PUT

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Content-Type:application/json

Request Payload:

```
{

"hanode_routemonitor_binding":{

"id":<Double_value>,

<b>"routemonitor":<String_value>,

</b>"netmask":<String_value>

}}
```

Response:

HTTP Status Code on Success: 201 Created HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error

delete

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/hanode_routemonitor_binding/id_value<Double>

Query-parameters:

args

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/hanode_routemonitor_binding/id_value<Double>?**args=routemonitor:<String_value>,netmask:<String_value>**

HTTP Method:DELETE

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue>

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error

get

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/hanode_routemonitor_binding/id_value<Double>

Query-parameters:

filter

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/hanode_routemonitor_binding/id_value<Double>?**filter=property-name1:property-value1,property-name2:property-value2**

Use this query-parameter to get the filtered set of hanode_routemonitor_binding resources configured on Citrix ADC. Filtering can be done on any of the properties of the resource.

pagination

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/hanode_routemonitor_binding/id_value<Double>?**pagesize=#no;pageno=#no**

Use this query-parameter to get the hanode_routemonitor_binding resources in chunks.

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "hanode_routemonitor_binding": [ {

"routemonitor":<String_value>,

"id":<Double_value>,

"netmask":<String_value>,

"routemonitorstate":<String_value>,

"flags":<Double_value>

}]}
```

get (all)

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/hanode_routemonitor_binding

Query-parameters:

bulkbindings

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/hanode_routemonitor_binding?**bulkbindings=yes**

NITRO allows you to fetch bindings in bulk.

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "hanode_routemonitor_binding": [ {

"routemonitor":<String_value>,

"id":<Double_value>,

"netmask":<String_value>,

"routemonitorstate":<String_value>,

"flags":<Double_value>

}]}
```

count

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/hanode_routemonitor_binding/id_value<Double>?**count=yes**

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{"hanode_routemonitor_binding": [ { "__count": "#no"} ] }
```

hasync

Configuration for sync resource.

Properties

(click to see Operations)

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
force	<Boolean>	Read-write	Force synchronization regardless of the state of HA propagation and HA synchronization on either node.
save	<String>	Read-write	After synchronization, automatically save the configuration in the secondary node configuration file (ns.conf) without prompting for confirmation. Possible values = YES, NO

Operations

(click to see Properties)

FORCE

Some options that you can use for each operations:

- **Getting warnings in response:**NITRO allows you to get warnings in an operation by specifying the "warning" query parameter as "yes". For example, to get warnings while connecting to the Citrix ADC appliance, the URL is as follows:

  http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/login?warning=yes

  If any, the warnings are displayed in the response payload with the HTTP code "209 X-NITRO-WARNING".

- **Authenticated access for individual NITRO operations:**NITRO allows you to logon to the Citrix ADC appliance to perform individual operations. You can use this option instead of creating a NITRO session (using the login object) and then using that session to perform all operations,

  To do this, you must specify the username and password in the request header of the NITRO request as follows:

  X-NITRO-USER:<username>

  X-NITRO-PASS:<password>

  Note: In such cases, make sure that the request header DOES not include the following:

  Cookie:NITRO_AUTH_TOKEN=<tokenvalue>

`*Note:`*

Mandatory parameters are marked in redand placeholder content is marked in <green>.

Force

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/hasync?**action=Force**

HTTP Method:POST

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Content-Type:application/json

Request Payload:

```
{"hasync":{
"force":<Boolean_value>,
"save":<String_value>
}}
```

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error

HTTP DoS Protection

HTTP DoS Protection

HTTP DOS protection configuration.

NAME	DESCRIPTION
dospolicy	Configuration for DoS policy

dospolicy

Configuration for DoS policy resource.

Properties

(click to see Operations)

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
name	<String>	Read-write	Name for the HTTP DoS protection policy. Must begin with a letter, number, or the underscore

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
			character (_). Other characters allowed, after the first character, are the hyphen (-), period (.) hash (#), space ( ), at (@), equals (=), and colon (:) characters. Minimum length = 1
qdepth	<Double>	Read-write	Queue depth. The queue size (the number of outstanding service requests on the system) before DoS protection is activated on the service to which the DoS protection policy is bound. Minimum value = 21
cltdetectrate	<Double>	Read-write	Client detect rate. Integer representing the percentage of traffic to which the HTTP DoS policy is to be applied after the queue depth condition is satisfied. Minimum value = 0 Maximum value = 100
__count	<Double>	Read-only	count parameter

Operations

(click to see Properties)

ADD| DELETE| UPDATE| UNSET| GET (ALL)| GET| COUNT

Some options that you can use for each operations:

- **Getting warnings in response:**NITRO allows you to get warnings in an operation by specifying the "warning" query parameter as "yes". For example, to get warnings while connecting to the Citrix ADC appliance, the URL is as follows:

  http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/login?warning=yes

  If any, the warnings are displayed in the response payload with the HTTP code "209 X-NITRO-WARNING".

- **Authenticated access for individual NITRO operations:**NITRO allows you to logon to the Citrix ADC appliance to perform individual operations. You can use this option instead of creating a NITRO session (using the login object) and then using that session to perform all operations,

  To do this, you must specify the username and password in the request header of the NITRO request as follows:

  X-NITRO-USER:<username>

  X-NITRO-PASS:<password>

  Note: In such cases, make sure that the request header DOES not include the following:

  Cookie:NITRO_AUTH_TOKEN=<tokenvalue>

`*Note:`*

Mandatory parameters are marked in redand placeholder content is marked in <green>.

add

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/dospolicy

HTTP Method:POST

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Content-Type:application/json

Request Payload:

```
{"dospolicy":{

<b>"name":<String_value>,

</b><b>"qdepth":<Double_value>,

</b>"cltdetectrate":<Double_value>

}}
```

Response:

HTTP Status Code on Success: 201 Created HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error

delete

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/dospolicy/name_value<String>

HTTP Method:DELETE

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue>

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error

update

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/dospolicy

HTTP Method:PUT

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Content-Type:application/json

Request Payload:

```
{"dospolicy":{

<b>"name":<String_value>,

</b>"qdepth":<Double_value>,

"cltdetectrate":<Double_value>

}}
```

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error

unset

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/dospolicy?**action=unset**

HTTP Method:POST

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Content-Type:application/json

Request Payload:

```
{"dospolicy":{

<b>"name":<String_value>,

</b>"cltdetectrate":true

}}
```

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error

get (all)

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/dospolicy

Query-parameters:

attrs

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/dospolicy?**attrs=property-name1,property-name2**

Use this query parameter to specify the resource details that you want to retrieve.

filter

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/dospolicy?**filter=property-name1:property-val1,property-name2:property-val2**

Use this query-parameter to get the filtered set of dospolicy resources configured on Citrix ADC. Filtering can be done on any of the properties of the resource.

view

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/dospolicy?**view=summary**

Note: By default, the retrieved results are displayed in detail view (?view=detail).

pagination

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/dospolicy?**pagesize=#no;pageno=#no**

Use this query-parameter to get the dospolicy resources in chunks.

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "dospolicy": [ {

"name":<String_value>,

"qdepth":<Double_value>,

"cltdetectrate":<Double_value>

}]}
```

get

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/dospolicy/name_value<String>

Query-parameters:

attrs

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/dospolicy/name_value<String>?**attrs=property-name1,property-name2**

Use this query parameter to specify the resource details that you want to retrieve.

view

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/dospolicy/name_value<String>?**view=summary**

Note: By default, the retrieved results are displayed in detail view (?view=detail).

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "dospolicy": [ {

"name":<String_value>,

"qdepth":<Double_value>,

"cltdetectrate":<Double_value>

}]}
```

count

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/dospolicy?**count=yes**

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "dospolicy": [ { "__count": "#no"} ] }
```

ICA

Ica

ICA configuration.

NAME	DESCRIPTION
icaaccessprofile	Configuration for ica accessprofile
icaaction	Configuration for ica action
icaglobal_binding	Binding object showing the resources that can be bound to icaglobal
icaglobal_icapolicy_binding	Binding object showing the icapolicy that can be bound to icaglobal
icalatencyprofile	Configuration for Profile for Latency monitoring
icaparameter	Configuration for Config Parameters for NS ICA
icapolicy	Configuration for ICA policy
icapolicy_binding	Binding object showing the resources that can be bound to icapolicy
icapolicy_crvserver_binding	Binding object showing the crvserver that can be bound to icapolicy
icapolicy_icaglobal_binding	Binding object showing the icaglobal that can be bound to icapolicy

NAME	DESCRIPTION
icapolicy_vpnvserver_binding	Binding object showing the vpnvserver that can be bound to icapolicy

icaaccessprofile

Configuration for ica accessprofile resource.

Properties

(click to see Operations)

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
name	<String>	Read-write	Name for the ICA accessprofile. Must begin with a letter, number, or the underscore character (_), and must contain only letters, numbers, and the hyphen (-), period (.) pound (#), space ( ), at (@), equals (=), colon (:), and underscore characters. Cannot be changed after the ICA accessprofile is added. The following requirement applies only to the Citrix ADC CLI:

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
			If the name includes one or more spaces, enclose the name in double or single quotation marks (for example, "my ica accessprofile" or 'my ica accessprofile'). Each of the features can be configured as DEFAULT/ DISABLED. Here, DISABLED means that the policy settings on the backend XenApp/ XenDesktop server are overridden and the Citrix ADC makes the decision to deny access. Whereas DEFAULT means that the Citrix ADC allows the request to reach the XenApp/ XenDesktop that takes the decision to allow/deny access based on the policy configured on it. For

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
			example, if ClientAudioRedirection is enabled on the backend XenApp/XenDesktop server, and the configured profile has ClientAudioRedirection as DISABLED, the Citrix ADC makes the decision to deny the request irrespective of the configuration on the backend. If the configured profile has ClientAudioRedirection as DEFAULT, then the Citrix ADC forwards the requests to the backend XenApp/XenDesktop server.It then makes the decision to allow/deny access based on the policy configured on it. Minimum length = 1
connectclientlptports	<String>	Read-write	Allow Default access/Disable automatic connection of LPT ports from the

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
			client when the user logs on. Default value: DISABLED Possible values = DEFAULT, DISABLED
clientaudioredirection	<String>	Read-write	Allow Default access/Disable applications hosted on the server to play sounds through a sound device installed on the client computer, also allows or prevents users to record audio input. Default value: DISABLED Possible values = DEFAULT, DISABLED
localremotedatasharing	<String>	Read-write	Allow Default access/Disable file/data sharing via the Receiver for HTML5. Default value: DISABLED Possible values = DEFAULT, DISABLED
clientclipboardredirection	<String>	Read-write	Allow Default access/Disable the

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
			clipboard on the client device to be mapped to the clipboard on the server. Default value: DISABLED Possible values = DEFAULT, DISABLED
clientcomportredirection	<String>	Read-write	Allow Default access/Disable COM port redirection to and from the client. Default value: DISABLED Possible values = DEFAULT, DISABLED
clientdriveredirection	<String>	Read-write	Allow Default access/Disables drive redirection to and from the client. Default value: DISABLED Possible values = DEFAULT, DISABLED
clientprinterredirection	<String>	Read-write	Allow Default access/Disable client printers to be mapped to a server when a user logs on to a session.

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
			Default value: DISABLED Possible values = DEFAULT, DISABLED
multistream	<String>	Read-write	Allow Default access/Disable the multistream feature for the specified users. Default value: DISABLED Possible values = DEFAULT, DISABLED
clientusbdriveredirection	<String>	Read-write	Allow Default access/Disable the redirection of USB devices to and from the client. Default value: DISABLED Possible values = DEFAULT, DISABLED
refcnt	<Double>	Read-only	Number of entities using this accessprofile.
isdefault	<Boolean>	Read-only	A value of true is returned if it is a default accessprofile.
__count	<Double>	Read-only	count parameter

Operations

(click to see Properties)

ADD| DELETE| UPDATE| UNSET| GET (ALL)| GET| COUNT

Some options that you can use for each operations:

- **Getting warnings in response:**NITRO allows you to get warnings in an operation by specifying the "warning" query parameter as "yes". For example, to get warnings while connecting to the Citrix ADC appliance, the URL is as follows:

  http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/login?warning=yes

  If any, the warnings are displayed in the response payload with the HTTP code "209 X-NITRO-WARNING".

- **Authenticated access for individual NITRO operations:**NITRO allows you to logon to the Citrix ADC appliance to perform individual operations. You can use this option instead of creating a NITRO session (using the login object) and then using that session to perform all operations,

  To do this, you must specify the username and password in the request header of the NITRO request as follows:

  X-NITRO-USER:<username>

  X-NITRO-PASS:<password>

  Note: In such cases, make sure that the request header DOES not include the following:

  Cookie:NITRO_AUTH_TOKEN=<tokenvalue>

`*Note:`*

Mandatory parameters are marked in redand placeholder content is marked in <green>.

add

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/icaaccessprofile

HTTP Method:POST

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Content-Type:application/json

Request Payload:

```
{"icaaccessprofile":{

<b>"name":<String_value>,

</b>"connectclientlptports":<String_value>,

"clientaudioredirection":<String_value>,

"localremotedatasharing":<String_value>,

"clientclipboardredirection":<String_value>,

"clientcomportredirection":<String_value>,

"clientdriveredirection":<String_value>,

"clientprinterredirection":<String_value>,

"multistream":<String_value>,

"clientusbdriveredirection":<String_value>

}}
```

Response:

HTTP Status Code on Success: 201 Created HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error

delete

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/icaaccessprofile/name_value<String>

HTTP Method:DELETE

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue>

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error

update

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/icaaccessprofile

HTTP Method:PUT

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Content-Type:application/json

Request Payload:

```
{"icaaccessprofile":{

<b>"name":<String_value>,

</b>"connectclientlptports":<String_value>,

"clientaudioredirection":<String_value>,

"localremotedatasharing":<String_value>,

"clientclipboardredirection":<String_value>,

"clientcomportredirection":<String_value>,

"clientdriveredirection":<String_value>,

"clientprinterredirection":<String_value>,

"multistream":<String_value>,

"clientusbdriveredirection":<String_value>

}}
```

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error

unset

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/icaaccessprofile?**action=unset**

HTTP Method:POST

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Content-Type:application/json

Request Payload:

```
{"icaaccessprofile":{

<b>"name":<String_value>,

</b>"connectclientlptports":true,

"clientaudioredirection":true,

"localremotedatasharing":true,

"clientclipboardredirection":true,

"clientcomportredirection":true,

"clientdriveredirection":true,

"clientprinterredirection":true,

"multistream":true,

"clientusbdriveredirection":true
```

```
}}
```

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error

get (all)

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/icaaccessprofile

Query-parameters:

attrs

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/icaaccessprofile?**attrs=property-name1,property-name2**

Use this query parameter to specify the resource details that you want to retrieve.

filter

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/icaaccessprofile?**filter=property-name1:property-val1,property-name2:property-val2**

Use this query-parameter to get the filtered set of icaaccessprofile resources configured on Citrix ADC. Filtering can be done on any of the properties of the resource.

view

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/icaaccessprofile?**view=summary**

Note: By default, the retrieved results are displayed in detail view (?view=detail).

pagination

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/icaaccessprofile?**pagesize=#no;pageno=#no**

Use this query-parameter to get the icaaccessprofile resources in chunks.

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "icaaccessprofile": [ {

"name":<String_value>,

"connectclientlptports":<String_value>,

"clientaudioredirection":<String_value>,

"localremotedatasharing":<String_value>,

"clientclipboardredirection":<String_value>,

"clientcomportredirection":<String_value>,

"clientdriveredirection":<String_value>,

"clientprinterredirection":<String_value>,

"multistream":<String_value>,

"clientusbdriveredirection":<String_value>,

"refcnt":<Double_value>,

"isdefault":<Boolean_value>

}]}
```

get

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/icaaccessprofile/name_value<String>

Query-parameters:

attrs

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/icaaccessprofile/name_value<String>?**attrs=property-name1,property-name2**

Use this query parameter to specify the resource details that you want to retrieve.

view

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/icaaccessprofile/name_value<String>?**view=summary**

Note: By default, the retrieved results are displayed in detail view (?view=detail).

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "icaaccessprofile": [ {

"name":<String_value>,

"connectclientlptports":<String_value>,
```

```
"clientaudioredirection":<String_value>,

"localremotedatasharing":<String_value>,

"clientclipboardredirection":<String_value>,

"clientcomportredirection":<String_value>,

"clientdriveredirection":<String_value>,

"clientprinterredirection":<String_value>,

"multistream":<String_value>,

"clientusbdriveredirection":<String_value>,

"refcnt":<Double_value>,

"isdefault":<Boolean_value>

}]}
```

count

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/icaaccessprofile?**count=yes**

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "icaaccessprofile": [ { "__count": "#no"} ] }
```

icaaction

Configuration for ica action resource.

Properties

(click to see Operations)

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
name	<String>	Read-write	Name for the ICA action. Must begin with a letter, number, or the underscore character (_), and must contain only letters, numbers, and the hyphen (-), period (.) pound (#), space ( ), at (@), equals (=), colon (:), and underscore characters. Cannot be changed after the ICA action is added. The following requirement applies only to the Citrix ADC CLI: If the name includes one or more spaces, enclose the name in double or single quotation marks (for example,

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
			"my ica action" or 'my ica action'). Minimum length = 1
accessprofilename	<String>	Read-write	Name of the ica accessprofile to be associated with this action.
latencyprofilename	<String>	Read-write	Name of the ica latencyprofile to be associated with this action.
newname	<String>	Read-write	New name for the ICA action. Must begin with an ASCII alphabetic or underscore (_) character, and must contain only ASCII alphanumeric, underscore, hash (#),period (.), space, colon (:), at (@), equals (=), and hyphen (-) characters. The following requirement applies only to the Citrix ADC CLI: If the name includes one or more spaces, enclose the name in double or single quotation marks (for

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
			example, "my ica action" or 'my ica action'). Minimum length = 1
hits	<Double>	Read-only	The number of times the action has been taken.
referencecount	<Double>	Read-only	The number of references to the action.
undefhits	<Double>	Read-only	The number of times the action resulted in UNDEF.
isdefault	<Boolean>	Read-only	A value of true is returned if it is a default action.
__count	<Double>	Read-only	count parameter

Operations

(click to see Properties)

ADD| DELETE| UPDATE| UNSET| RENAME| GET (ALL)| GET| COUNT

Some options that you can use for each operations:

- **Getting warnings in response:**NITRO allows you to get warnings in an operation by specifying the "warning" query parameter as "yes". For example, to get warnings while connecting to the Citrix ADC appliance, the URL is as follows:

  http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/login?warning=yes

  If any, the warnings are displayed in the response payload with the HTTP code "209 X-NITRO-WARNING".

- **Authenticated access for individual NITRO operations:**NITRO allows you to logon to the Citrix ADC appliance to perform individual operations. You can use this option instead of creating a NITRO session (using the login object) and then using that session to perform all operations,

  To do this, you must specify the username and password in the request header of the NITRO request as follows:

  X-NITRO-USER:<username>

  X-NITRO-PASS:<password>

  Note: In such cases, make sure that the request header DOES not include the following:

  Cookie:NITRO_AUTH_TOKEN=<tokenvalue>

`*Note:`*

Mandatory parameters are marked in redand placeholder content is marked in <green>.

add

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/icaaction

HTTP Method:POST

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Content-Type:application/json

Request Payload:

```
{"icaaction":{

<b>"name":<String_value>,

</b>"accessprofilename":<String_value>,

"latencyprofilename":<String_value>

}}
```

Response:

HTTP Status Code on Success: 201 Created HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error

delete

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/icaaction/name_value<String>

HTTP Method:DELETE

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue>

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error

update

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/icaaction

HTTP Method:PUT

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Content-Type:application/json

Request Payload:

```
{"icaaction":{

<b>"name":<String_value>,

</b>"accessprofilename":<String_value>,

"latencyprofilename":<String_value>

}}
```

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error

unset

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/icaaction?**action=unset**

HTTP Method:POST

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Content-Type:application/json

Request Payload:

```
{"icaaction":{

<b>"name":<String_value>,

</b>"accessprofilename":true,

"latencyprofilename":true

}}
```

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error

rename

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/icaaction?**action=rename**

HTTP Method:POST

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Content-Type:application/json

Request Payload:

```
{"icaaction":{

<b>"name":<String_value>,

</b><b>"newname":<String_value>

</b>}}
```

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error

get (all)

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/icaaction

Query-parameters:

attrs

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/icaaction?**attrs=property-name1,property-name2**

Use this query parameter to specify the resource details that you want to retrieve.

filter

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/icaaction?**filter=property-name1:property-val1,property-name2:property-val2**

Use this query-parameter to get the filtered set of icaaction resources configured on Citrix ADC. Filtering can be done on any of the properties of the resource.

view

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/icaaction?**view=summary**

Note: By default, the retrieved results are displayed in detail view (?view=detail).

pagination

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/icaaction?**pagesize=#no;pageno=#no**

Use this query-parameter to get the icaaction resources in chunks.

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "icaaction": [ {

"name":<String_value>,

"accessprofilename":<String_value>,

"latencyprofilename":<String_value>,

"hits":<Double_value>,

"referencecount":<Double_value>,

"undefhits":<Double_value>,

"isdefault":<Boolean_value>
```

```
}]}
```

get

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/icaaction/name_value<String>

Query-parameters:

attrs

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/icaaction/name_value<String>?**attrs=property-name1,property-name2**

Use this query parameter to specify the resource details that you want to retrieve.

view

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/icaaction/name_value<String>?**view=summary**

Note: By default, the retrieved results are displayed in detail view (?view=detail).

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "icaaction": [ {

"name":<String_value>,
```

```
"accessprofilename":<String_value>,

"latencyprofilename":<String_value>,

"hits":<Double_value>,

"referencecount":<Double_value>,

"undefhits":<Double_value>,

"isdefault":<Boolean_value>

}]}
```

count

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/icaaction?**count=yes**

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "icaaction": [ { "__count": "#no"} ] }
```

icaglobal_binding

Binding object which returns the resources bound to icaglobal.

Properties

(click to see Operations)

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
icaglobal_icapolicy_binding	<icaglobal_icapolicy_binding[]>	Read-only	icapolicy that can be bound to icaglobal.

Operations

(click to see Properties)

GET| GET (ALL)

Some options that you can use for each operations:

- **Getting warnings in response:**NITRO allows you to get warnings in an operation by specifying the "warning" query parameter as "yes". For example, to get warnings while connecting to the Citrix ADC appliance, the URL is as follows:

  http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/login?warning=yes

  If any, the warnings are displayed in the response payload with the HTTP code "209 X-NITRO-WARNING".
- **Authenticated access for individual NITRO operations:**NITRO allows you to logon to the Citrix ADC appliance to perform individual operations. You can use this option instead of creating a NITRO session (using the login object) and then using that session to perform all operations,

  To do this, you must specify the username and password in the request header of the NITRO request as follows:

  X-NITRO-USER:<username>

  X-NITRO-PASS:<password>

  Note: In such cases, make sure that the request header DOES not include the following:

  Cookie:NITRO_AUTH_TOKEN=<tokenvalue>

`*Note:`*

Mandatory parameters are marked in redand placeholder content is marked in <green>.

get

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/icaglobal_binding

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "icaglobal_binding": [ {

"icaglobal_icapolicy_binding":<icaglobal_icapolicy_binding[]_value>

}]}
```

get (all)

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/icaglobal_binding

Query-parameters:

bulkbindings

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/icaglobal_binding?**bulkbindings=yes**

NITRO allows you to fetch bindings in bulk.

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "icaglobal_binding": [ {

"icaglobal_icapolicy_binding":<icaglobal_icapolicy_binding[]_value>

}]}
```

icaglobal_icapolicy_binding

Binding object showing the icapolicy that can be bound to icaglobal.

Properties

(click to see Operations)

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
priority	<Double>	Read-write	Specifies the priority of the policy.
globalbindtype	<String>	Read-write	. Default value: SYSTEM_GLOBAL Possible values = SYSTEM_GLOBAL, VPN_GLOBAL, RNAT_GLOBAL

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
policyname	<String>	Read-write	Name of the ICA policy.
gotopriorityexpression	<String>	Read-write	Expression specifying the priority of the next policy which will get evaluated if the current policy rule evaluates to TRUE.
type	<String>	Read-write	Global bind point for which to show detailed information about the policies bound to the bind point. Possible values = ICA_REQ_OVERRIDE, ICA_REQ_DEFAULT
flowtype	<Double>	Read-only	Flow type of the bound ICA policy.
numpol	<Double>	Read-only	The number of policies bound to the bindpoint.
__count	<Double>	Read-only	count parameter

Operations

(click to see Properties)

ADD:| DELETE:| GET| GET (ALL)| COUNT

Some options that you can use for each operations:

- **Getting warnings in response:**NITRO allows you to get warnings in an operation by specifying the "warning" query parameter as "yes". For example, to get warnings while connecting to the Citrix ADC appliance, the URL is as follows:

  http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/login?warning=yes

  If any, the warnings are displayed in the response payload with the HTTP code "209 X-NITRO-WARNING".

- **Authenticated access for individual NITRO operations:**NITRO allows you to logon to the Citrix ADC appliance to perform individual operations. You can use this option instead of creating a NITRO session (using the login object) and then using that session to perform all operations,

  To do this, you must specify the username and password in the request header of the NITRO request as follows:

  X-NITRO-USER:<username>

  X-NITRO-PASS:<password>

  Note: In such cases, make sure that the request header DOES not include the following:

  Cookie:NITRO_AUTH_TOKEN=<tokenvalue>

`*Note:`*

Mandatory parameters are marked in redand placeholder content is marked in <green>.

add

URL:http://<Citrix-ADC-IP-address(NSIP)/nitro/v1/config/icaglobal_icapolicy_binding

HTTP Method:PUT

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Content-Type:application/json

Request Payload:

```
{
```

```
"icaglobal_icapolicy_binding":{

<b>"policyname":<String_value>,

</b><b>"priority":<Double_value>,

</b>"gotopriorityexpression":<String_value>,

"type":<String_value>

}}
```

Response:

HTTP Status Code on Success: 201 Created HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error

delete

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/icaglobal_icapolicy_binding

Query-parameters:

args

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/icaglobal_icapolicy_binding?**args=policyname:<String_value>,type:<String_value>,priority:<Double_value>**

HTTP Method:DELETE

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue>

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error

get

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/icaglobal_icapolicy_binding

Query-parameters:

filter

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/icaglobal_icapolicy_binding?**filter=property-name1:property-value1,property-name2:property-value2**

Use this query-parameter to get the filtered set of icaglobal_icapolicy_binding resources configured on Citrix ADC. Filtering can be done on any of the properties of the resource.

pagination

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/icaglobal_icapolicy_binding?**pagesize=#no;pageno=#no**

Use this query-parameter to get the icaglobal_icapolicy_binding resources in chunks.

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "icaglobal_icapolicy_binding": [ {

"priority":<Double_value>,

"globalbindtype":<String_value>,
```

```
"policyname":<String_value>,

"gotopriorityexpression":<String_value>,

"type":<String_value>,

"flowtype":<Double_value>,

"numpol":<Double_value>

}]}
```

get (all)

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/icaglobal_icapolicy_binding

Query-parameters:

bulkbindings

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/icaglobal_icapolicy_binding?**bulkbindings=yes**

NITRO allows you to fetch bindings in bulk.

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "icaglobal_icapolicy_binding": [ {

"priority":<Double_value>,

"globalbindtype":<String_value>,

"policyname":<String_value>,

"gotopriorityexpression":<String_value>,

"type":<String_value>,

"flowtype":<Double_value>,

"numpol":<Double_value>

}]}
```

count

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/icaglobal_icapolicy_binding?**count=yes**

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "icaglobal_icapolicy_binding": [ { "__count": "#no"} ] }
```

icalatencyprofile

Configuration for Profile for Latency monitoring resource.

Properties

(click to see Operations)

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
name	<String>	Read-write	Name for the ICA latencyprofile. Must begin with a letter, number, or the underscore character (_), and must contain only letters, numbers, and the hyphen (-), period (.) pound (#), space ( ), at (@), equals (=), colon (:), and underscore characters. Cannot be changed after the ICA latency profile is added. The following requirement applies only to the Citrix ADC CLI: If the name includes one or more spaces, enclose the

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
			name in double or single quotation marks (for example, "my ica l7latencyprofile" or 'my ica l7latencyprofile'). Minimum length = 1
l7latencymonitoring	<String>	Read-write	Enable/Disable L7 Latency monitoring for L7 latency notifications. Default value: DISABLED Possible values = ENABLED, DISABLED
l7latencythresholdfactor	<Double>	Read-write	L7 Latency threshold factor. This is the factor by which the active latency should be greater than the minimum observed value to determine that the latency is high and may need to be reported. Default value: 4 Minimum value = 2 Maximum value = 65535
l7latencywaittime	<Double>	Read-write	L7 Latency Wait time. This is the time

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
			for which the Citrix ADC waits after the threshold is exceeded before it sends out a Notification to the Insight Center. Default value: 20 Minimum value = 1 Maximum value = 65535
l7latencynotifyinterval	<Double>	Read-write	L7 Latency Notify Interval. This is the interval at which the Citrix ADC sends out notifications to the Insight Center after the wait time has passed. Default value: 20 Minimum value = 1 Maximum value = 65535
l7latencymaxnotifycount	<Double>	Read-write	L7 Latency Max notify Count. This is the upper limit on the number of notifications sent to the Insight Center within an interval where the Latency is above the threshold. Default value: 5

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
			Minimum value = 1 Maximum value = 65535
refcnt	<Double>	Read-only	Number of entities using this l7latencyprofile.
isdefault	<Boolean>	Read-only	A value of true is returned if it is a default l7latencyprofile.
__count	<Double>	Read-only	count parameter

Operations

(click to see Properties)

ADD| DELETE| UPDATE| UNSET| GET (ALL)| GET| COUNT

Some options that you can use for each operations:

- **Getting warnings in response:**NITRO allows you to get warnings in an operation by specifying the "warning" query parameter as "yes". For example, to get warnings while connecting to the Citrix ADC appliance, the URL is as follows:

  http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/login?warning=yes

  If any, the warnings are displayed in the response payload with the HTTP code "209 X-NITRO-WARNING".

- **Authenticated access for individual NITRO operations:**NITRO allows you to logon to the Citrix ADC appliance to perform individual operations. You can use this option instead of creating a NITRO session (using the login object) and then using that session to perform all operations,

  To do this, you must specify the username and password in the request header of the NITRO request as follows:

  X-NITRO-USER:<username>

X-NITRO-PASS:<password>

Note: In such cases, make sure that the request header DOES not include the following:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue>

`*Note:`*

Mandatory parameters are marked in redand placeholder content is marked in <green>.

add

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/icalatencyprofile

HTTP Method:POST

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Content-Type:application/json

Request Payload:

```
{"icalatencyprofile":{

<b>"name":<String_value>,

</b>"l7latencymonitoring":<String_value>,

"l7latencythresholdfactor":<Double_value>,

"l7latencywaittime":<Double_value>,

"l7latencynotifyinterval":<Double_value>,

"l7latencymaxnotifycount":<Double_value>

}}
```

Response:

HTTP Status Code on Success: 201 Created HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error

delete

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/icalatencyprofile/name_value<String>

HTTP Method:DELETE

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue>

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error

update

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/icalatencyprofile

HTTP Method:PUT

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Content-Type:application/json

Request Payload:

```
{"icalatencyprofile":{

<b>"name":<String_value>,

</b>"l7latencymonitoring":<String_value>,

"l7latencythresholdfactor":<Double_value>,

"l7latencywaittime":<Double_value>,

"l7latencynotifyinterval":<Double_value>,
```

```
"l7latencymaxnotifycount":<Double_value>

}}
```

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error

unset

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/icalatencyprofile?**action=unset**

HTTP Method:POST

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Content-Type:application/json

Request Payload:

```
{"icalatencyprofile":{

<b>"name":<String_value>,

</b>"l7latencymonitoring":true,

"l7latencythresholdfactor":true,

"l7latencywaittime":true,

"l7latencynotifyinterval":true,

"l7latencymaxnotifycount":true

}}
```

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error

get (all)

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/icalatencyprofile

Query-parameters:

attrs

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/icalatencyprofile?**attrs=property-name1,property-name2**

Use this query parameter to specify the resource details that you want to retrieve.

filter

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/icalatencyprofile?**filter=property-name1:property-val1,property-name2:property-val2**

Use this query-parameter to get the filtered set of icalatencyprofile resources configured on Citrix ADC. Filtering can be done on any of the properties of the resource.

view

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/icalatencyprofile?**view=summary**

Note: By default, the retrieved results are displayed in detail view (?view=detail).

pagination

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/icalatencyprofile?**pagesize=#no;pageno=#no**

Use this query-parameter to get the icalatencyprofile resources in chunks.

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "icalatencyprofile": [ {

"name":<String_value>,

"l7latencymonitoring":<String_value>,

"l7latencythresholdfactor":<Double_value>,

"l7latencywaittime":<Double_value>,

"l7latencynotifyinterval":<Double_value>,

"l7latencymaxnotifycount":<Double_value>,

"refcnt":<Double_value>,

"isdefault":<Boolean_value>

}]}
```

get

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/icalatencyprofile/name_value<String>

Query-parameters:

attrs

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/icalatencyprofile/name_value<String>?**attrs=property-name1,property-name2**

Use this query parameter to specify the resource details that you want to retrieve.

view

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/icalatencyprofile/name_value<String>?**view=summary**

Note: By default, the retrieved results are displayed in detail view (?view=detail).

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "icalatencyprofile": [ {

"name":<String_value>,

"l7latencymonitoring":<String_value>,

"l7latencythresholdfactor":<Double_value>,

"l7latencywaittime":<Double_value>,

"l7latencynotifyinterval":<Double_value>,

"l7latencymaxnotifycount":<Double_value>,

"refcnt":<Double_value>,

"isdefault":<Boolean_value>

}]}
```

count

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/icalatencyprofile?**count=yes**

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "icalatencyprofile": [ { "__count": "#no"} ] }
```

icaparameter

Configuration for Config Parameters for NS ICA resource.

Properties

(click to see Operations)

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
enablesronhafailover	<String>	Read-write	Enable/Disable Session Reliability on HA failover. The default value is No. Default value: NO

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
			Possible values = YES, NO
hdxinsightnonnsap	<String>	Read-write	Enable/Disable HDXInsight for Non NSAP ICA Sessions. The default value is Yes. Default value: YES Possible values = YES, NO

Operations

(click to see Properties)

UPDATE| UNSET| GET (ALL)

Some options that you can use for each operations:

- **Getting warnings in response:**NITRO allows you to get warnings in an operation by specifying the "warning" query parameter as "yes". For example, to get warnings while connecting to the Citrix ADC appliance, the URL is as follows:

  http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/login?warning=yes

  If any, the warnings are displayed in the response payload with the HTTP code "209 X-NITRO-WARNING".

- **Authenticated access for individual NITRO operations:**NITRO allows you to logon to the Citrix ADC appliance to perform individual operations. You can use this option instead of creating a NITRO session (using the login object) and then using that session to perform all operations,

  To do this, you must specify the username and password in the request header of the NITRO request as follows:

  X-NITRO-USER:<username>

  X-NITRO-PASS:<password>

Note: In such cases, make sure that the request header DOES not include the following:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue>

`*Note:` *

Mandatory parameters are marked in redand placeholder content is marked in <green>.

update

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/icaparameter

HTTP Method:PUT

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Content-Type:application/json

Request Payload:

```
{"icaparameter":{

"enablesronhafailover":<String_value>,

"hdxinsightnonnsap":<String_value>

}}
```

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error

unset

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/icaparameter?**action=unset**

HTTP Method:POST

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Content-Type:application/json

Request Payload:

```
{"icaparameter":{

"enablesronhafailover":true,

"hdxinsightnonnsap":true

}}
```

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error

get (all)

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/icaparameter

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "icaparameter": [ {

"enablesronhafailover":<String_value>,
```

```
"hdxinsightnonnsap":<String_value>

}]}
```

icapolicy

Configuration for ICA policy resource.

Properties

(click to see Operations)

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
name	<String>	Read-write	Name for the policy. Must begin with an ASCII alphabetic or underscore (_) character, and must contain only ASCII alphanumeric, underscore, hash (#), period (.), space, colon (:), at (@), equals (=), and hyphen (-) characters. The following requirement applies only to the Citrix ADC CLI: If the name includes one or more spaces, enclose the name in double or single quotation marks (for example,

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
			"my ica policy" or 'my ica policy').
rule	<String>	Read-write	Expression or other value against which the traffic is evaluated. Must be a Boolean expression. The following requirements apply only to the Citrix ADC CLI: * If the expression includes one or more spaces, enclose the entire expression in double quotation marks. * If the expression itself includes double quotation marks, escape the quotations by using the \ character. * Alternatively, you can use single quotation marks to enclose the rule, in which case you do not have to escape the double quotation marks.

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
action	<String>	Read-write	Name of the ica action to be associated with this policy.
comment	<String>	Read-write	Any type of information about this ICA policy.
logaction	<String>	Read-write	Name of the messagelog action to use for requests that match this policy.
newname	<String>	Read-write	New name for the policy. Must begin with an ASCII alphabetic or underscore (_)character, and must contain only ASCII alphanumeric, underscore, hash (#), period (.), s pace, colon (:), at (@), equals (=), and hyphen (-) characters. The following requirement applies only to the Citrix ADC CLI: If the name includes one or more spaces, enclose the name in double or single quotation marks (for example,

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
			"my ica policy" or 'my ica policy'). Minimum length = 1
hits	<Double>	Read-only	Number of hits.
undefhits	<Double>	Read-only	Number of policy UNDEF hits.
__count	<Double>	Read-only	count parameter

Operations

(click to see Properties)

ADD| DELETE| UPDATE| UNSET| RENAME| GET (ALL)| GET| COUNT

Some options that you can use for each operations:

- **Getting warnings in response:**NITRO allows you to get warnings in an operation by specifying the "warning" query parameter as "yes". For example, to get warnings while connecting to the Citrix ADC appliance, the URL is as follows:

  http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/login?warning=yes

  If any, the warnings are displayed in the response payload with the HTTP code "209 X-NITRO-WARNING".

- **Authenticated access for individual NITRO operations:**NITRO allows you to logon to the Citrix ADC appliance to perform individual operations. You can use this option instead of creating a NITRO session (using the login object) and then using that session to perform all operations,

  To do this, you must specify the username and password in the request header of the NITRO request as follows:

  X-NITRO-USER:<username>

  X-NITRO-PASS:<password>

  Note: In such cases, make sure that the request header DOES not include the following:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue>

`*Note:` *

Mandatory parameters are marked in redand placeholder content is marked in <green>.

add

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/icapolicy

HTTP Method:POST

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Content-Type:application/json

Request Payload:

```
{"icapolicy":{

<b>"name":<String_value>,

</b><b>"rule":<String_value>,

</b><b>"action":<String_value>,

</b>"comment":<String_value>,

"logaction":<String_value>

}}
```

Response:

HTTP Status Code on Success: 201 Created HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error

delete

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/icapolicy/name_value<String>

HTTP Method:DELETE

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue>

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error

update

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/icapolicy

HTTP Method:PUT

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Content-Type:application/json

Request Payload:

```
{"icapolicy":{

<b>"name":<String_value>,

</b>"rule":<String_value>,

"action":<String_value>,

"comment":<String_value>,

"logaction":<String_value>

}}
```

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error

unset

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/icapolicy?**action=unset**

HTTP Method:POST

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Content-Type:application/json

Request Payload:

```
{"icapolicy":{

<b>"name":<String_value>,

</b>"comment":true,

"logaction":true

}}
```

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error

rename

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/icapolicy?**action=rename**

HTTP Method:POST

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Content-Type:application/json

Request Payload:

```
{"icapolicy":{

<b>"name":<String_value>,

</b><b>"newname":<String_value>

</b>}}
```

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error

get (all)

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/icapolicy

Query-parameters:

attrs

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/icapolicy?**attrs=property-name1,property-name2**

Use this query parameter to specify the resource details that you want to retrieve.

filter

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/icapolicy?**filter=property-name1:property-val1,property-name2:property-val2**

Use this query-parameter to get the filtered set of icapolicy resources configured on Citrix ADC. Filtering can be done on any of the properties of the resource.

view

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/icapolicy?**view=summary**

Note: By default, the retrieved results are displayed in detail view (?view=detail).

pagination

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/icapolicy?**pagesize=#no;pageno=#no**

Use this query-parameter to get the icapolicy resources in chunks.

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "icapolicy": [ {

"name":<String_value>,

"rule":<String_value>,

"action":<String_value>,

"hits":<Double_value>,

"undefhits":<Double_value>,

"comment":<String_value>,

"logaction":<String_value>

}]}
```

get

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/icapolicy/name_value<String>

Query-parameters:

attrs

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/icapolicy/name_value<String>?**attrs=property-name1,property-name2**

Use this query parameter to specify the resource details that you want to retrieve.

view

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/icapolicy/name_value<String>?**view=summary**

Note: By default, the retrieved results are displayed in detail view (?view=detail).

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "icapolicy": [ {

"name":<String_value>,

"rule":<String_value>,
```

```
"action":<String_value>,

"hits":<Double_value>,

"undefhits":<Double_value>,

"comment":<String_value>,

"logaction":<String_value>

}]}
```

count

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/icapolicy?**count=yes**

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "icapolicy": [ { "__count": "#no"} ] }
```

icapolicy_binding

Binding object which returns the resources bound to icapolicy.

Properties

(click to see Operations)

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
name	<String>	Read-write	Name of the policy about which to display detailed information.
icapolicy_vpnvserver_binding	<icapolicy_vpnvserver_binding[]>	Read-only	vpnvserver that can be bound to icapolicy.
icapolicy_icaglobal_binding	<icapolicy_icaglobal_binding[]>	Read-only	icaglobal that can be bound to icapolicy.
icapolicy_crvserver_binding	<icapolicy_crvserver_binding[]>	Read-only	crvserver that can be bound to icapolicy.

Operations

(click to see Properties)

GET| GET (ALL)

Some options that you can use for each operations:

- **Getting warnings in response:**NITRO allows you to get warnings in an operation by specifying the "warning" query parameter as "yes". For example, to get warnings while connecting to the Citrix ADC appliance, the URL is as follows:

  http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/login?warning=yes

  If any, the warnings are displayed in the response payload with the HTTP code "209 X-NITRO-WARNING".
- **Authenticated access for individual NITRO operations:**NITRO allows you to logon to the Citrix ADC appliance to perform individual operations. You can use this option instead of creating a NITRO session (using the login object) and then using that session to perform all operations,

To do this, you must specify the username and password in the request header of the NITRO request as follows:

X-NITRO-USER:<username>

X-NITRO-PASS:<password>

Note: In such cases, make sure that the request header DOES not include the following:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue>

`*Note:`*

Mandatory parameters are marked in redand placeholder content is marked in <green>.

get

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/icapolicy_binding/name_value<String>

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "icapolicy_binding": [ {

"name":<String_value>,

"icapolicy_crvserver_binding":<icapolicy_crvserver_binding[]_value>,

"icapolicy_icaglobal_binding":<icapolicy_icaglobal_binding[]_value>,

"icapolicy_vpnvserver_binding":<icapolicy_vpnvserver_binding[]_value>
```

```
}]}
```

get (all)

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/icapolicy_binding

Query-parameters:

bulkbindings

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/icapolicy_binding?**bulkbindings=yes**

NITRO allows you to fetch bindings in bulk.

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "icapolicy_binding": [ {

"name":<String_value>,

"icapolicy_crvserver_binding":<icapolicy_crvserver_binding[]_value>,

"icapolicy_icaglobal_binding":<icapolicy_icaglobal_binding[]_value>,

"icapolicy_vpnvserver_binding":<icapolicy_vpnvserver_binding[]_value>

}]}
```

icapolicy_crvserver_binding

Binding object showing the crvserver that can be bound to icapolicy.

Properties

(click to see Operations)

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
boundto	<String>	Read-write	Location where policy is bound.
name	<String>	Read-write	Name of the policy about which to display detailed information.
priority	<Double>	Read-only	Specifies the priority of the policy.
gotopriorityexpression	<String>	Read-only	Expression specifying the priority of the next policy which will get evaluated if the current policy rule evaluates to TRUE.
activepolicy	<Integer>	Read-only	Indicates whether policy is bound or not.
__count	<Double>	Read-only	count parameter

Operations

(click to see Properties)

GET| GET (ALL)| COUNT

Some options that you can use for each operations:

- **Getting warnings in response:**NITRO allows you to get warnings in an operation by specifying the "warning" query parameter as "yes". For example, to get warnings while connecting to the Citrix ADC appliance, the URL is as follows:

  http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/login?warning=yes

  If any, the warnings are displayed in the response payload with the HTTP code "209 X-NITRO-WARNING".

- **Authenticated access for individual NITRO operations:**NITRO allows you to logon to the Citrix ADC appliance to perform individual operations. You can use this option instead of creating a NITRO session (using the login object) and then using that session to perform all operations,

  To do this, you must specify the username and password in the request header of the NITRO request as follows:

  X-NITRO-USER:<username>

  X-NITRO-PASS:<password>

  Note: In such cases, make sure that the request header DOES not include the following:

  Cookie:NITRO_AUTH_TOKEN=<tokenvalue>

`*Note:`*

Mandatory parameters are marked in redand placeholder content is marked in <green>.

get

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/icapolicy_crvserver_binding/name_value<String>

Query-parameters:

filter

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/icapolicy_crvserver_binding/name_value<String>?**filter=property-name1:property-value1,property-name2:property-value2**

Use this query-parameter to get the filtered set of icapolicy_crvserver_binding resources configured on Citrix ADC. Filtering can be done on any of the properties of the resource.

pagination

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/icapolicy_crvserver_binding/name_value<String>?**pagesize=#no;pageno=#no**

Use this query-parameter to get the icapolicy_crvserver_binding resources in chunks.

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "icapolicy_crvserver_binding": [ {

"boundto":<String_value>,

"name":<String_value>,

"priority":<Double_value>,

"gotopriorityexpression":<String_value>,

"activepolicy":<Integer_value>

}]}
```

get (all)

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/icapolicy_crvserver_binding

Query-parameters:

bulkbindings

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/icapolicy_crvserver_binding?**bulkbindings=yes**

NITRO allows you to fetch bindings in bulk.

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "icapolicy_crvserver_binding": [ {

"boundto":<String_value>,

"name":<String_value>,

"priority":<Double_value>,

"gotopriorityexpression":<String_value>,

"activepolicy":<Integer_value>

}]}
```

count

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/icapolicy_crvserver_binding/name_value<String>?**count=yes**

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{"icapolicy_crvserver_binding": [ { "__count": "#no"} ] }
```

icapolicy_icaglobal_binding

Binding object showing the icaglobal that can be bound to icapolicy.

Properties

(click to see Operations)

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
boundto	<String>	Read-write	Location where policy is bound.
name	<String>	Read-write	Name of the policy about which to display detailed information.
priority	<Double>	Read-only	Specifies the priority of the policy.

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
gotopriorityexpression	<String>	Read-only	Expression specifying the priority of the next policy which will get evaluated if the current policy rule evaluates to TRUE.
activepolicy	<Integer>	Read-only	Indicates whether policy is bound or not.
__count	<Double>	Read-only	count parameter

Operations

(click to see Properties)

GET| GET (ALL)| COUNT

Some options that you can use for each operations:

- **Getting warnings in response:**NITRO allows you to get warnings in an operation by specifying the "warning" query parameter as "yes". For example, to get warnings while connecting to the Citrix ADC appliance, the URL is as follows:

  http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/login?warning=yes

  If any, the warnings are displayed in the response payload with the HTTP code "209 X-NITRO-WARNING".

- **Authenticated access for individual NITRO operations:**NITRO allows you to logon to the Citrix ADC appliance to perform individual operations. You can use this option instead of creating a NITRO session (using the login object) and then using that session to perform all operations,

  To do this, you must specify the username and password in the request header of the NITRO request as follows:

  X-NITRO-USER:<username>

  X-NITRO-PASS:<password>

Note: In such cases, make sure that the request header DOES not include the following:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue>

`*Note:`*

Mandatory parameters are marked in redand placeholder content is marked in <green>.

get

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/icapolicy_icaglobal_binding/name_value<String>

Query-parameters:

filter

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/icapolicy_icaglobal_binding/name_value<String>?**filter=property-name1:property-value1,property-name2:property-value2**

Use this query-parameter to get the filtered set of icapolicy_icaglobal_binding resources configured on Citrix ADC. Filtering can be done on any of the properties of the resource.

pagination

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/icapolicy_icaglobal_binding/name_value<String>?**pagesize=#no;pageno=#no**

Use this query-parameter to get the icapolicy_icaglobal_binding resources in chunks.

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "icapolicy_icaglobal_binding": [ {

"boundto":<String_value>,

"name":<String_value>,

"priority":<Double_value>,

"gotopriorityexpression":<String_value>,

"activepolicy":<Integer_value>

}]}
```

get (all)

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/icapolicy_icaglobal_binding

Query-parameters:

bulkbindings

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/icapolicy_icaglobal_binding?**bulkbindings=yes**

NITRO allows you to fetch bindings in bulk.

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "icapolicy_icaglobal_binding": [ {

"boundto":<String_value>,

"name":<String_value>,

"priority":<Double_value>,

"gotopriorityexpression":<String_value>,

"activepolicy":<Integer_value>

}]}
```

count

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/icapolicy_icaglobal_binding/name_value<String>?**count=yes**

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{"icapolicy_icaglobal_binding": [ { "__count": "#no"} ] }
```

icapolicy_vpnvserver_binding

Binding object showing the vpnvserver that can be bound to icapolicy.

Properties

(click to see Operations)

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
boundto	<String>	Read-write	Location where policy is bound.
name	<String>	Read-write	Name of the policy about which to display detailed information.
priority	<Double>	Read-only	Specifies the priority of the policy.
gotopriorityexpression	<String>	Read-only	Expression specifying the priority of the next policy which will get evaluated if the current policy rule evaluates to TRUE.
activepolicy	<Integer>	Read-only	Indicates whether policy is bound or not.
__count	<Double>	Read-only	count parameter

Operations

(click to see Properties)

GET| GET (ALL)| COUNT

Some options that you can use for each operations:

- **Getting warnings in response:**NITRO allows you to get warnings in an operation by specifying the "warning" query parameter as "yes". For example, to get warnings while connecting to the Citrix ADC appliance, the URL is as follows:

  http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/login?warning=yes

  If any, the warnings are displayed in the response payload with the HTTP code "209 X-NITRO-WARNING".

- **Authenticated access for individual NITRO operations:**NITRO allows you to logon to the Citrix ADC appliance to perform individual operations. You can use this option instead of creating a NITRO session (using the login object) and then using that session to perform all operations,

  To do this, you must specify the username and password in the request header of the NITRO request as follows:

  X-NITRO-USER:<username>

  X-NITRO-PASS:<password>

  Note: In such cases, make sure that the request header DOES not include the following:

  Cookie:NITRO_AUTH_TOKEN=<tokenvalue>

`*Note:`*

Mandatory parameters are marked in redand placeholder content is marked in <green>.

get

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/icapolicy_vpnvserver_binding/name_value<String>

Query-parameters:

filter

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/icapolicy_vpnvserver_binding/name_value<String>?**filter=property-name1:property-value1,property-name2:property-value2**

Use this query-parameter to get the filtered set of icapolicy_vpnvserver_binding resources configured on Citrix ADC. Filtering can be done on any of the properties of the resource.

pagination

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/icapolicy_vpnvserver_binding/name_value<String>?**pagesize=#no;pageno=#no**

Use this query-parameter to get the icapolicy_vpnvserver_binding resources in chunks.

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "icapolicy_vpnvserver_binding": [ {

"boundto":<String_value>,

"name":<String_value>,

"priority":<Double_value>,

"gotopriorityexpression":<String_value>,

"activepolicy":<Integer_value>

}]}
```

get (all)

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/icapolicy_vpnvserver_binding

Query-parameters:

bulkbindings

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/icapolicy_vpnvserver_binding?**bulkbindings=yes**

NITRO allows you to fetch bindings in bulk.

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "icapolicy_vpnvserver_binding": [ {

"boundto":<String_value>,

"name":<String_value>,

"priority":<Double_value>,

"gotopriorityexpression":<String_value>,

"activepolicy":<Integer_value>

}]}
```

count

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/icapolicy_vpnvserver_binding/name_value<String>?**count=yes**

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{"icapolicy_vpnvserver_binding": [ { "__count": "#no"} ] }
```

Integrated caching

Integrated Caching

Integrated Caching Configuration. The Integrated Caching feature is used to cache static and dynamic web application data.

NAME	DESCRIPTION
cachecontentgroup	Configuration for Integrated Cache content group
cacheforwardproxy	Configuration for forward proxy
cacheglobal_binding	Binding object showing the resources that can be bound to cacheglobal
cacheglobal_cachepolicy_binding	Binding object showing the cachepolicy that can be bound to cacheglobal
cacheobject	Configuration for cache object

NAME	DESCRIPTION
cacheparameter	Configuration for cache parameter
cachepolicy	Configuration for Integrated Cache policy
cachepolicy_binding	Binding object showing the resources that can be bound to cachepolicy
cachepolicy_cacheglobal_binding	Binding object showing the cacheglobal that can be bound to cachepolicy
cachepolicy_cachepolicylabel_binding	Binding object showing the cachepolicylabel that can be bound to cachepolicy
cachepolicy_csvserver_binding	Binding object showing the csvserver that can be bound to cachepolicy
cachepolicy_lbvserver_binding	Binding object showing the lbvserver that can be bound to cachepolicy
cachepolicylabel	Configuration for cache policy label
cachepolicylabel_binding	Binding object showing the resources that can be bound to cachepolicylabel
cachepolicylabel_cachepolicy_binding	Binding object showing the cachepolicy that can be bound to cachepolicylabel
cachepolicylabel_policybinding_binding	Binding object showing the policybinding that can be bound to cachepolicylabel
cacheselector	Configuration for cache selector

cachecontentgroup

Configuration for Integrated Cache content group resource.

Properties

(click to see Operations)

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
name	<String>	Read-write	Name for the content group. Must begin with an ASCII alphabetic or underscore (_) character, and must contain only ASCII alphanumeric, underscore, hash (#), period (.), space, colon (:), at (@), equals (=), and hyphen (-) characters. Cannot be changed after the content group is created. Minimum length = 1
weakposrelexpiry	<Double>	Read-write	Relative expiry time, in seconds, for expiring positive responses with response codes between 200 and 399. Cannot be used in combination with other Expiry attributes. Similar to -relExpiry but has lower precedence. Minimum value = 0 Maximum value = 31536000
heurexpiryparam	<Double>	Read-write	Heuristic expiry time, in percent of the duration, since the

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
			object was last modified. Minimum value = 0 Maximum value = 100
relexpiry	<Double>	Read-write	Relative expiry time, in seconds, after which to expire an object cached in this content group. Minimum value = 0 Maximum value = 31536000
relexpirymillisec	<Double>	Read-write	Relative expiry time, in milliseconds, after which to expire an object cached in this content group. Minimum value = 0 Maximum value = 86400000
absexpiry	<String[]>	Read-write	Local time, up to 4 times a day, at which all objects in the content group must expire. CLI Users: For example, to specify that the objects in the content group should expire by 11:00 PM, type the

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
			following command: add cache contentgroup <contentgroup name> -absexpiry 23:00 To specify that the objects in the content group should expire at 10:00 AM, 3 PM, 6 PM, and 11:00 PM, type: add cache contentgroup <contentgroup name> -absexpiry 10:00 15:00 18:00 23:00.
absexpirygmt	<String[]>	Read-write	Coordinated Universal Time (GMT), up to 4 times a day, when all objects in the content group must expire.
weaknegrelexpiry	<Double>	Read-write	Relative expiry time, in seconds, for expiring negative responses. This value is used only if the expiry time cannot be determined from any other source. It is applicable only to the following status

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
			codes: 307, 403, 404, and 410. Minimum value = 0 Maximum value = 31536000
hitparams	<String[]>	Read-write	Parameters to use for parameterized hit evaluation of an object. Up to 128 parameters can be specified. Mutually exclusive with the Hit Selector parameter. Minimum length = 1
invalparams	<String[]>	Read-write	Parameters for parameterized invalidation of an object. You can specify up to 8 parameters. Mutually exclusive with invalSelector. Minimum length = 1
ignoreparamvaluecase	<String>	Read-write	Ignore case when comparing parameter values during parameterized hit evaluation. (Parameter value case is ignored by default during parameterized invalidation.). Possible values = YES, NO

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
matchcookies	<String>	Read-write	Evaluate for parameters in the cookie header also. Possible values = YES, NO
invalrestrictedtohost	<String>	Read-write	Take the host header into account during parameterized invalidation. Possible values = YES, NO
polleverytime	<String>	Read-write	Always poll for the objects in this content group. That is, retrieve the objects from the origin server whenever they are requested. Default value: NO Possible values = YES, NO
ignorereloadreq	<String>	Read-write	Ignore any request to reload a cached object from the origin server. To guard against Denial of Service attacks, set this parameter to YES. For RFC-compliant behavior, set it to NO.

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
			Default value: YES Possible values = YES, NO
removecookies	<String>	Read-write	Remove cookies from responses. Default value: YES Possible values = YES, NO
prefetch	<String>	Read-write	Attempt to refresh objects that are about to go stale. Default value: YES Possible values = YES, NO
prefetchperiod	<Double>	Read-write	Time period, in seconds before an object's calculated expiry time, during which to attempt prefetch. Minimum value = 0 Maximum value = 4294967294
prefetchperiodmillisec	<Double>	Read-write	Time period, in milliseconds before an object's calculated expiry time, during which to attempt prefetch. Minimum value = 0 Maximum value = 4294967290

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
prefetchmaxpending	<Double>	Read-write	Maximum number of outstanding prefetches that can be queued for the content group. Minimum value = 0 Maximum value = 4294967294
flashcache	<String>	Read-write	Perform flash cache. Mutually exclusive with Poll Every Time (PET) on the same content group. Default value: NO Possible values = YES, NO
expireatlastbyte	<String>	Read-write	Force expiration of the content immediately after the response is downloaded (upon receipt of the last byte of the response body). Applicable only to positive responses. Default value: NO Possible values = YES, NO
insertvia	<String>	Read-write	Insert a Via header into the response. Default value: YES

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
			Possible values = YES, NO
insertage	<String>	Read-write	Insert an Age header into the response. An Age header contains information about the age of the object, in seconds, as calculated by the integrated cache. Default value: YES Possible values = YES, NO
insertetag	<String>	Read-write	Insert an ETag header in the response. With ETag header insertion, the integrated cache does not serve full responses on repeat requests. Default value: YES Possible values = YES, NO
cachecontrol	<String>	Read-write	Insert a Cache-Control header into the response. Minimum length = 1
quickabortsize	<Double>	Read-write	If the size of an object that is being downloaded is less than or equal to the quick abort value, and a client aborts during the download,

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
			the cache stops downloading the response. If the object is larger than the quick abort size, the cache continues to download the response. Default value: 4194303 Minimum value = 0 Maximum value = 4194303
minressize	<Double>	Read-write	Minimum size of a response that can be cached in this content group. Default minimum response size is 0. Minimum value = 0 Maximum value = 2097151
maxressize	<Double>	Read-write	Maximum size of a response that can be cached in this content group. Default value: 80 Minimum value = 0 Maximum value = 2097151
memlimit	<Double>	Read-write	Maximum amount of memory that the

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
			cache can use. The effective limit is based on the available memory of the Citrix ADC. Default value: 65536
ignorereqcachinghdrs	<String>	Read-write	Ignore Cache-Control and Pragma headers in the incoming request. Default value: YES Possible values = YES, NO
minhits	<Integer>	Read-write	Number of hits that qualifies a response for storage in this content group. Default value: 0
alwaysevalpolicies	<String>	Read-write	Force policy evaluation for each response arriving from the origin server. Cannot be set to YES if the Prefetch parameter is also set to YES. Default value: NO Possible values = YES, NO
persistha	<String>	Read-write	Setting persistHA to YES causes IC to save objects in contentgroup to

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
			Secondary node in HA deployment. Default value: NO Possible values = YES, NO
pinned	<String>	Read-write	Do not flush objects from this content group under memory pressure. Default value: NO Possible values = YES, NO
lazydnsresolve	<String>	Read-write	Perform DNS resolution for responses only if the destination IP address in the request does not match the destination IP address of the cached response. Default value: YES Possible values = YES, NO
hitselector	<String>	Read-write	Selector for evaluating whether an object gets stored in a particular content group. A selector is an abstraction for a collection of PIXL expressions.

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
invalselector	<String>	Read-write	Selector for invalidating objects in the content group. A selector is an abstraction for a collection of PIXL expressions.
type	<String>	Read-write	The type of the content group. Default value: HTTP Possible values = HTTP, MYSQL, MSSQL
query	<String>	Read-write	Query string specifying individual objects to flush from this group by using parameterized invalidation. If this parameter is not set, all objects are flushed from the group. Minimum length = 1
host	<String>	Read-write	Flush only objects that belong to the specified host. Do not use except with parameterized invalidation. Also, the Invalidation Restricted to Host parameter for the group must be set to YES. Minimum length = 1

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
selectorvalue	<String>	Read-write	Value of the selector to be used for flushing objects from the content group. Requires that an invalidation selector be configured for the content group. Minimum length = 1
tosecondary	<String>	Read-write	content group whose objects are to be sent to secondary. Default value: NO Possible values = YES, NO
flags	<Double>	Read-only	Flags.
prefetchcur	<Double>	Read-only	Current outstanding prefetches.
memusage	<Double>	Read-only	Current memory usage.
memdusage	<Double>	Read-only	Current disk memory usage.
disklimit	<Double>	Read-only	Maximum amount of disk that the cache can use. The effective limit is based on the available memory of the Citrix ADC. Default value: 0
cachenon304hits	<Double>	Read-only	Cache non 304 hits.
cache304hits	<Double>	Read-only	Cache 304 hits.

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
cachecells	<Double>	Read-only	Number of cells.
cachegroupincarnation	<Double>	Read-only	Cache group incarnation.
persist	<String>	Read-only	Setting persist to YES causes IC to save objects in contentgroup to disk. Default value: NO Possible values = YES, NO
policyname	<String[]>	Read-only	Active cache policies refering to this group.
cachenuminvalpolicy	<Double>	Read-only	Number of active Invalidation policies refering to this group.
markercells	<Double>	Read-only	Numbers of marker cells in this group.
__count	<Double>	Read-only	count parameter

Operations

(click to see Properties)

ADD| DELETE| UPDATE| UNSET| EXPIRE| FLUSH| SAVE| GET (ALL)| GET| COUNT

Some options that you can use for each operations:

- **Getting warnings in response:**NITRO allows you to get warnings in an operation by specifying the "warning" query parameter as "yes". For example, to get warnings while connecting to the Citrix ADC appliance, the URL is as follows:

  http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/login?warning=yes

  If any, the warnings are displayed in the response payload with the HTTP code "209 X-NITRO-WARNING".

- **Authenticated access for individual NITRO operations:**NITRO allows you to logon to the Citrix ADC appliance to perform individual operations. You can use this option instead of creating a NITRO session (using the login object) and then using that session to perform all operations,

  To do this, you must specify the username and password in the request header of the NITRO request as follows:

  X-NITRO-USER:<username>

  X-NITRO-PASS:<password>

  Note: In such cases, make sure that the request header DOES not include the following:

  Cookie:NITRO_AUTH_TOKEN=<tokenvalue>

`*Note:`*

Mandatory parameters are marked in redand placeholder content is marked in <green>.

add

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/cachecontentgroup

HTTP Method:POST

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Content-Type:application/json

Request Payload:

```
{"cachecontentgroup":{

<b>"name":<String_value>,

</b>"weakposrelexpiry":<Double_value>,

"heurexpiryparam":<Double_value>,

"relexpiry":<Double_value>,

"relexpirymillisec":<Double_value>,
```

```
"absexpiry":<String[]_value>,

"absexpirygmt":<String[]_value>,

"weaknegrelexpiry":<Double_value>,

"hitparams":<String[]_value>,

"invalparams":<String[]_value>,

"ignoreparamvaluecase":<String_value>,

"matchcookies":<String_value>,

"invalrestrictedtohost":<String_value>,

"polleverytime":<String_value>,

"ignorereloadreq":<String_value>,

"removecookies":<String_value>,

"prefetch":<String_value>,

"prefetchperiod":<Double_value>,

"prefetchperiodmillisec":<Double_value>,

"prefetchmaxpending":<Double_value>,

"flashcache":<String_value>,

"expireatlastbyte":<String_value>,

"insertvia":<String_value>,

"insertage":<String_value>,

"insertetag":<String_value>,

"cachecontrol":<String_value>,

"quickabortsize":<Double_value>,
```

```
"minressize":<Double_value>,

"maxressize":<Double_value>,

"memlimit":<Double_value>,

"ignorereqcachinghdrs":<String_value>,

"minhits":<Integer_value>,

"alwaysevalpolicies":<String_value>,

"persistha":<String_value>,

"pinned":<String_value>,

"lazydnsresolve":<String_value>,

"hitselector":<String_value>,

"invalselector":<String_value>,

"type":<String_value>

}}
```

Response:

HTTP Status Code on Success: 201 Created HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error

delete

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/cachecontentgroup/name_value<String>

HTTP Method:DELETE

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue>

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error

update

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/cachecontentgroup

HTTP Method:PUT

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Content-Type:application/json

Request Payload:

```
{"cachecontentgroup":{

<b>"name":<String_value>,

</b>"weakposrelexpiry":<Double_value>,

"heurexpiryparam":<Double_value>,

"relexpiry":<Double_value>,

"relexpirymillisec":<Double_value>,

"absexpiry":<String[]_value>,

"absexpirygmt":<String[]_value>,

"weaknegrelexpiry":<Double_value>,

"hitparams":<String[]_value>,

"invalparams":<String[]_value>,

"ignoreparamvaluecase":<String_value>,
```

```
"matchcookies":<String_value>,

"invalrestrictedtohost":<String_value>,

"polleverytime":<String_value>,

"ignorereloadreq":<String_value>,

"removecookies":<String_value>,

"prefetch":<String_value>,

"prefetchperiod":<Double_value>,

"prefetchperiodmillisec":<Double_value>,

"prefetchmaxpending":<Double_value>,

"flashcache":<String_value>,

"expireatlastbyte":<String_value>,

"insertvia":<String_value>,

"insertage":<String_value>,

"insertetag":<String_value>,

"cachecontrol":<String_value>,

"quickabortsize":<Double_value>,

"minressize":<Double_value>,

"maxressize":<Double_value>,

"memlimit":<Double_value>,

"ignorereqcachinghdrs":<String_value>,

"minhits":<Integer_value>,

"alwaysevalpolicies":<String_value>,
```

```
"persistha":<String_value>,

"pinned":<String_value>,

"lazydnsresolve":<String_value>,

"hitselector":<String_value>,

"invalselector":<String_value>

}}
```

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error

unset

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/cachecontentgroup?**action=unset**

HTTP Method:POST

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Content-Type:application/json

Request Payload:

```
{"cachecontentgroup":{

<b>"name":<String_value>,

</b>"weakposrelexpiry":true,

"heurexpiryparam":true,

"relexpiry":true,
```

```
"relexpirymillisec":true,

"absexpiry":true,

"absexpirygmt":true,

"weaknegrelexpiry":true,

"hitparams":true,

"invalparams":true,

"ignoreparamvaluecase":true,

"matchcookies":true,

"invalrestrictedtohost":true,

"polleverytime":true,

"ignorereloadreq":true,

"removecookies":true,

"prefetch":true,

"prefetchperiod":true,

"prefetchperiodmillisec":true,

"prefetchmaxpending":true,

"flashcache":true,

"expireatlastbyte":true,

"insertvia":true,

"insertage":true,

"insertetag":true,

"cachecontrol":true,
```

```
"quickabortsize":true,

"minressize":true,

"maxressize":true,

"memlimit":true,

"ignorereqcachinghdrs":true,

"minhits":true,

"alwaysevalpolicies":true,

"persistha":true,

"pinned":true,

"lazydnsresolve":true,

"hitselector":true,

"invalselector":true

}}
```

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error

expire

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/cachecontentgroup?**action=expire**

HTTP Method:POST

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Content-Type:application/json

Request Payload:

```
{"cachecontentgroup":{

<b>"name":<String_value>

</b>}}
```

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error

flush

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/cachecontentgroup?**action=flush**

HTTP Method:POST

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Content-Type:application/json

Request Payload:

```
{"cachecontentgroup":{

<b>"name":<String_value>,

</b>"query":<String_value>,

"host":<String_value>,

"selectorvalue":<String_value>

}}
```

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error

save

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/cachecontentgroup?**action=save**

HTTP Method:POST

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Content-Type:application/json

Request Payload:

```
{"cachecontentgroup":{

<b>"name":<String_value>,

</b>"tosecondary":<String_value>

}}
```

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error

get (all)

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/cachecontentgroup

Query-parameters:

attrs

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/cachecontentgroup?**attrs=property-name1,property-name2**

Use this query parameter to specify the resource details that you want to retrieve.

filter

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/cachecontentgroup?**filter=property-name1:property-val1,property-name2:property-val2**

Use this query-parameter to get the filtered set of cachecontentgroup resources configured on Citrix ADC. Filtering can be done on any of the properties of the resource.

view

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/cachecontentgroup?**view=summary**

Note: By default, the retrieved results are displayed in detail view (?view=detail).

pagination

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/cachecontentgroup?**pagesize=#no;pageno=#no**

Use this query-parameter to get the cachecontentgroup resources in chunks.

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "cachecontentgroup": [ {
```

```
"name":<String_value>,

"flags":<Double_value>,

"type":<String_value>,

"relexpiry":<Double_value>,

"relexpirymillisec":<Double_value>,

"absexpiry":<String[]_value>,

"absexpirygmt":<String[]_value>,

"heurexpiryparam":<Double_value>,

"weakposrelexpiry":<Double_value>,

"weaknegrelexpiry":<Double_value>,

"hitparams":<String[]_value>,

"invalparams":<String[]_value>,

"ignoreparamvaluecase":<String_value>,

"matchcookies":<String_value>,

"invalrestrictedtohost":<String_value>,

"polleverytime":<String_value>,

"ignorereloadreq":<String_value>,

"removecookies":<String_value>,

"prefetch":<String_value>,

"prefetchperiod":<Double_value>,

"prefetchperiodmillisec":<Double_value>,

"prefetchcur":<Double_value>,
```

```
"prefetchmaxpending":<Double_value>,

"flashcache":<String_value>,

"expireatlastbyte":<String_value>,

"insertvia":<String_value>,

"insertage":<String_value>,

"insertetag":<String_value>,

"cachecontrol":<String_value>,

"quickabortsize":<Double_value>,

"minressize":<Double_value>,

"maxressize":<Double_value>,

"memusage":<Double_value>,

"memdusage":<Double_value>,

"disklimit":<Double_value>,

"memlimit":<Double_value>,

"ignorereqcachinghdrs":<String_value>,

"cachenon304hits":<Double_value>,

"cache304hits":<Double_value>,

"cachecells":<Double_value>,

"cachegroupincarnation":<Double_value>,

"minhits":<Integer_value>,

"alwaysevalpolicies":<String_value>,

"persist":<String_value>,
```

```
"persistha":<String_value>,

"pinned":<String_value>,

"lazydnsresolve":<String_value>,

"hitselector":<String_value>,

"invalselector":<String_value>,

"policyname":<String[]_value>,

"cachenuminvalpolicy":<Double_value>,

"markercells":<Double_value>

}]}
```

get

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/cachecontentgroup/name_value<String>

Query-parameters:

attrs

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/cachecontentgroup/name_value<String>?**attrs=property-name1,property-name2**

Use this query parameter to specify the resource details that you want to retrieve.

view

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/cachecontentgroup/name_value<String>?**view=summary**

Note: By default, the retrieved results are displayed in detail view (?view=detail).

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "cachecontentgroup": [ {

"name":<String_value>,

"flags":<Double_value>,

"type":<String_value>,

"relexpiry":<Double_value>,

"relexpirymillisec":<Double_value>,

"absexpiry":<String[]_value>,

"absexpirygmt":<String[]_value>,

"heurexpiryparam":<Double_value>,

"weakposrelexpiry":<Double_value>,

"weaknegrelexpiry":<Double_value>,

"hitparams":<String[]_value>,

"invalparams":<String[]_value>,

"ignoreparamvaluecase":<String_value>,

"matchcookies":<String_value>,

"invalrestrictedtohost":<String_value>,
```

```
"polleverytime":<String_value>,

"ignorereloadreq":<String_value>,

"removecookies":<String_value>,

"prefetch":<String_value>,

"prefetchperiod":<Double_value>,

"prefetchperiodmillisec":<Double_value>,

"prefetchcur":<Double_value>,

"prefetchmaxpending":<Double_value>,

"flashcache":<String_value>,

"expireatlastbyte":<String_value>,

"insertvia":<String_value>,

"insertage":<String_value>,

"insertetag":<String_value>,

"cachecontrol":<String_value>,

"quickabortsize":<Double_value>,

"minressize":<Double_value>,

"maxressize":<Double_value>,

"memusage":<Double_value>,

"memdusage":<Double_value>,

"disklimit":<Double_value>,

"memlimit":<Double_value>,

"ignorereqcachinghdrs":<String_value>,
```

```
"cachenon304hits":<Double_value>,

"cache304hits":<Double_value>,

"cachecells":<Double_value>,

"cachegroupincarnation":<Double_value>,

"minhits":<Integer_value>,

"alwaysevalpolicies":<String_value>,

"persist":<String_value>,

"persistha":<String_value>,

"pinned":<String_value>,

"lazydnsresolve":<String_value>,

"hitselector":<String_value>,

"invalselector":<String_value>,

"policyname":<String[]_value>,

"cachenuminvalpolicy":<Double_value>,

"markercells":<Double_value>

}]}
```

count

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/cachecontentgroup?**count=yes**

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "cachecontentgroup": [ { "__count": "#no"} ] }
```

cacheforwardproxy

Configuration for forward proxy resource.

Properties

(click to see Operations)

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
ipaddress	<String>	Read-write	IP address of the Citrix ADC or a cache server for which the cache acts as a proxy. Requests coming to the Citrix ADC with the configured IP address are forwarded to the particular address, without involving the Integrated Cache in any way. Minimum length = 1

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
port	<Integer>	Read-write	Port on the Citrix ADC or a server for which the cache acts as a proxy. Minimum value = 1 Range 1 - 65535 * in CLI is represented as 65535 in NITRO API
__count	<Double>	Read-only	count parameter

Operations

(click to see Properties)

ADD| DELETE| GET (ALL)| COUNT

Some options that you can use for each operations:

- **Getting warnings in response:**NITRO allows you to get warnings in an operation by specifying the "warning" query parameter as "yes". For example, to get warnings while connecting to the Citrix ADC appliance, the URL is as follows:

  http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/login?warning=yes

  If any, the warnings are displayed in the response payload with the HTTP code "209 X-NITRO-WARNING".

- **Authenticated access for individual NITRO operations:**NITRO allows you to logon to the Citrix ADC appliance to perform individual operations. You can use this option instead of creating a NITRO session (using the login object) and then using that session to perform all operations,

  To do this, you must specify the username and password in the request header of the NITRO request as follows:

  X-NITRO-USER:<username>

X-NITRO-PASS:<password>

Note: In such cases, make sure that the request header DOES not include the following:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue>

`*Note:`*

Mandatory parameters are marked in redand placeholder content is marked in <green>.

add

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/cacheforwardproxy

HTTP Method:POST

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Content-Type:application/json

Request Payload:

```
{"cacheforwardproxy":{

<b>"ipaddress":<String_value>,

</b><b>"port":<Integer_value>

</b>}}
```

Response:

HTTP Status Code on Success: 201 Created HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error

delete

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/cacheforwardproxy/ipaddress_value<String>

HTTP Method:DELETE

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue>

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error

get (all)

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/cacheforwardproxy

Query-parameters:

attrs

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/cacheforwardproxy?**attrs=property-name1,property-name2**

Use this query parameter to specify the resource details that you want to retrieve.

filter

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/cacheforwardproxy?**filter=property-name1:property-val1,property-name2:property-val2**

Use this query-parameter to get the filtered set of cacheforwardproxy resources configured on Citrix ADC. Filtering can be done on any of the properties of the resource.

view

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/cacheforwardproxy?**view=summary**

Note: By default, the retrieved results are displayed in detail view (?view=detail).

pagination

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/cacheforwardproxy?**pagesize=#no;pageno=#no**

Use this query-parameter to get the cacheforwardproxy resources in chunks.

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "cacheforwardproxy": [ {

"ipaddress":<String_value>,

"port":<Integer_value>

}]}
```

count

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/cacheforwardproxy?**count=yes**

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "cacheforwardproxy": [ { "__count": "#no"} ] }
```

cacheglobal_binding

Binding object which returns the resources bound to cacheglobal.

Properties

(click to see Operations)

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
cacheglobal_cachepolicy_binding	<cacheglobal_cachepolicy_binding[]>	Read-only	cachepolicy that can be bound to cacheglobal.

Operations

(click to see Properties)

GET| GET (ALL)

Some options that you can use for each operations:

- **Getting warnings in response:**NITRO allows you to get warnings in an operation by specifying the "warning" query parameter as "yes". For example, to get warnings while connecting to the Citrix ADC appliance, the URL is as follows:

  http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/login?warning=yes

  If any, the warnings are displayed in the response payload with the HTTP code "209 X-NITRO-WARNING".

- **Authenticated access for individual NITRO operations:**NITRO allows you to logon to the Citrix ADC appliance to perform individual operations. You can use this option instead of creating a NITRO session (using the login object) and then using that session to perform all operations,

  To do this, you must specify the username and password in the request header of the NITRO request as follows:

X-NITRO-USER:<username>

X-NITRO-PASS:<password>

Note: In such cases, make sure that the request header DOES not include the following:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue>

`*Note:`*

Mandatory parameters are marked in redand placeholder content is marked in <green>.

get

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/cacheglobal_binding

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "cacheglobal_binding": [ {

"cacheglobal_cachepolicy_binding":<cacheglobal_cachepolicy_binding[]_value
>

}]}
```

get (all)

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/cacheglobal_binding

Query-parameters:

bulkbindings

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/cacheglobal_binding?**bulkbindings=yes**

NITRO allows you to fetch bindings in bulk.

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "cacheglobal_binding": [ {

"cacheglobal_cachepolicy_binding":<cacheglobal_cachepolicy_binding[]_value
>

}]}
```

cacheglobal_cachepolicy_binding

Binding object showing the cachepolicy that can be bound to cacheglobal.

Properties

(click to see Operations)

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
priority	<Double>	Read-write	Specifies the priority of the policy.
globalbindtype	<String>	Read-write	. Default value: SYSTEM_GLOBAL Possible values = SYSTEM_GLOBAL, VPN_GLOBAL, RNAT_GLOBAL
gotopriorityexpression	<String>	Read-write	Expression specifying the priority of the next policy which will get evaluated if the current policy rule evaluates to TRUE.
policy	<String>	Read-write	Name of the cache policy.
type	<String>	Read-write	The bind point to which policy is bound. When you specify the type, detailed information about that bind point appears. Possible values = REQ_OVERRIDE, REQ_DEFAULT, RES_OVERRIDE, RES_DEFAULT
precededefrules	<String>	Read-write	Specify whether this policy should be evaluated.

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
			Default value: NO Possible values = YES, NO
labeltype	<String>	Read-write	Type of policy label to invoke. Possible values = reqvserver, resvserver, policylabel
labelname	<String>	Read-write	Name of the label to invoke if the current policy rule evaluates to TRUE. (To invoke a label associated with a virtual server, specify the name of the virtual server.).
invoke	<Boolean>	Read-write	Invoke policies bound to a virtual server or a user-defined policy label. After the invoked policies are evaluated, the flow returns to the policy with the next priority. Applicable only to default-syntax policies.
numpol	<Double>	Read-only	The number of policies bound to the bindpoint.
flowtype	<Double>	Read-only	flowtype of the bound cache policy.

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
__count	<Double>	Read-only	count parameter

Operations

(click to see Properties)

ADD:| DELETE:| GET| GET (ALL)| COUNT

Some options that you can use for each operations:

- **Getting warnings in response:**NITRO allows you to get warnings in an operation by specifying the "warning" query parameter as "yes". For example, to get warnings while connecting to the Citrix ADC appliance, the URL is as follows:

  http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/login?warning=yes

  If any, the warnings are displayed in the response payload with the HTTP code "209 X-NITRO-WARNING".
- **Authenticated access for individual NITRO operations:**NITRO allows you to logon to the Citrix ADC appliance to perform individual operations. You can use this option instead of creating a NITRO session (using the login object) and then using that session to perform all operations,

  To do this, you must specify the username and password in the request header of the NITRO request as follows:

  X-NITRO-USER:<username>

  X-NITRO-PASS:<password>

  Note: In such cases, make sure that the request header DOES not include the following:

  Cookie:NITRO_AUTH_TOKEN=<tokenvalue>

`*Note:`*

Mandatory parameters are marked in redand placeholder content is marked in <green>.

add

URL:http://<Citrix-ADC-IP-address(NSIP)/nitro/v1/config/cacheglobal_cachepolicy_binding

HTTP Method:PUT

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Content-Type:application/json

Request Payload:

```
{

"cacheglobal_cachepolicy_binding":{

<b>"policy":<String_value>,

</b><b>"priority":<Double_value>,

</b>"precededefrules":<String_value>,

"gotopriorityexpression":<String_value>,

"type":<String_value>,

"invoke":<Boolean_value>,

"labeltype":<String_value>,

"labelname":<String_value>

}}
```

Response:

HTTP Status Code on Success: 201 Created HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error

delete

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/cacheglobal_cachepolicy_binding

Query-parameters:

args

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/cacheglobal_cachepolicy_binding?**args=policy:<String_value>,type:<String_value>,priority:<Double_value>**

HTTP Method:DELETE

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue>

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error

get

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/cacheglobal_cachepolicy_binding

Query-parameters:

filter

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/cacheglobal_cachepolicy_binding?**filter=property-name1:property-value1,property-name2:property-value2**

Use this query-parameter to get the filtered set of cacheglobal_cachepolicy_binding resources configured on Citrix ADC. Filtering can be done on any of the properties of the resource.

pagination

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/cacheglobal_cachepolicy_binding?**pagesize=#no;pageno=#no**

Use this query-parameter to get the cacheglobal_cachepolicy_binding resources in chunks.

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "cacheglobal_cachepolicy_binding": [ {

"priority":<Double_value>,

"globalbindtype":<String_value>,

"gotopriorityexpression":<String_value>,

"policy":<String_value>,

"type":<String_value>,

"precededefrules":<String_value>,

"labeltype":<String_value>,

"labelname":<String_value>,

"invoke":<Boolean_value>,

"numpol":<Double_value>,

"flowtype":<Double_value>

}]}
```

get (all)

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/cacheglobal_cachepolicy_binding

Query-parameters:

bulkbindings

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/cacheglobal_cachepolicy_binding?**bulkbindings=yes**

NITRO allows you to fetch bindings in bulk.

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "cacheglobal_cachepolicy_binding": [ {

"priority":<Double_value>,

"globalbindtype":<String_value>,

"gotopriorityexpression":<String_value>,

"policy":<String_value>,

"type":<String_value>,

"precededefrules":<String_value>,

"labeltype":<String_value>,

"labelname":<String_value>,

"invoke":<Boolean_value>,

"numpol":<Double_value>,
```

```
"flowtype":<Double_value>

}]}
```

count

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/cacheglobal_cachepolicy_binding?**count=yes**

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "cacheglobal_cachepolicy_binding": [ { "__count": "#no"} ] }
```

cacheobject

Configuration for cache object resource.

Properties

(click to see Operations)

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
url	<String>	Read-write	URL of the particular object whose details

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
			is required. Parameter "host" must be specified along with the URL. Minimum length = 1
locator	<Double>	Read-write	ID of the cached object.
httpstatus	<Double>	Read-write	HTTP status of the object.
host	<String>	Read-write	Host name of the object. Parameter "url" must be specified. Minimum length = 1
port	<Integer>	Read-write	Host port of the object. You must also set the Host parameter. Default value: 80 Minimum value = 1
groupname	<String>	Read-write	Name of the content group to which the object belongs. It will display only the objects belonging to the specified content group. You must also set the Host parameter.
httpmethod	<String>	Read-write	HTTP request method that caused the object to be stored.

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
			Default value: GET Possible values = GET, POST
group	<String>	Read-write	Name of the content group whose objects should be listed.
ignoremarkerobjects	<String>	Read-write	Ignore marker objects. Marker objects are created when a response exceeds the maximum or minimum response size for the content group or has not yet received the minimum number of hits for the content group. Possible values = ON, OFF
includenotreadyobjects	<String>	Read-write	Include responses that have not yet reached a minimum number of hits before being cached. Possible values = ON, OFF
nodeid	<Double>	Read-write	Unique number that identifies the cluster node. Minimum value = 0 Maximum value = 31

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
tosecondary	<String>	Read-write	Object will be saved onto Secondary. Default value: NO Possible values = YES, NO
cacheressize	<Double>	Read-only	Cache response size of the object.
cachereshdrsize	<Double>	Read-only	Cache response header size of the object.
cacheetag	<String>	Read-only	Cache ETag of the object.
httpstatusoutput	<Double>	Read-only	HTTP status of the object.
cachereslastmod	<String>	Read-only	Value of "Last-modified" header.
cachecontrol	<String>	Read-only	Cache-Control header of the object.
cacheresdate	<String>	Read-only	Value of "Date" header.
contentgroup	<String>	Read-only	Name of the contentgroup in which it is stored.
destipv46	<String>	Read-only	Destination IP.
destport	<Integer>	Read-only	Destination Port. Range 1 - 65535 * in CLI is represented as

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
			65535 in NITRO API
cachecellcomplex	<String>	Read-only	The state of the parameterized caching on this cell. Possible values = YES, NO
hitparams	<String[]>	Read-only	Parameterized hit evaluation of an object.
hitvalues	<String[]>	Read-only	Values of hitparams for this object.
cachecellreqtime	<Double>	Read-only	Required time of the cache cell object.
cachecellrestime	<Double>	Read-only	Response time to the cache cell object.
cachecurage	<Double>	Read-only	Current age of the cache object.
cachecellexpires	<Double>	Read-only	Expiry time of the cache cell object in seconds.
cachecellexpiresmillisec	<Double>	Read-only	Expiry time of the cache cell object in milliseconds.
flushed	<String>	Read-only	Specifies whether the object is flushed. Possible values = YES, NO
prefetch	<String>	Read-only	Specifies whether Integrated Cache should attempt to

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
			refresh an object immediately before it goes stale. Possible values = YES, NO
prefetchperiod	<Double>	Read-only	The duration in seconds of the period during which prefetch should be attempted, immediately before the object's calculated expiry time.
prefetchperiodmillisec	<Double>	Read-only	The duration in milliseconds of the period during which prefetch should be attempted, immediately before the object's calculated expiry time.
cachecellcurreaders	<Double>	Read-only	Current readers of the cache cell object.
cachecellcurmisses	<Double>	Read-only	Current misses of the cache cell object.
cachecellhits	<Double>	Read-only	Cache cell hits.
cachecellmisses	<Double>	Read-only	Cache cell misses.
cachecelldhits	<Double>	Read-only	Cache cell disk hits.
cachecellcompressionformat	<String>	Read-only	Compression format of this object. Identity

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
			means not compressed.
cachecellappfwmetadataexists	<String>	Read-only	AppFirewall cache object. Possible values = YES, NO
cachecellhttp11	<String>	Read-only	The state of the response to be HTTP/1.1. Possible values = YES, NO
cachecellweaketag	<String>	Read-only	The state of the weak HTTP Entity Tag in the cell. Possible values = YES, NO
cachecellresbadsize	<String>	Read-only	The marked state of the cell. Possible values = YES, NO
markerreason	<String>	Read-only	Reason for marking the cell. Possible values = Waiting for min hit, Response header is too big, Content-length header said response size is not in group size limit, Content-length response received more data, Content-length response received less data,

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
			Content-length response data is not in group size limit, Chunk response received more data, Chunk response data is not in group size limit, Bad chunk format, Fin terminated response data is not in group size limit
cachecellpolleverytime	<String>	Read-only	The state to poll every time on object. Possible values = YES, NO
cachecelletaginserted	<String>	Read-only	The state of the ETag to be inserted by IC for this object. Possible values = YES, NO
cachecellreadywithlastbyte	<String>	Read-only	The state of the complete arrived response. Possible values = YES, NO
cacheinmemory	<String>	Read-only	The cache data is available in memory. Possible values = YES, NO
cacheindisk	<String>	Read-only	The cache data is available in disk.

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
			Possible values = YES, NO
cacheinsecondary	<String>	Read-only	The cache data is available in secondary in HA deployment. Possible values = YES, NO
cachedirname	<String>	Read-only	The directory name used if saved.
cachefilename	<String>	Read-only	The filename used if saved.
cachecelldestipverified	<String>	Read-only	The state of DNS verification. Possible values = YES, NO
cachecellfwpxyobj	<String>	Read-only	The state of the object to be stored on a request to a forward proxy. Possible values = YES, NO
cachecellbasefile	<String>	Read-only	The state of delta being used as a basefile. Possible values = YES, NO
cachecellminhitflag	<String>	Read-only	The state of the minhit feature on this cell. Possible values = YES, NO

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
cachecellminhit	<Integer>	Read-only	Min hit value for the object.
policy	<Integer>	Read-only	Policy info for the object.
policyname	<String>	Read-only	Policy which created the object.
selectorname	<String[]>	Read-only	The hit selector for the object.
rule	<String[]>	Read-only	Selectors for this object.
selectorvalue	<String[]>	Read-only	The HTTP request method that caused the object to be stored.
cacheurls	<String>	Read-only	List of cache object URLs.
warnbucketskip	<Double>	Read-only	Bucket skipped warning.
totalobjs	<Double>	Read-only	Total objects.
httpcalloutcell	<String>	Read-only	Is it a http callout cell ?. Possible values = YES, NO
httpcalloutname	<String>	Read-only	Name of the http callout.
returntype	<String>	Read-only	Return type of the http callout. Possible values = BOOL, NUM, TEXT

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
httpcalloutresult	<String>	Read-only	First few bytes of http callout response.
locatorshow	<Double>	Read-only	ID of the cached object.
ceflags	<Double>	Read-only	Indicates state and type of cached cell.
__count	<Double>	Read-only	count parameter

Operations

(click to see Properties)

EXPIRE| FLUSH| SAVE| GET (ALL)| COUNT

Some options that you can use for each operations:

- **Getting warnings in response:**NITRO allows you to get warnings in an operation by specifying the "warning" query parameter as "yes". For example, to get warnings while connecting to the Citrix ADC appliance, the URL is as follows:

  http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/login?warning=yes

  If any, the warnings are displayed in the response payload with the HTTP code "209 X-NITRO-WARNING".

- **Authenticated access for individual NITRO operations:**NITRO allows you to logon to the Citrix ADC appliance to perform individual operations. You can use this option instead of creating a NITRO session (using the login object) and then using that session to perform all operations,

  To do this, you must specify the username and password in the request header of the NITRO request as follows:

  X-NITRO-USER:<username>

  X-NITRO-PASS:<password>

  Note: In such cases, make sure that the request header DOES not include the following:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue>

`*Note:` *

Mandatory parameters are marked in redand placeholder content is marked in <green>.

expire

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/cacheobject?**action=expire**

HTTP Method:POST

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Content-Type:application/json

Request Payload:

```
{"cacheobject":{

"locator":<Double_value>,

"url":<String_value>,

"host":<String_value>,

"port":<Integer_value>,

"groupname":<String_value>,

"httpmethod":<String_value>

}}
```

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error

flush

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/cacheobject?**action=flush**

HTTP Method:POST

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Content-Type:application/json

Request Payload:

```
{"cacheobject":{

"locator":<Double_value>,

"url":<String_value>,

"host":<String_value>,

"port":<Integer_value>,

"groupname":<String_value>,

"httpmethod":<String_value>

}}
```

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error

save

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/cacheobject?**action=save**

HTTP Method:POST

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Content-Type:application/json

Request Payload:

```
{"cacheobject":{

"locator":<Double_value>,

"tosecondary":<String_value>

}}
```

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error

get (all)

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/cacheobject

Query-parameters:

args

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/cacheobject?**args=url:<String_value>,locator:<Double_value>,httpstatus:<Double_value>,host:<String_value>,port:<**

Use this query-parameter to get cacheobject resources based on additional properties.

attrs

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/cacheobject?**attrs=property-name1,property-name2**

Use this query parameter to specify the resource details that you want to retrieve.

filter

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/cacheobject?**filter=property-name1:property-val1,property-name2:property-val2**

Use this query-parameter to get the filtered set of cacheobject resources configured on Citrix ADC. Filtering can be done on any of the properties of the resource.

view

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/cacheobject?**view=summary**

Note: By default, the retrieved results are displayed in detail view (?view=detail).

pagination

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/cacheobject?**pagesize=#no;pageno=#no**

Use this query-parameter to get the cacheobject resources in chunks.

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "cacheobject": [ {

url:<String_value>,locator:<Double_value>,httpstatus:<Double_value>,host:<
String_value>,port:<Integer_value>,groupname:<String_value>,httpmethod:<St
ring_value>,group:<String_value>,ignoremarkerobjects:<String_value>,includ
enotreadyobjects:<String_value>,nodeid:<Double_value>"cacheressize":<Doubl
e_value>,

"cachereshdrsize":<Double_value>,
```

```
"cacheetag":<String_value>,

"httpstatusoutput":<Double_value>,

"cachereslastmod":<String_value>,

"cachecontrol":<String_value>,

"cacheresdate":<String_value>,

"contentgroup":<String_value>,

"destip":<Double_value>,

"destipv46":<String_value>,

"destport":<Integer_value>,

"cachecellcomplex":<String_value>,

"hitparams":<String[]_value>,

"hitvalues":<String[]_value>,

"cachecellreqtime":<Double_value>,

"cachecellrestime":<Double_value>,

"cachecurage":<Double_value>,

"cachecellexpires":<Double_value>,

"cachecellexpiresmillisec":<Double_value>,

"flushed":<String_value>,

"prefetch":<String_value>,

"prefetchperiod":<Double_value>,

"prefetchperiodmillisec":<Double_value>,

"cachecellcurreaders":<Double_value>,
```

```
"cachecellcurmisses":<Double_value>,

"cachecellhits":<Double_value>,

"cachecellmisses":<Double_value>,

"cachecelldhits":<Double_value>,

"cachecellgzipcompressed":<String_value>,

"cachecelldeflatecompressed":<String_value>,

"cachecellcompressionformat":<String_value>,

"cachecellappfwmetadataexists":<String_value>,

"cachecellhttp11":<String_value>,

"cachecellweaketag":<String_value>,

"cachecellresbadsize":<String_value>,

"markerreason":<String_value>,

"cachecellpolleverytime":<String_value>,

"cachecelletaginserted":<String_value>,

"cachecellreadywithlastbyte":<String_value>,

"cacheinmemory":<String_value>,

"cacheindisk":<String_value>,

"cacheinsecondary":<String_value>,

"cachedirname":<String_value>,

"cachefilename":<String_value>,

"cachecelldestipverified":<String_value>,

"cachecellfwpxyobj":<String_value>,
```

```
"cachecellbasefile":<String_value>,

"cachecellminhitflag":<String_value>,

"cachecellminhit":<Integer_value>,

"policy":<Integer_value>,

"policyname":<String_value>,

"selectorname":<String[]_value>,

"rule":<String[]_value>,

"selectorvalue":<String[]_value>,

"cacheurls":<String_value>,

"warnbucketskip":<Double_value>,

"totalobjs":<Double_value>,

"httpcalloutcell":<String_value>,

"httpcalloutname":<String_value>,

"returntype":<String_value>,

"httpcalloutresult":<String_value>,

"locatorshow":<Double_value>,

"ceflags":<Double_value>

}]}
```

count

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/cacheobject?**count=yes**

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "cacheobject": [ { "__count": "#no"} ] }
```

cacheparameter

Configuration for cache parameter resource.

Properties

(click to see Operations)

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
memlimit	<Double>	Read-write	Amount of memory available for storing the cache objects. In practice, the amount of memory available for caching can be less than half the total memory of the Citrix ADC.
via	<String>	Read-write	String to include in the Via header. A Via header is inserted into all responses served from a

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
			content group if its Insert Via flag is set. Minimum length = 1
verifyusing	<String>	Read-write	Criteria for deciding whether a cached object can be served for an incoming HTTP request. Available settings function as follows: HOSTNAME - The URL, host name, and host port values in the incoming HTTP request header must match the cache policy. The IP address and the TCP port of the destination host are not evaluated. Do not use the HOSTNAME setting unless you are certain that no rogue client can access a rogue server through the cache. HOSTNAME_AND_IP - The URL, host name, host port in the incoming HTTP request header, and

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
			the IP address and TCP port of the destination server, must match the cache policy. DNS - The URL, host name and host port in the incoming HTTP request, and the TCP port must match the cache policy. The host name is used for DNS lookup of the destination server's IP address, and is compared with the set of addresses returned by the DNS lookup. Possible values = HOSTNAME, HOSTNAME_AND_IP, DNS
maxpostlen	<Double>	Read-write	Maximum number of POST body bytes to consider when evaluating parameters for a content group for which you have configured hit parameters and

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
			invalidation parameters. Default value: 4096 Minimum value = 0 Maximum value = 131072
prefetchmaxpending	<Double>	Read-write	Maximum number of outstanding prefetches in the Integrated Cache.
enablebypass	<String>	Read-write	Evaluate the request-time policies before attempting hit selection. If set to NO, an incoming request for which a matching object is found in cache storage results in a response regardless of the policy configuration. If the request matches a policy with a NOCACHE action, the request bypasses all cache processing. This parameter does not affect processing of requests that match any invalidation policy.

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
			Possible values = YES, NO
undefaction	<String>	Read-write	Action to take when a policy cannot be evaluated. Possible values = NOCACHE, RESET
enablehaobjpersist	<String>	Read-write	The HA object persisting parameter. When this value is set to YES, cache objects can be synced to Secondary in a HA deployment. If set to NO, objects will never be synced to Secondary node. Default value: NO Possible values = YES, NO
disklimit	<Double>	Read-only	The disk limit for the Integrated Cache.
maxdisklimit	<Double>	Read-only	The maximum value of the memory limit for the Integrated Cache.
memlimitactive	<Double>	Read-only	Active value of the memory limit for the Integrated Cache.
maxmemlimit	<Double>	Read-only	The maximum value of the memory limit for the Integrated Cache.

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
prefetchcur	<Double>	Read-only	Number of current outstanding prefetches in the IC.

Operations

(click to see Properties)

UPDATE| UNSET| GET (ALL)

Some options that you can use for each operations:

- **Getting warnings in response:**NITRO allows you to get warnings in an operation by specifying the "warning" query parameter as "yes". For example, to get warnings while connecting to the Citrix ADC appliance, the URL is as follows:

  http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/login?warning=yes

  If any, the warnings are displayed in the response payload with the HTTP code "209 X-NITRO-WARNING".

- **Authenticated access for individual NITRO operations:**NITRO allows you to logon to the Citrix ADC appliance to perform individual operations. You can use this option instead of creating a NITRO session (using the login object) and then using that session to perform all operations,

  To do this, you must specify the username and password in the request header of the NITRO request as follows:

  X-NITRO-USER:<username>

  X-NITRO-PASS:<password>

  Note: In such cases, make sure that the request header DOES not include the following:

  Cookie:NITRO_AUTH_TOKEN=<tokenvalue>

`*Note:`*

Mandatory parameters are marked in redand placeholder content is marked in <green>.

update

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/cacheparameter

HTTP Method:PUT

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Content-Type:application/json

Request Payload:

```
{"cacheparameter":{

"memlimit":<Double_value>,

"via":<String_value>,

"verifyusing":<String_value>,

"maxpostlen":<Double_value>,

"prefetchmaxpending":<Double_value>,

"enablebypass":<String_value>,

"undefaction":<String_value>,

"enablehaobjpersist":<String_value>

}}
```

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error

unset

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/cacheparameter?**action=unset**

HTTP Method:POST

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Content-Type:application/json

Request Payload:

```
{"cacheparameter":{
"memlimit":true,
"via":true,
"verifyusing":true,
"maxpostlen":true,
"prefetchmaxpending":true,
"enablebypass":true,
"undefaction":true,
"enablehaobjpersist":true
}}
```

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error

get (all)

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/cacheparameter

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "cacheparameter": [ {

"disklimit":<Double_value>,

"maxdisklimit":<Double_value>,

"memlimit":<Double_value>,

"memlimitactive":<Double_value>,

"maxmemlimit":<Double_value>,

"via":<String_value>,

"verifyusing":<String_value>,

"maxpostlen":<Double_value>,

"prefetchcur":<Double_value>,

"prefetchmaxpending":<Double_value>,

"enablebypass":<String_value>,

"undefaction":<String_value>,

"enablehaobjpersist":<String_value>

}]}
```

cachepolicy

Configuration for Integrated Cache policy resource.

Properties

(click to see Operations)

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
policyname	<String>	Read-write	Name for the policy. Must begin with an ASCII alphabetic or underscore (_) character, and must contain only ASCII alphanumeric, underscore, hash (#), period (.), space, colon (:), at (@), equals (=), and hyphen (-) characters. Can be changed after the policy is created. Minimum length = 1
rule	<String>	Read-write	Expression against which the traffic is evaluated. The following requirements apply only to the Citrix ADC CLI: * If the expression includes one or more spaces, enclose the entire expression in

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
			double quotation marks. * If the expression itself includes double quotation marks, escape the quotations by using the \ character. * Alternatively, you can use single quotation marks to enclose the rule, in which case you do not have to escape the double quotation marks.
action	<String>	Read-write	Action to apply to content that matches the policy. * CACHE or MAY_CACHE action - positive cachability policy * NOCACHE or MAY_NOCACHE action - negative cachability policy * INVAL action - Dynamic Invalidation Policy. Possible values = CACHE,

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
			NOCACHE, MAY_CACHE, MAY_NOCACHE, INVAL
storeingroup	<String>	Read-write	Name of the content group in which to store the object when the final result of policy evaluation is CACHE. The content group must exist before being mentioned here. Use the "show cache contentgroup" command to view the list of existing content groups. Minimum length = 1
invalgroups	<String[]>	Read-write	Content group(s) to be invalidated when the INVAL action is applied. Maximum number of content groups that can be specified is 16. Minimum length = 1
invalobjects	<String[]>	Read-write	Content groups(s) in which the objects will be invalidated if the action is INVAL. Minimum length = 1
undefaction	<String>	Read-write	Action to be performed when the result of rule

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
			evaluation is undefined. Possible values = NOCACHE, RESET
newname	<String>	Read-write	New name for the cache policy. Must begin with an ASCII alphabetic or underscore (_) character, and must contain only ASCII alphanumeric, underscore, hash (#), period (.), space, colon (:), at (@), equals (=), and hyphen (-) characters. Minimum length = 1
hits	<Double>	Read-only	Hits.
undefhits	<Double>	Read-only	Number of Undef hits.
flags	<Double>	Read-only	Flag.
__count	<Double>	Read-only	count parameter

Operations

(click to see Properties)

ADD| DELETE| UPDATE| UNSET| RENAME| GET (ALL)| GET| COUNT

Some options that you can use for each operations:

- **Getting warnings in response:**NITRO allows you to get warnings in an operation by specifying the "warning" query parameter as "yes". For example, to get warnings while connecting to the Citrix ADC appliance, the URL is as follows:

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/login?warning=yes

If any, the warnings are displayed in the response payload with the HTTP code "209 X-NITRO-WARNING".

- **Authenticated access for individual NITRO operations:**NITRO allows you to logon to the Citrix ADC appliance to perform individual operations. You can use this option instead of creating a NITRO session (using the login object) and then using that session to perform all operations,

  To do this, you must specify the username and password in the request header of the NITRO request as follows:

  X-NITRO-USER:<username>

  X-NITRO-PASS:<password>

  Note: In such cases, make sure that the request header DOES not include the following:

  Cookie:NITRO_AUTH_TOKEN=<tokenvalue>

`*Note:`*

Mandatory parameters are marked in redand placeholder content is marked in <green>.

add

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/cachepolicy

HTTP Method:POST

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Content-Type:application/json

Request Payload:

```
{"cachepolicy":{

<b>"policyname":<String_value>,

</b><b>"rule":<String_value>,

</b><b>"action":<String_value>,
```

```
</b>"storeingroup":<String_value>,

"invalgroups":<String[]_value>,

"invalobjects":<String[]_value>,

"undefaction":<String_value>

}}
```

Response:

HTTP Status Code on Success: 201 Created HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error

delete

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/cachepolicy/policyname_value<String>

HTTP Method:DELETE

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue>

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error

update

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/cachepolicy

HTTP Method:PUT

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Content-Type:application/json

Request Payload:

```
{"cachepolicy":{

<b>"policyname":<String_value>,

</b>"rule":<String_value>,

"action":<String_value>,

"storeingroup":<String_value>,

"invalgroups":<String[]_value>,

"invalobjects":<String[]_value>,

"undefaction":<String_value>

}}
```

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error

unset

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/cachepolicy?**action=unset**

HTTP Method:POST

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Content-Type:application/json

Request Payload:

```
{"cachepolicy":{
```

```
<b>"policyname":<String_value>,

</b>"storeingroup":true,

"invalgroups":true,

"invalobjects":true,

"undefaction":true

}}
```

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error

rename

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/cachepolicy?**action=rename**

HTTP Method:POST

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Content-Type:application/json

Request Payload:

```
{"cachepolicy":{

<b>"policyname":<String_value>,

</b><b>"newname":<String_value>

</b>}}
```

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error

get (all)

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/cachepolicy

Query-parameters:

attrs

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/cachepolicy?**attrs=property-name1,property-name2**

Use this query parameter to specify the resource details that you want to retrieve.

filter

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/cachepolicy?**filter=property-name1:property-val1,property-name2:property-val2**

Use this query-parameter to get the filtered set of cachepolicy resources configured on Citrix ADC. Filtering can be done on any of the properties of the resource.

view

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/cachepolicy?**view=summary**

Note: By default, the retrieved results are displayed in detail view (?view=detail).

pagination

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/cachepolicy?**pagesize=#no;pageno=#no**

Use this query-parameter to get the cachepolicy resources in chunks.

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "cachepolicy": [ {

"policyname":<String_value>,

"rule":<String_value>,

"action":<String_value>,

"storeingroup":<String_value>,

"invalgroups":<String[]_value>,

"invalobjects":<String[]_value>,

"hits":<Double_value>,

"undefaction":<String_value>,

"undefhits":<Double_value>,

"flags":<Double_value>,

"precededefrules":<String_value>

}]}
```

get

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/cachepolicy/policyname_value<String>

Query-parameters:

attrs

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/cachepolicy/policyname_value<String>?**attrs=property-name1,property-name2**

Use this query parameter to specify the resource details that you want to retrieve.

view

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/cachepolicy/policyname_value<String>?**view=summary**

Note: By default, the retrieved results are displayed in detail view (?view=detail).

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "cachepolicy": [ {

"policyname":<String_value>,

"rule":<String_value>,

"action":<String_value>,

"storeingroup":<String_value>,

"invalgroups":<String[]_value>,

"invalobjects":<String[]_value>,
```

```
"hits":<Double_value>,

"undefaction":<String_value>,

"undefhits":<Double_value>,

"flags":<Double_value>,

"precededefrules":<String_value>

}]}
```

count

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/cachepolicy?**count=yes**

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "cachepolicy": [ { "__count": "#no"} ] }
```

cachepolicylabel

Configuration for cache policy label resource.

Properties

(click to see Operations)

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
labelname	<String>	Read-write	Name for the label. Must begin with an ASCII alphabetic or underscore (_) character, and must contain only ASCII alphanumeric, underscore, hash (#), period (.), space, colon (:), at (@), equals (=), and hyphen (-) characters. Can be changed after the label is created.
evaluates	<String>	Read-write	When to evaluate policies bound to this label: request-time or response-time. Possible values = REQ, RES, MSSQL_REQ, MSSQL_RES, MYSQL_REQ, MYSQL_RES
newname	<String>	Read-write	New name for the cache-policy label. Must begin with an ASCII alphabetic or underscore (_) character, and must contain only ASCII alphanumeric, underscore, hash (#), period (.), space,

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
			colon (:), at (@), equals (=), and hyphen (-) characters. Minimum length = 1
numpol	<Double>	Read-only	Number of polices bound to label.
hits	<Double>	Read-only	Number of times policy label was invoked.
priority	<Double>	Read-only	Specifies the priority of the policy.
gotopriorityexpression	<String>	Read-only	Expression specifying the priority of the next policy which will get evaluated if the current policy rule evaluates to TRUE.
labeltype	<String>	Read-only	Type of policy label to invoke: an unnamed label associated with a virtual server, or user-defined policy label. Possible values = reqvserver, resvserver, policylabel
invoke_labelname	<String>	Read-only	Name of the policy label to invoke if the current policy rule evaluates to TRUE.

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
flowtype	<Double>	Read-only	Flowtype of the bound cache policy.
__count	<Double>	Read-only	count parameter

Operations

(click to see Properties)

ADD| DELETE| RENAME| GET (ALL)| GET| COUNT

Some options that you can use for each operations:

- **Getting warnings in response:**NITRO allows you to get warnings in an operation by specifying the "warning" query parameter as "yes". For example, to get warnings while connecting to the Citrix ADC appliance, the URL is as follows:

  http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/login?warning=yes

  If any, the warnings are displayed in the response payload with the HTTP code "209 X-NITRO-WARNING".

- **Authenticated access for individual NITRO operations:**NITRO allows you to logon to the Citrix ADC appliance to perform individual operations. You can use this option instead of creating a NITRO session (using the login object) and then using that session to perform all operations,

  To do this, you must specify the username and password in the request header of the NITRO request as follows:

  X-NITRO-USER:<username>

  X-NITRO-PASS:<password>

  Note: In such cases, make sure that the request header DOES not include the following:

  Cookie:NITRO_AUTH_TOKEN=<tokenvalue>

`*Note:`*

Mandatory parameters are marked in redand placeholder content is marked in <green>.

add

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/cachepolicylabel

HTTP Method:POST

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Content-Type:application/json

Request Payload:

```
{"cachepolicylabel":{

<b>"labelname":<String_value>,

</b><b>"evaluates":<String_value>

</b>}}
```

Response:

HTTP Status Code on Success: 201 Created HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error

delete

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/cachepolicylabel/labelname_value<String>

HTTP Method:DELETE

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue>

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error

rename

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/cachepolicylabel?**action=rename**

HTTP Method:POST

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Content-Type:application/json

Request Payload:

```
{"cachepolicylabel":{

<b>"labelname":<String_value>,

</b><b>"newname":<String_value>

</b>}}
```

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error

get (all)

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/cachepolicylabel

Query-parameters:

attrs

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/cachepolicylabel?**attrs=property-name1,property-name2**

Use this query parameter to specify the resource details that you want to retrieve.

filter

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/cachepolicylabel?**filter=property-name1:property-val1,property-name2:property-val2**

Use this query-parameter to get the filtered set of cachepolicylabel resources configured on Citrix ADC. Filtering can be done on any of the properties of the resource.

view

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/cachepolicylabel?**view=summary**

Note: By default, the retrieved results are displayed in detail view (?view=detail).

pagination

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/cachepolicylabel?**pagesize=#no;pageno=#no**

Use this query-parameter to get the cachepolicylabel resources in chunks.

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "cachepolicylabel": [ {

"labelname":<String_value>,

"evaluates":<String_value>,

"numpol":<Double_value>,
```

```
"hits":<Double_value>,

"priority":<Double_value>,

"gotopriorityexpression":<String_value>,

"labeltype":<String_value>,

"invoke_labelname":<String_value>,

"flowtype":<Double_value>

}]}
```

get

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/cachepolicylabel/labelname_value<String>

Query-parameters:

attrs

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/cachepolicylabel/labelname_value<String>?**attrs=property-name1,property-name2**

Use this query parameter to specify the resource details that you want to retrieve.

view

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/cachepolicylabel/labelname_value<String>?**view=summary**

Note: By default, the retrieved results are displayed in detail view (?view=detail).

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "cachepolicylabel": [ {

"labelname":<String_value>,

"evaluates":<String_value>,

"numpol":<Double_value>,

"hits":<Double_value>,

"priority":<Double_value>,

"gotopriorityexpression":<String_value>,

"labeltype":<String_value>,

"invoke_labelname":<String_value>,

"flowtype":<Double_value>

}]}
```

count

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/cachepolicylabel?**count=yes**

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "cachepolicylabel": [ { "__count": "#no"} ] }
```

cachepolicylabel_binding

Binding object which returns the resources bound to cachepolicylabel.

Properties

(click to see Operations)

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
labelname	<String>	Read-write	Name of the cache-policy label about which to display information.
cachepolicylabel_cachepolicy_binding	<cachepolicylabel_cachepolicy_binding[]>	Read-only	cachepolicy that can be bound to cachepolicylabel.
cachepolicylabel_policybinding_binding	<cachepolicylabel_policybinding_binding[]>	Read-only	policybinding that can be bound to cachepolicylabel.

Operations

(click to see Properties)

GET| GET (ALL)

Some options that you can use for each operations:

- **Getting warnings in response:**NITRO allows you to get warnings in an operation by specifying the "warning" query parameter as "yes". For example, to get warnings while connecting to the Citrix ADC appliance, the URL is as follows:

  http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/login?warning=yes

  If any, the warnings are displayed in the response payload with the HTTP code "209 X-NITRO-WARNING".

- **Authenticated access for individual NITRO operations:**NITRO allows you to logon to the Citrix ADC appliance to perform individual operations. You can use this option instead of creating a NITRO session (using the login object) and then using that session to perform all operations,

  To do this, you must specify the username and password in the request header of the NITRO request as follows:

  X-NITRO-USER:<username>

  X-NITRO-PASS:<password>

  Note: In such cases, make sure that the request header DOES not include the following:

  Cookie:NITRO_AUTH_TOKEN=<tokenvalue>

`*Note:`*

Mandatory parameters are marked in redand placeholder content is marked in <green>.

get

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/cachepolicylabel_binding/labelname_value<String>

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "cachepolicylabel_binding": [ {

"labelname":<String_value>,

"cachepolicylabel_policybinding_binding":<cachepolicylabel_policybinding_binding[]_value>,

"cachepolicylabel_cachepolicy_binding":<cachepolicylabel_cachepolicy_binding[]_value>

}]}
```

get (all)

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/cachepolicylabel_binding

Query-parameters:

bulkbindings

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/cachepolicylabel_binding?**bulkbindings=yes**

NITRO allows you to fetch bindings in bulk.

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "cachepolicylabel_binding": [ {

"labelname":<String_value>,

"cachepolicylabel_policybinding_binding":<cachepolicylabel_policybinding_binding[]_value>,

"cachepolicylabel_cachepolicy_binding":<cachepolicylabel_cachepolicy_binding[]_value>

}]}
```

cachepolicylabel_cachepolicy_binding

Binding object showing the cachepolicy that can be bound to cachepolicylabel.

Properties

(click to see Operations)

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
priority	<Double>	Read-write	Specifies the priority of the policy.
policyname	<String>	Read-write	Name of the cache policy to bind to the policy label.
labelname	<String>	Read-write	Name of the cache policy label to which to bind the policy.
invoke_labelname	<String>	Read-write	Name of the policy label to invoke if the current policy rule evaluates to TRUE.

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
gotopriorityexpression	<String>	Read-write	Expression specifying the priority of the next policy which will get evaluated if the current policy rule evaluates to TRUE.
invoke	<Boolean>	Read-write	Invoke policies bound to a virtual server or a user-defined policy label. After the invoked policies are evaluated, the flow returns to the policy with the next-lower priority.
labeltype	<String>	Read-write	Type of policy label to invoke: an unnamed label associated with a virtual server, or user-defined policy label. Possible values = reqvserver, resvserver, policylabel
__count	<Double>	Read-write	count parameter

Operations

(click to see Properties)

ADD:| DELETE:| GET| GET (ALL)| COUNT

Some options that you can use for each operations:

- **Getting warnings in response:**NITRO allows you to get warnings in an operation by specifying the "warning" query parameter as "yes". For example, to get warnings while connecting to the Citrix ADC appliance, the URL is as follows:

  http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/login?warning=yes

  If any, the warnings are displayed in the response payload with the HTTP code "209 X-NITRO-WARNING".

- **Authenticated access for individual NITRO operations:**NITRO allows you to logon to the Citrix ADC appliance to perform individual operations. You can use this option instead of creating a NITRO session (using the login object) and then using that session to perform all operations,

  To do this, you must specify the username and password in the request header of the NITRO request as follows:

  X-NITRO-USER:<username>

  X-NITRO-PASS:<password>

  Note: In such cases, make sure that the request header DOES not include the following:

  Cookie:NITRO_AUTH_TOKEN=<tokenvalue>

`*Note:`*

Mandatory parameters are marked in redand placeholder content is marked in <green>.

add

URL:http://<Citrix-ADC-IP-address(NSIP)/nitro/v1/config/cachepolicylabel_cachepolicy_binding

HTTP Method:PUT

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Content-Type:application/json

Request Payload:

```
{
```

```
"cachepolicylabel_cachepolicy_binding":{

<b>"labelname":<String_value>,

</b><b>"policyname":<String_value>,

</b><b>"priority":<Double_value>,

</b>"gotopriorityexpression":<String_value>,

"invoke":<Boolean_value>,

"labeltype":<String_value>,

"invoke_labelname":<String_value>

}}
```

Response:

HTTP Status Code on Success: 201 Created HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error

delete

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/cachepolicylabel_cachepolicy_binding/labelname_value<String>

Query-parameters:

args

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/cachepolicylabel_cachepolicy_binding/labelname_value<String>?**args=policyname:<String_value>,priority:<Double_value>**

HTTP Method:DELETE

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue>

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error

get

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/cachepolicylabel_cachepolicy_binding/labelname_value<String>

Query-parameters:

filter

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/cachepolicylabel_cachepolicy_binding/labelname_value<String>?**filter=property-name1:property-value1,property-name2:property-value2**

Use this query-parameter to get the filtered set of cachepolicylabel_cachepolicy_binding resources configured on Citrix ADC. Filtering can be done on any of the properties of the resource.

pagination

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/cachepolicylabel_cachepolicy_binding/labelname_value<String>?**pagesize=#no;pageno=#no**

Use this query-parameter to get the cachepolicylabel_cachepolicy_binding resources in chunks.

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "cachepolicylabel_cachepolicy_binding": [ {

"priority":<Double_value>,

"policyname":<String_value>,

"labelname":<String_value>,

"invoke_labelname":<String_value>,

"gotopriorityexpression":<String_value>,

"invoke":<Boolean_value>,

"labeltype":<String_value>

}]}
```

get (all)

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/cachepolicylabel_cachepolicy_binding

Query-parameters:

bulkbindings

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/cachepolicylabel_cachepolicy_binding?**bulkbindings=yes**

NITRO allows you to fetch bindings in bulk.

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "cachepolicylabel_cachepolicy_binding": [ {

"priority":<Double_value>,

"policyname":<String_value>,

"labelname":<String_value>,

"invoke_labelname":<String_value>,

"gotopriorityexpression":<String_value>,

"invoke":<Boolean_value>,

"labeltype":<String_value>

}]}
```

count

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/cachepolicylabel_cachepolicy_binding/labelname_value<String>?**count=yes**

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{"cachepolicylabel_cachepolicy_binding": [ { "__count": "#no"} ] }
```

cachepolicylabel_policybinding_binding

Binding object showing the policybinding that can be bound to cachepolicylabel.

Properties

(click to see Operations)

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
priority	<Double>	Read-write	Specifies the priority of the policy.
policyname	<String>	Read-write	Name of the cache policy to bind to the policy label.
labelname	<String>	Read-write	Name of the cache policy label to which to bind the policy.
invoke_labelname	<String>	Read-write	Name of the policy label to invoke if the current policy rule evaluates to TRUE.
gotopriorityexpression	<String>	Read-write	Expression specifying the priority of the next policy which will get

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
			evaluated if the current policy rule evaluates to TRUE.
invoke	<Boolean>	Read-write	Invoke policies bound to a virtual server or a user-defined policy label. After the invoked policies are evaluated, the flow returns to the policy with the next-lower priority.
labeltype	<String>	Read-write	Type of policy label to invoke: an unnamed label associated with a virtual server, or user-defined policy label. Possible values = reqvserver, resvserver, policylabel
__count	<Double>	Read-write	count parameter

Operations

(click to see Properties)

GET| GET (ALL)| COUNT

Some options that you can use for each operations:

- **Getting warnings in response:**NITRO allows you to get warnings in an operation by specifying the "warning" query parameter as "yes". For example, to get warnings while connecting to the Citrix ADC appliance, the URL is as follows:

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/login?warning=yes

If any, the warnings are displayed in the response payload with the HTTP code "209 X-NITRO-WARNING".

- **Authenticated access for individual NITRO operations:**NITRO allows you to logon to the Citrix ADC appliance to perform individual operations. You can use this option instead of creating a NITRO session (using the login object) and then using that session to perform all operations,

  To do this, you must specify the username and password in the request header of the NITRO request as follows:

  X-NITRO-USER:<username>

  X-NITRO-PASS:<password>

  Note: In such cases, make sure that the request header DOES not include the following:

  Cookie:NITRO_AUTH_TOKEN=<tokenvalue>

`*Note:`*

Mandatory parameters are marked in redand placeholder content is marked in <green>.

get

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/cachepolicylabel_policybinding_binding/labelname_value<String>

Query-parameters:

filter

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/cachepolicylabel_policybinding_binding/labelname_value<String>?**filter=property-name1:property-value1,property-name2:property-value2**

Use this query-parameter to get the filtered set of cachepolicylabel_policybinding_binding resources configured on Citrix ADC. Filtering can be done on any of the properties of the resource.

pagination

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/cachepolicylabel_policybinding_binding/labelname_value<String>?**pagesize=#no;pageno=#no**

Use this query-parameter to get the cachepolicylabel_policybinding_binding resources in chunks.

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "cachepolicylabel_policybinding_binding": [ {

"priority":<Double_value>,

"policyname":<String_value>,

"labelname":<String_value>,

"invoke_labelname":<String_value>,

"gotopriorityexpression":<String_value>,

"invoke":<Boolean_value>,

"labeltype":<String_value>

}]}
```

get (all)

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/cachepolicylabel_policybinding_binding

Query-parameters:

bulkbindings

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/cachepolicylabel_policybinding_binding?**bulkbindings=yes**

NITRO allows you to fetch bindings in bulk.

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "cachepolicylabel_policybinding_binding": [ {

"priority":<Double_value>,

"policyname":<String_value>,

"labelname":<String_value>,

"invoke_labelname":<String_value>,

"gotopriorityexpression":<String_value>,

"invoke":<Boolean_value>,

"labeltype":<String_value>

}]}
```

count

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/cachepolicylabel_policybinding_binding/labelname_value<String>?**count=yes**

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{"cachepolicylabel_policybinding_binding": [ { "__count": "#no"} ] }
```

cachepolicy_binding

Binding object which returns the resources bound to cachepolicy.

Properties

(click to see Operations)

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
policyname	<String>	Read-write	Name of the cache policy about which to display details. Minimum length = 1
cachepolicy_cacheglobal_binding	<cachepolicy_cacheglobal_binding[]>	Read-only	cacheglobal that can be bound to cachepolicy.

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
cachepolicy_cachepolicylabel_binding	<cachepolicy_cachepolicylabel_binding[]>	Read-only	cachepolicylabel that can be bound to cachepolicy.
cachepolicy_csvserver_binding	<cachepolicy_csvserver_binding[]>	Read-only	csvserver that can be bound to cachepolicy.
cachepolicy_lbvserver_binding	<cachepolicy_lbvserver_binding[]>	Read-only	lbvserver that can be bound to cachepolicy.

Operations

(click to see Properties)

GET| GET (ALL)

Some options that you can use for each operations:

- **Getting warnings in response:**NITRO allows you to get warnings in an operation by specifying the "warning" query parameter as "yes". For example, to get warnings while connecting to the Citrix ADC appliance, the URL is as follows:

  http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/login?warning=yes

  If any, the warnings are displayed in the response payload with the HTTP code "209 X-NITRO-WARNING".

- **Authenticated access for individual NITRO operations:**NITRO allows you to logon to the Citrix ADC appliance to perform individual operations. You can use this option instead of creating a NITRO session (using the login object) and then using that session to perform all operations,

  To do this, you must specify the username and password in the request header of the NITRO request as follows:

  X-NITRO-USER:<username>

  X-NITRO-PASS:<password>

  Note: In such cases, make sure that the request header DOES not include the following:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue>

`*Note:` *

Mandatory parameters are marked in redand placeholder content is marked in <green>.

get

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/cachepolicy_binding/policyname_value<String>

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "cachepolicy_binding": [ {

"policyname":<String_value>,

"cachepolicy_lbvserver_binding":<cachepolicy_lbvserver_binding[]_value>,

"cachepolicy_cachepolicylabel_binding":<cachepolicy_cachepolicylabel_binding[]_value>,

"cachepolicy_cacheglobal_binding":<cachepolicy_cacheglobal_binding[]_value>,

"cachepolicy_csvserver_binding":<cachepolicy_csvserver_binding[]_value>

}]}
```

get (all)

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/cachepolicy_binding

Query-parameters:

bulkbindings

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/cachepolicy_binding?**bulkbindings=yes**

NITRO allows you to fetch bindings in bulk.

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "cachepolicy_binding": [ {

"policyname":<String_value>,

"cachepolicy_lbvserver_binding":<cachepolicy_lbvserver_binding[]_value>,

"cachepolicy_cachepolicylabel_binding":<cachepolicy_cachepolicylabel_bindi
ng[]_value>,

"cachepolicy_cacheglobal_binding":<cachepolicy_cacheglobal_binding[]_value
>,

"cachepolicy_csvserver_binding":<cachepolicy_csvserver_binding[]_value>
```

```
}]}
```

cachepolicy_cacheglobal_binding

Binding object showing the cacheglobal that can be bound to cachepolicy.

Properties

(click to see Operations)

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
policyname	<String>	Read-write	Name of the cache policy about which to display details. Minimum length = 1
boundto	<String>	Read-write	Location where policy is bound.
priority	<Double>	Read-only	Priority.
labelname	<String>	Read-only	Name of the label to invoke if the current policy rule evaluates to TRUE.
gotopriorityexpression	<String>	Read-only	Expression specifying the priority of the next policy which will get evaluated if the current policy rule evaluates to TRUE.
labeltype	<String>	Read-only	Type of policy label invocation. Possible values = reqvserver,

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
			resvserver, policylabel
activepolicy	<Integer>	Read-only	Indicates whether policy is bound or not.
__count	<Double>	Read-only	count parameter

Operations

(click to see Properties)

GET| GET (ALL)| COUNT

Some options that you can use for each operations:

- **Getting warnings in response:**NITRO allows you to get warnings in an operation by specifying the "warning" query parameter as "yes". For example, to get warnings while connecting to the Citrix ADC appliance, the URL is as follows:

  http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/login?warning=yes

  If any, the warnings are displayed in the response payload with the HTTP code "209 X-NITRO-WARNING".

- **Authenticated access for individual NITRO operations:**NITRO allows you to logon to the Citrix ADC appliance to perform individual operations. You can use this option instead of creating a NITRO session (using the login object) and then using that session to perform all operations,

  To do this, you must specify the username and password in the request header of the NITRO request as follows:

  X-NITRO-USER:<username>

  X-NITRO-PASS:<password>

  Note: In such cases, make sure that the request header DOES not include the following:

  Cookie:NITRO_AUTH_TOKEN=<tokenvalue>

`*Note:` *

Mandatory parameters are marked in redand placeholder content is marked in <green>.

get

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/cachepolicy_cacheglobal_binding/policyname_value<String>

Query-parameters:

filter

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/cachepolicy_cacheglobal_binding/policyname_value<String>?**filter=property-name1:property-value1,property-name2:property-value2**

Use this query-parameter to get the filtered set of cachepolicy_cacheglobal_binding resources configured on Citrix ADC. Filtering can be done on any of the properties of the resource.

pagination

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/cachepolicy_cacheglobal_binding/policyname_value<String>?**pagesize=#no;pageno=#no**

Use this query-parameter to get the cachepolicy_cacheglobal_binding resources in chunks.

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "cachepolicy_cacheglobal_binding": [ {
```

```
"policyname":<String_value>,

"boundto":<String_value>,

"priority":<Double_value>,

"labelname":<String_value>,

"gotopriorityexpression":<String_value>,

"labeltype":<String_value>,

"activepolicy":<Integer_value>

}]}
```

get (all)

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/cachepolicy_cacheglobal_binding

Query-parameters:

bulkbindings

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/cachepolicy_cacheglobal_binding?**bulkbindings=yes**

NITRO allows you to fetch bindings in bulk.

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "cachepolicy_cacheglobal_binding": [ {

"policyname":<String_value>,

"boundto":<String_value>,

"priority":<Double_value>,

"labelname":<String_value>,

"gotopriorityexpression":<String_value>,

"labeltype":<String_value>,

"activepolicy":<Integer_value>

}]}
```

count

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/cachepolicy_cacheglobal_binding/policyname_value<String>?**count=yes**

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{"cachepolicy_cacheglobal_binding": [ { "__count": "#no"} ] }
```

cachepolicy_cachepolicylabel_binding

Binding object showing the cachepolicylabel that can be bound to cachepolicy.

Properties

(click to see Operations)

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
policyname	<String>	Read-write	Name of the cache policy about which to display details. Minimum length = 1
boundto	<String>	Read-write	Location where policy is bound.
priority	<Double>	Read-only	Priority.
labelname	<String>	Read-only	Name of the label to invoke if the current policy rule evaluates to TRUE.
gotopriorityexpression	<String>	Read-only	Expression specifying the priority of the next policy which will get evaluated if the current policy rule evaluates to TRUE.
labeltype	<String>	Read-only	Type of policy label invocation. Possible values = reqvserver,

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
			resvserver, policylabel
activepolicy	<Integer>	Read-only	Indicates whether policy is bound or not.
__count	<Double>	Read-only	count parameter

Operations

(click to see Properties)

GET| GET (ALL)| COUNT

Some options that you can use for each operations:

- **Getting warnings in response:**NITRO allows you to get warnings in an operation by specifying the "warning" query parameter as "yes". For example, to get warnings while connecting to the Citrix ADC appliance, the URL is as follows:

  http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/login?warning=yes

  If any, the warnings are displayed in the response payload with the HTTP code "209 X-NITRO-WARNING".

- **Authenticated access for individual NITRO operations:**NITRO allows you to logon to the Citrix ADC appliance to perform individual operations. You can use this option instead of creating a NITRO session (using the login object) and then using that session to perform all operations,

  To do this, you must specify the username and password in the request header of the NITRO request as follows:

  X-NITRO-USER:<username>

  X-NITRO-PASS:<password>

  Note: In such cases, make sure that the request header DOES not include the following:

  Cookie:NITRO_AUTH_TOKEN=<tokenvalue>

`*Note:` *

Mandatory parameters are marked in redand placeholder content is marked in <green>.

get

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/cachepolicy_cachepolicylabel_binding/policyname_value<String>

Query-parameters:

filter

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/cachepolicy_cachepolicylabel_binding/policyname_value<String>?**filter=property-name1:property-value1,property-name2:property-value2**

Use this query-parameter to get the filtered set of cachepolicy_cachepolicylabel_binding resources configured on Citrix ADC. Filtering can be done on any of the properties of the resource.

pagination

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/cachepolicy_cachepolicylabel_binding/policyname_value<String>?**pagesize=#no;pageno=#no**

Use this query-parameter to get the cachepolicy_cachepolicylabel_binding resources in chunks.

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "cachepolicy_cachepolicylabel_binding": [ {
```

```
"policyname":<String_value>,

"boundto":<String_value>,

"priority":<Double_value>,

"labelname":<String_value>,

"gotopriorityexpression":<String_value>,

"labeltype":<String_value>,

"activepolicy":<Integer_value>

}]}
```

get (all)

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/cachepolicy_cachepolicylabel_binding

Query-parameters:

bulkbindings

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/cachepolicy_cachepolicylabel_binding?**bulkbindings=yes**

NITRO allows you to fetch bindings in bulk.

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "cachepolicy_cachepolicylabel_binding": [ {

"policyname":<String_value>,

"boundto":<String_value>,

"priority":<Double_value>,

"labelname":<String_value>,

"gotopriorityexpression":<String_value>,

"labeltype":<String_value>,

"activepolicy":<Integer_value>

}]}
```

count

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/cachepolicy_cachepolicylabel_binding/policyname_value<String>?**count=yes**

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{"cachepolicy_cachepolicylabel_binding": [ { "__count": "#no"} ] }
```

cachepolicy_csvserver_binding

Binding object showing the csvserver that can be bound to cachepolicy.

Properties

(click to see Operations)

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
policyname	<String>	Read-write	Name of the cache policy about which to display details. Minimum length = 1
boundto	<String>	Read-write	Location where policy is bound.
priority	<Double>	Read-only	Priority.
labelname	<String>	Read-only	Name of the label to invoke if the current policy rule evaluates to TRUE.
gotopriorityexpression	<String>	Read-only	Expression specifying the priority of the next policy which will get evaluated if the current policy rule evaluates to TRUE.
labeltype	<String>	Read-only	Type of policy label invocation. Possible values = reqvserver,

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
			resvserver, policylabel
activepolicy	<Integer>	Read-only	Indicates whether policy is bound or not.
__count	<Double>	Read-only	count parameter

Operations

(click to see Properties)

GET| GET (ALL)| COUNT

Some options that you can use for each operations:

- **Getting warnings in response:**NITRO allows you to get warnings in an operation by specifying the "warning" query parameter as "yes". For example, to get warnings while connecting to the Citrix ADC appliance, the URL is as follows:

  http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/login?warning=yes

  If any, the warnings are displayed in the response payload with the HTTP code "209 X-NITRO-WARNING".

- **Authenticated access for individual NITRO operations:**NITRO allows you to logon to the Citrix ADC appliance to perform individual operations. You can use this option instead of creating a NITRO session (using the login object) and then using that session to perform all operations,

  To do this, you must specify the username and password in the request header of the NITRO request as follows:

  X-NITRO-USER:<username>

  X-NITRO-PASS:<password>

  Note: In such cases, make sure that the request header DOES not include the following:

  Cookie:NITRO_AUTH_TOKEN=<tokenvalue>

`*Note:`*

Mandatory parameters are marked in redand placeholder content is marked in <green>.

get

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/cachepolicy_csvserver_binding/policyname_value<String>

Query-parameters:

filter

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/cachepolicy_csvserver_binding/policyname_value<String>?**filter=property-name1:property-value1,property-name2:property-value2**

Use this query-parameter to get the filtered set of cachepolicy_csvserver_binding resources configured on Citrix ADC. Filtering can be done on any of the properties of the resource.

pagination

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/cachepolicy_csvserver_binding/policyname_value<String>?**pagesize=#no;pageno=#no**

Use this query-parameter to get the cachepolicy_csvserver_binding resources in chunks.

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "cachepolicy_csvserver_binding": [ {
```

```
"policyname":<String_value>,

"boundto":<String_value>,

"priority":<Double_value>,

"labelname":<String_value>,

"gotopriorityexpression":<String_value>,

"labeltype":<String_value>,

"activepolicy":<Integer_value>

}]}
```

get (all)

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/cachepolicy_csvserver_binding

Query-parameters:

bulkbindings

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/cachepolicy_csvserver_binding?**bulkbindings=yes**

NITRO allows you to fetch bindings in bulk.

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "cachepolicy_csvserver_binding": [ {

"policyname":<String_value>,

"boundto":<String_value>,

"priority":<Double_value>,

"labelname":<String_value>,

"gotopriorityexpression":<String_value>,

"labeltype":<String_value>,

"activepolicy":<Integer_value>

}]}
```

count

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/cachepolicy_csvserver_binding/policyname_value<String>?**count=yes**

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{"cachepolicy_csvserver_binding": [ { "__count": "#no"} ] }
```

cachepolicy_lbvserver_binding

Binding object showing the lbvserver that can be bound to cachepolicy.

Properties

(click to see Operations)

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
policyname	<String>	Read-write	Name of the cache policy about which to display details. Minimum length = 1
boundto	<String>	Read-write	Location where policy is bound.
priority	<Double>	Read-only	Priority.
labelname	<String>	Read-only	Name of the label to invoke if the current policy rule evaluates to TRUE.
gotopriorityexpression	<String>	Read-only	Expression specifying the priority of the next policy which will get evaluated if the current policy rule evaluates to TRUE.
labeltype	<String>	Read-only	Type of policy label invocation. Possible values = reqvserver,

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
			resvserver, policylabel
activepolicy	<Integer>	Read-only	Indicates whether policy is bound or not.
__count	<Double>	Read-only	count parameter

Operations

(click to see Properties)

GET| GET (ALL)| COUNT

Some options that you can use for each operations:

- **Getting warnings in response:**NITRO allows you to get warnings in an operation by specifying the "warning" query parameter as "yes". For example, to get warnings while connecting to the Citrix ADC appliance, the URL is as follows:

  http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/login?warning=yes

  If any, the warnings are displayed in the response payload with the HTTP code "209 X-NITRO-WARNING".

- **Authenticated access for individual NITRO operations:**NITRO allows you to logon to the Citrix ADC appliance to perform individual operations. You can use this option instead of creating a NITRO session (using the login object) and then using that session to perform all operations,

  To do this, you must specify the username and password in the request header of the NITRO request as follows:

  X-NITRO-USER:<username>

  X-NITRO-PASS:<password>

  Note: In such cases, make sure that the request header DOES not include the following:

  Cookie:NITRO_AUTH_TOKEN=<tokenvalue>

`*Note:` *

Mandatory parameters are marked in redand placeholder content is marked in <green>.

get

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/cachepolicy_lbvserver_binding/policyname_value<String>

Query-parameters:

filter

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/cachepolicy_lbvserver_binding/policyname_value<String>?**filter=property-name1:property-value1,property-name2:property-value2**

Use this query-parameter to get the filtered set of cachepolicy_lbvserver_binding resources configured on Citrix ADC. Filtering can be done on any of the properties of the resource.

pagination

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/cachepolicy_lbvserver_binding/policyname_value<String>?**pagesize=#no;pageno=#no**

Use this query-parameter to get the cachepolicy_lbvserver_binding resources in chunks.

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "cachepolicy_lbvserver_binding": [ {
```

```
"policyname":<String_value>,

"boundto":<String_value>,

"priority":<Double_value>,

"labelname":<String_value>,

"gotopriorityexpression":<String_value>,

"labeltype":<String_value>,

"activepolicy":<Integer_value>

}]}
```

get (all)

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/cachepolicy_lbvserver_binding

Query-parameters:

bulkbindings

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/cachepolicy_lbvserver_binding?**bulkbindings=yes**

NITRO allows you to fetch bindings in bulk.

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "cachepolicy_lbvserver_binding": [ {

"policyname":<String_value>,

"boundto":<String_value>,

"priority":<Double_value>,

"labelname":<String_value>,

"gotopriorityexpression":<String_value>,

"labeltype":<String_value>,

"activepolicy":<Integer_value>

}]}
```

count

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/cachepolicy_lbvserver_binding/policyname_value<String>?**count=yes**

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{"cachepolicy_lbvserver_binding": [ { "__count": "#no"} ] }
```

cacheselector

Configuration for cache selector resource.

Properties

(click to see Operations)

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
selectorname	<String>	Read-write	Name for the selector. Must begin with an ASCII alphabetic or underscore (_) character, and must contain only ASCII alphanumeric, underscore, hash (#), period (.), space, colon (:), at (@), equals (=), and hyphen (-) characters.
rule	<String[]>	Read-write	One or multiple PIXL expressions for evaluating an HTTP request or response. Minimum length = 1
flags	<Double>	Read-only	Flags.
__count	<Double>	Read-only	count parameter

Operations

(click to see Properties)

ADD| DELETE| UPDATE| GET (ALL)| GET| COUNT

Some options that you can use for each operations:

- **Getting warnings in response:**NITRO allows you to get warnings in an operation by specifying the "warning" query parameter as "yes". For example, to get warnings while connecting to the Citrix ADC appliance, the URL is as follows:

  http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/login?warning=yes

  If any, the warnings are displayed in the response payload with the HTTP code "209 X-NITRO-WARNING".

- **Authenticated access for individual NITRO operations:**NITRO allows you to logon to the Citrix ADC appliance to perform individual operations. You can use this option instead of creating a NITRO session (using the login object) and then using that session to perform all operations,

  To do this, you must specify the username and password in the request header of the NITRO request as follows:

  X-NITRO-USER:<username>

  X-NITRO-PASS:<password>

  Note: In such cases, make sure that the request header DOES not include the following:

  Cookie:NITRO_AUTH_TOKEN=<tokenvalue>

`*Note:`*

Mandatory parameters are marked in redand placeholder content is marked in <green>.

add

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/cacheselector

HTTP Method:POST

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Content-Type:application/json

Request Payload:

```
{"cacheselector":{

<b>"selectorname":<String_value>,

</b><b>"rule":<String[]_value>

</b>}}
```

Response:

HTTP Status Code on Success: 201 Created HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error

delete

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/cacheselector/selectorname_value<String>

HTTP Method:DELETE

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue>

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error

update

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/cacheselector

HTTP Method:PUT

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Content-Type:application/json

Request Payload:

```
{"cacheselector":{

<b>"selectorname":<String_value>,

</b><b>"rule":<String[]_value>

</b>}}
```

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error

get (all)

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/cacheselector

Query-parameters:

attrs

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/cacheselector?**attrs=property-name1,property-name2**

Use this query parameter to specify the resource details that you want to retrieve.

filter

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/cacheselector?**filter=property-name1:property-val1,property-name2:property-val2**

Use this query-parameter to get the filtered set of cacheselector resources configured on Citrix ADC. Filtering can be done on any of the properties of the resource.

view

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/cacheselector?**view=summary**

Note: By default, the retrieved results are displayed in detail view (?view=detail).

pagination

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/cacheselector?**pagesize=#no;pageno=#no**

Use this query-parameter to get the cacheselector resources in chunks.

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "cacheselector": [ {

"selectorname":<String_value>,

"flags":<Double_value>,

"rule":<String[]_value>

}]}
```

get

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/cacheselector/selectorname_value<String>

Query-parameters:

attrs

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/cacheselector/selectorname_value<String>?**attrs=property-name1,property-name2**

Use this query parameter to specify the resource details that you want to retrieve.

view

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/cacheselector/selectorname_value<String>?**view=summary**

Note: By default, the retrieved results are displayed in detail view (?view=detail).

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "cacheselector": [ {

"selectorname":<String_value>,

"flags":<Double_value>,

"rule":<String[]_value>

}]}
```

count

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/cacheselector?**count=yes**

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "cacheselector": [ { "__count": "#no"} ] }
```

IPsec

Ipsec

IPSEC configuration.

NAME	DESCRIPTION
ipsecparameter	Configuration for IPSEC paramter
ipsecprofile	Configuration for IPSEC profile

ipsecparameter

Configuration for IPSEC paramter resource.

Properties

(click to see Operations)

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
ikeversion	<String>	Read-write	IKE Protocol Version. Default value: V2 Possible values = V1, V2
encalgo	<String[]>	Read-write	Type of encryption algorithm (Note: Selection of AES enables AES128). Default value: AES Possible values = AES, 3DES, AES192, AES256
hashalgo	<String[]>	Read-write	Type of hashing algorithm. Default value: HMAC_SHA256 Possible values = HMAC_SHA1, HMAC_SHA256, HMAC_SHA384, HMAC_SHA512, HMAC_MD5
lifetime	<Double>	Read-write	Lifetime of IKE SA in seconds. Lifetime of IPSec SA will be (lifetime of IKE SA/ 8). Minimum value = 480 Maximum value = 31536000

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
livenesscheckinterval	<Double>	Read-write	Number of seconds after which a notify payload is sent to check the liveliness of the peer. Additional retries are done as per retransmit interval setting. Zero value disables liveliness checks. Minimum value = 0 Maximum value = 64999
replaywindowsize	<Double>	Read-write	IPSec Replay window size for the data traffic. Minimum value = 0 Maximum value = 16384
ikeretryinterval	<Double>	Read-write	IKE retry interval for bringing up the connection. Minimum value = 60 Maximum value = 3600
perfectforwardsecrecy	<String>	Read-write	Enable/Disable PFS. Default value: DISABLE Possible values = ENABLE, DISABLE

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
retransmissiontime	<Double>	Read-write	The interval in seconds to retry sending the IKE messages to peer, three consecutive attempts are done with doubled interval after every failure, increases for every retransmit till 6 retransmits. Minimum value = 1 Maximum value = 99
responderonly	<String>	Read-only	Responder Only config for IKED. Default value: NO Possible values = YES, NO

Operations

(click to see Properties)

UPDATE| UNSET| GET (ALL)

Some options that you can use for each operations:

- **Getting warnings in response:**NITRO allows you to get warnings in an operation by specifying the "warning" query parameter as "yes". For example, to get warnings while connecting to the Citrix ADC appliance, the URL is as follows:

  http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/login?warning=yes

  If any, the warnings are displayed in the response payload with the HTTP code "209 X-NITRO-WARNING".

- **Authenticated access for individual NITRO operations:**NITRO allows you to logon to the Citrix ADC appliance to perform individual operations. You can use this option instead of creating a NITRO session (using the login object) and then using that session to perform all operations,

  To do this, you must specify the username and password in the request header of the NITRO request as follows:

  X-NITRO-USER:<username>

  X-NITRO-PASS:<password>

  Note: In such cases, make sure that the request header DOES not include the following:

  Cookie:NITRO_AUTH_TOKEN=<tokenvalue>

`*Note:`*

Mandatory parameters are marked in redand placeholder content is marked in <green>.

update

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/ipsecparameter

HTTP Method:PUT

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Content-Type:application/json

Request Payload:

```
{"ipsecparameter":{

"ikeversion":<String_value>,

"encalgo":<String[]_value>,

"hashalgo":<String[]_value>,

"lifetime":<Double_value>,

"livenesscheckinterval":<Double_value>,
```

```
"replaywindowsize":<Double_value>,

"ikeretryinterval":<Double_value>,

"perfectforwardsecrecy":<String_value>,

"retransmissiontime":<Double_value>

}}
```

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error

unset

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/ipsecparameter?**action=unset**

HTTP Method:POST

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Content-Type:application/json

Request Payload:

```
{"ipsecparameter":{

"ikeversion":true,

"encalgo":true,

"hashalgo":true,

"lifetime":true,

"livenesscheckinterval":true,
```

```
"replaywindowsize":true,

"ikeretryinterval":true,

"perfectforwardsecrecy":true,

"retransmissiontime":true

}}
```

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error

get (all)

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/ipsecparameter

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "ipsecparameter": [ {

"ikeversion":<String_value>,
```

```
"encalgo":<String[]_value>,

"hashalgo":<String[]_value>,

"lifetime":<Double_value>,

"livenesscheckinterval":<Double_value>,

"replaywindowsize":<Double_value>,

"ikeretryinterval":<Double_value>,

"perfectforwardsecrecy":<String_value>,

"responderonly":<String_value>,

"retransmissiontime":<Double_value>

}]}
```

ipsecprofile

Configuration for IPSEC profile resource.

Properties

(click to see Operations)

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
name	<String>	Read-write	The name of the ipsec profile. Minimum length = 1 Maximum length = 32
ikeversion	<String>	Read-write	IKE Protocol Version. Possible values = V1, V2

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
encalgo	<String[]>	Read-write	Type of encryption algorithm (Note: Selection of AES enables AES128). Possible values = AES, 3DES, AES192, AES256
hashalgo	<String[]>	Read-write	Type of hashing algorithm. Possible values = HMAC_SHA1, HMAC_SHA256, HMAC_SHA384, HMAC_SHA512, HMAC_MD5
lifetime	<Double>	Read-write	Lifetime of IKE SA in seconds. Lifetime of IPSec SA will be (lifetime of IKE SA/ 8). Minimum value = 480 Maximum value = 31536000
psk	<String>	Read-write	Pre shared key value.
publickey	<String>	Read-write	Public key file path.
privatekey	<String>	Read-write	Private key file path.
peerpublickey	<String>	Read-write	Peer public key file path.

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
livenesscheckinterval	<Double>	Read-write	Number of seconds after which a notify payload is sent to check the liveliness of the peer. Additional retries are done as per retransmit interval setting. Zero value disables liveliness checks. Minimum value = 0 Maximum value = 64999
replaywindowsize	<Double>	Read-write	IPSec Replay window size for the data traffic. Minimum value = 0 Maximum value = 16384
ikeretryinterval	<Double>	Read-write	IKE retry interval for bringing up the connection. Minimum value = 60 Maximum value = 3600
retransmissiontime	<Double>	Read-write	The interval in seconds to retry sending the IKE messages to peer, three consecutive attempts are done with doubled interval after every failure.

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
			Minimum value = 1 Maximum value = 99
perfectforwardsecrecy	<String>	Read-write	Enable/Disable PFS. Possible values = ENABLE, DISABLE
responderonly	<String>	Read-only	Responder Only config for IKED. Possible values = YES, NO
__count	<Double>	Read-only	count parameter

Operations

(click to see Properties)

ADD| DELETE| GET (ALL)| GET| COUNT

Some options that you can use for each operations:

- **Getting warnings in response:**NITRO allows you to get warnings in an operation by specifying the "warning" query parameter as "yes". For example, to get warnings while connecting to the Citrix ADC appliance, the URL is as follows:

  http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/login?warning=yes

  If any, the warnings are displayed in the response payload with the HTTP code "209 X-NITRO-WARNING".

- **Authenticated access for individual NITRO operations:**NITRO allows you to logon to the Citrix ADC appliance to perform individual operations. You can use this option instead of creating a NITRO session (using the login object) and then using that session to perform all operations,

  To do this, you must specify the username and password in the request header of the NITRO request as follows:

  X-NITRO-USER:<username>

X-NITRO-PASS:<password>

Note: In such cases, make sure that the request header DOES not include the following:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue>

`*Note:`*

Mandatory parameters are marked in redand placeholder content is marked in <green>.

add

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/ipsecprofile

HTTP Method:POST

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Content-Type:application/json

Request Payload:

```
{"ipsecprofile":{

<b>"name":<String_value>,

</b>"ikeversion":<String_value>,

"encalgo":<String[]_value>,

"hashalgo":<String[]_value>,

"lifetime":<Double_value>,

"psk":<String_value>,

"publickey":<String_value>,

"privatekey":<String_value>,

"peerpublickey":<String_value>,

"livenesscheckinterval":<Double_value>,
```

```
"replaywindowsize":<Double_value>,

"ikeretryinterval":<Double_value>,

"retransmissiontime":<Double_value>,

"perfectforwardsecrecy":<String_value>

}}
```

Response:

HTTP Status Code on Success: 201 Created HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error

delete

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/ipsecprofile/name_value<String>

HTTP Method:DELETE

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue>

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error

get (all)

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/ipsecprofile

Query-parameters:

attrs

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/ipsecprofile?**attrs=property-name1,property-name2**

Use this query parameter to specify the resource details that you want to retrieve.

filter

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/ipsecprofile?**filter=property-name1:property-val1,property-name2:property-val2**

Use this query-parameter to get the filtered set of ipsecprofile resources configured on Citrix ADC. Filtering can be done on any of the properties of the resource.

view

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/ipsecprofile?**view=summary**

Note: By default, the retrieved results are displayed in detail view (?view=detail).

pagination

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/ipsecprofile?**pagesize=#no;pageno=#no**

Use this query-parameter to get the ipsecprofile resources in chunks.

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "ipsecprofile": [ {

"name":<String_value>,

"ikeversion":<String_value>,

"encalgo":<String[]_value>,
```

```
"hashalgo":<String[]_value>,

"lifetime":<Double_value>,

"livenesscheckinterval":<Double_value>,

"replaywindowsize":<Double_value>,

"retransmissiontime":<Double_value>,

"psk":<String_value>,

"publickey":<String_value>,

"privatekey":<String_value>,

"peerpublickey":<String_value>,

"ikeretryinterval":<Double_value>,

"perfectforwardsecrecy":<String_value>,

"responderonly":<String_value>

}]}
```

get

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/ipsecprofile/name_value<String>

Query-parameters:

attrs

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/ipsecprofile/name_value<String>?**attrs=property-name1,property-name2**

Use this query parameter to specify the resource details that you want to retrieve.

view

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/ipsecprofile/name_value<String>?**view=summary**

Note: By default, the retrieved results are displayed in detail view (?view=detail).

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "ipsecprofile": [ {

"name":<String_value>,

"ikeversion":<String_value>,

"encalgo":<String[]_value>,

"hashalgo":<String[]_value>,

"lifetime":<Double_value>,

"livenesscheckinterval":<Double_value>,

"replaywindowsize":<Double_value>,

"retransmissiontime":<Double_value>,

"psk":<String_value>,

"publickey":<String_value>,
```

```
"privatekey":<String_value>,

"peerpublickey":<String_value>,

"ikeretryinterval":<Double_value>,

"perfectforwardsecrecy":<String_value>,

"responderonly":<String_value>

}]}
```

count

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/ipsecprofile?**count=yes**

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "ipsecprofile": [ { "__count": "#no"} ] }
```

IPsecalg

Ipsecalg

IPSECALG configuration.

NAME	DESCRIPTION
ipsecalgprofile	Configuration for IPSEC ALG profile
ipsecalgsession	Configuration for IPSEC ALG session

ipsecalgprofile

Configuration for IPSEC ALG profile resource.

Properties

(click to see Operations)

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
name	<String>	Read-write	The name of the ipsec alg profile. Minimum length = 1 Maximum length = 32
ikesessiontimeout	<Double>	Read-write	IKE session timeout in minutes. Default value: 60 Minimum value = 1 Maximum value = 1440
espsessiontimeout	<Double>	Read-write	ESP session timeout in minutes. Default value: 60 Minimum value = 1

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
			Maximum value = 1440
espgatetimeout	<Double>	Read-write	Timeout ESP in seconds as no ESP packets are seen after IKE negotiation. Default value: 30 Minimum value = 3 Maximum value = 1200
connfailover	<String>	Read-write	Mode in which the connection failover feature must operate for the IPSec Alg. After a failover, established UDP connections and ESP packet flows are kept active and resumed on the secondary appliance. Recomended setting is ENABLED. Default value: ENABLED Possible values = ENABLED, DISABLED
__count	<Double>	Read-only	count parameter

Operations

(click to see Properties)

ADD| UPDATE| UNSET| DELETE| GET (ALL)| GET| COUNT

Some options that you can use for each operations:

- **Getting warnings in response:**NITRO allows you to get warnings in an operation by specifying the "warning" query parameter as "yes". For example, to get warnings while connecting to the Citrix ADC appliance, the URL is as follows:

  http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/login?warning=yes

  If any, the warnings are displayed in the response payload with the HTTP code "209 X-NITRO-WARNING".

- **Authenticated access for individual NITRO operations:**NITRO allows you to logon to the Citrix ADC appliance to perform individual operations. You can use this option instead of creating a NITRO session (using the login object) and then using that session to perform all operations,

  To do this, you must specify the username and password in the request header of the NITRO request as follows:

  X-NITRO-USER:<username>

  X-NITRO-PASS:<password>

  Note: In such cases, make sure that the request header DOES not include the following:

  Cookie:NITRO_AUTH_TOKEN=<tokenvalue>

`*Note:`*

Mandatory parameters are marked in redand placeholder content is marked in <green>.

add

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/ipsecalgprofile

HTTP Method:POST

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Content-Type:application/json

Request Payload:

```
{"ipsecalgprofile":{

<b>"name":<String_value>,

</b>"ikesessiontimeout":<Double_value>,

"espsessiontimeout":<Double_value>,

"espgatetimeout":<Double_value>,

"connfailover":<String_value>

}}
```

Response:

HTTP Status Code on Success: 201 Created HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error

update

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/ipsecalgprofile

HTTP Method:PUT

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Content-Type:application/json

Request Payload:

```
{"ipsecalgprofile":{

<b>"name":<String_value>,

</b>"ikesessiontimeout":<Double_value>,
```

```
"espsessiontimeout":<Double_value>,

"espgatetimeout":<Double_value>,

"connfailover":<String_value>

}}
```

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error

unset

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/ipsecalgprofile?**action=unset**

HTTP Method:POST

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Content-Type:application/json

Request Payload:

```
{"ipsecalgprofile":{

<b>"name":<String_value>,

</b>"ikesessiontimeout":true,

"espsessiontimeout":true,

"espgatetimeout":true,

"connfailover":true

}}
```

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error

delete

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/ipsecalgprofile/name_value<String>

HTTP Method:DELETE

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue>

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error

get (all)

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/ipsecalgprofile

Query-parameters:

attrs

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/ipsecalgprofile?**attrs=property-name1,property-name2**

Use this query parameter to specify the resource details that you want to retrieve.

filter

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/ipsecalgprofile?**filter=property-name1:property-val1,property-name2:property-val2**

Use this query-parameter to get the filtered set of ipsecalgprofile resources configured on Citrix ADC. Filtering can be done on any of the properties of the resource.

view

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/ipsecalgprofile?**view=summary**

Note: By default, the retrieved results are displayed in detail view (?view=detail).

pagination

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/ipsecalgprofile?**pagesize=#no;pageno=#no**

Use this query-parameter to get the ipsecalgprofile resources in chunks.

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "ipsecalgprofile": [ {

"name":<String_value>,

"ikesessiontimeout":<Double_value>,

"espsessiontimeout":<Double_value>,

"espgatetimeout":<Double_value>,

"connfailover":<String_value>

}]}
```

get

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/ipsecalgprofile/name_value<String>

Query-parameters:

attrs

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/ipsecalgprofile/name_value<String>?**attrs=property-name1,property-name2**

Use this query parameter to specify the resource details that you want to retrieve.

view

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/ipsecalgprofile/name_value<String>?**view=summary**

Note: By default, the retrieved results are displayed in detail view (?view=detail).

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "ipsecalgprofile": [ {

"name":<String_value>,

"ikesessiontimeout":<Double_value>,

"espsessiontimeout":<Double_value>,
```

```
"espgatetimeout":<Double_value>,

"connfailover":<String_value>

}]}
```

count

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/ipsecalgprofile?**count=yes**

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "ipsecalgprofile": [ { "__count": "#no"} ] }
```

ipsecalgsession

Configuration for IPSEC ALG session resource.

Properties

(click to see Operations)

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
sourceip_alg	<String>	Read-write	Original Source IP address. Minimum length = 1
natip_alg	<String>	Read-write	Natted Source IP address. Minimum length = 1
destip_alg	<String>	Read-write	Destination IP address. Minimum length = 1
sourceip	<String>	Read-write	Original Source IP address. Minimum length = 1
natip	<String>	Read-write	Natted Source IP address. Minimum length = 1
destip	<String>	Read-write	Destination IP address. Minimum length = 1
spiin	<Double>	Read-only	Client SPI for ESP.
spiout	<Double>	Read-only	Server SPI for ESP.
__count	<Double>	Read-only	count parameter

Operations

(click to see Properties)

FLUSH| GET (ALL)| COUNT

Some options that you can use for each operations:

- **Getting warnings in response:**NITRO allows you to get warnings in an operation by specifying the "warning" query parameter as "yes". For example, to get warnings while connecting to the Citrix ADC appliance, the URL is as follows:

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/login?warning=yes

If any, the warnings are displayed in the response payload with the HTTP code "209 X-NITRO-WARNING".

- **Authenticated access for individual NITRO operations:**NITRO allows you to logon to the Citrix ADC appliance to perform individual operations. You can use this option instead of creating a NITRO session (using the login object) and then using that session to perform all operations,

  To do this, you must specify the username and password in the request header of the NITRO request as follows:

  X-NITRO-USER:<username>

  X-NITRO-PASS:<password>

  Note: In such cases, make sure that the request header DOES not include the following:

  Cookie:NITRO_AUTH_TOKEN=<tokenvalue>

`*Note:`*

Mandatory parameters are marked in redand placeholder content is marked in <green>.

flush

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/ipsecalgsession?**action=flush**

HTTP Method:POST

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Content-Type:application/json

Request Payload:

```
{"ipsecalgsession":{

"sourceip":<String_value>,

"natip":<String_value>,

"destip":<String_value>
```

```
}}
```

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error

get (all)

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/ipsecalgsession

Query-parameters:

args

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/ipsecalgsession?**args=sourceip_alg:<String_value>,natip_alg:<String_value>,destip_alg:<String_value>**

Use this query-parameter to get ipsecalgsession resources based on additional properties.

attrs

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/ipsecalgsession?**attrs=property-name1,property-name2**

Use this query parameter to specify the resource details that you want to retrieve.

filter

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/ipsecalgsession?**filter=property-name1:property-val1,property-name2:property-val2**

Use this query-parameter to get the filtered set of ipsecalgsession resources configured on Citrix ADC. Filtering can be done on any of the properties of the resource.

view

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/ipsecalgsession?**view=summary**

Note: By default, the retrieved results are displayed in detail view (?view=detail).

pagination

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/ipsecalgsession?**pagesize=#no;pageno=#no**

Use this query-parameter to get the ipsecalgsession resources in chunks.

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "ipsecalgsession": [ {

sourceip_alg:<String_value>,natip_alg:<String_value>,destip_alg:<String_va
lue>"sourceip":<String_value>,

"natip":<String_value>,

"destip":<String_value>,

"spiin":<Double_value>,

"spiout":<Double_value>

}]}
```

count

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/ipsecalgsession?**count=yes**

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "ipsecalgsession": [ { "__count": "#no"} ] }
```

LLDP

LLDP

Link Layer Discovery Protocol.

NAME	DESCRIPTION
lldpneighbors	Configuration for lldp neighbors
lldpparam	Configuration for lldp params

lldpneighbors

Configuration for lldp neighbors resource.

Properties

(click to see Operations)

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
ifnum	<String>	Read-write	Interface Name.

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
nodeid	<Double>	Read-write	Unique number that identifies the cluster node. Minimum value = 0 Maximum value = 31
chassisidsubtype	<String>	Read-only	Chassis id sub type. Possible values = NONE, Component, Interface Alias, Port Component, MAC Address, Network Address, Interface Name, Locally Assigned
chassisid	<String>	Read-only	Chassis Id.
portidsubtype	<String>	Read-only	Port id subtype. Possible values = NONE, Interface Alias, Port Component, MAC Address, Network Address, Interface Name, Agent Circuit ID, Locally Assigned
portid	<String>	Read-only	Port Id.
ttl	<Double>	Read-only	Time to Live.
portdescription	<String>	Read-only	Port Description.
sys	<String>	Read-only	System Name.
sysdesc	<String>	Read-only	System Description.

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
mgmtaddresssubtype	<String>	Read-only	Management Address Type. Possible values = OTHER, IPv4, IPv6
mgmtaddress	<String>	Read-only	Management Address.
iftype	<String>	Read-only	Interface subtype. Possible values = UNKNOWN, ifIndex, system port number
ifnumber	<Double>	Read-only	Interface Number.
vlan	<String>	Read-only	vlan name.
vlanid	<Double>	Read-only	Vlan Id.
portprotosupported	<Double>	Read-only	Flag to show Port Protocol Support.
portprotoenabled	<Double>	Read-only	Flag to show Port Protocol support enabled.
portprotoid	<Double>	Read-only	Port Protocol ID.
portvlanid	<Double>	Read-only	Port Vlan Id.
protocolid	<String>	Read-only	port vlan id name.
linkaggrcapable	<String>	Read-only	Is neighbor Link Aggregation Capable. Possible values = NO, YES

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
linkaggrenabled	<String>	Read-only	Is link aggregation Enabled. Possible values = NO, YES
linkaggrid	<Double>	Read-only	Link Aggregation Id.
flag	<Double>	Read-only	.
syscapabilities	<String>	Read-only	Acronyms for remote system capabilities: OT : Other. RE : Repeater(IETF RFC 2108). BR : MAC Bridge(IEEE Std 802.1D). WL : WLAN Access Point(IEEE Std 802.11 MIB). RO : ROuter(IETF RFC 1812). TE : Telephone(IETF RFC 4293). DO : DOCSIS cable device(IETF RFC 4639 and IETF RFC 4546). ST : STation only(IETF RFC 4293).

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
			CV : C-VLAN component of a VLAN Bridge(IEEE Std 802.1Q). SV : S-VLAN component of a VLAN Bridge(IEEE Std 802.1Q). MR : Two port MAC Relay(IEEE Std 802.1Q).
syscapenabled	<String>	Read-only	Acronyms for remote system enabled capabilities: OT : Other. RE : Repeater(IETF RFC 2108). BR : MAC Bridge(IEEE Std 802.1D). WL : WLAN Access Point(IEEE Std 802.11 MIB). RO : ROuter(IETF RFC 1812). TE : Telephone(IETF RFC 4293). DO : DOCSIS cable device(IETF RFC

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
			4639 and IETF RFC 4546). ST : STation only(IETF RFC 4293). CV : C-VLAN component of a VLAN Bridge(IEEE Std 802.1Q). SV : S-VLAN component of a VLAN Bridge(IEEE Std 802.1Q). MR : Two port MAC Relay(IEEE Std 802.1Q).
autonegsupport	<String>	Read-only	MAC/PHY autonegotiation support. Possible values = NO, YES
autonegenabled	<String>	Read-only	MAC/PHY autonegotiation enabled. Possible values = NO, YES
autonegadvertised	<String>	Read-only	MAC/PHY autonegotiation advetised.
autonegmautype	<String>	Read-only	MAC/PHY Medium Attachment Unit (MAU) type of the

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
			port. Description listed below: AUI - no internal MAU, view from AUI 10Base5 - thick coax MAU Foirl - FOIRL MAU 10Base2 - thin coax MAU 10BaseT - UTP MAU 10BaseFP - passive fiber MAU 10BaseFB - sync fiber MAU 10BaseFL - async fiber MAU 10Broad36 - broadband DTE MAU 10BaseTHD - UTP MAU, half duplex mode 10BaseTFD - UTP MAU, full duplex mode 10BaseFLHD - async fiber MAU, half duplex mode

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
			10BaseFLDF - async fiber MAU, full duplex mode 10BaseT4 - 4 pair category 3 UTP 100BaseTXHD - 2 pair category 5 UTP, half duplex mode 100BaseTXFD - 2 pair category 5 UTP, full duplex mode 100BaseFXHD - X fiber over PMT, half duplex mode 100BaseFXFD - X fiber over PMT, full duplex mode 100BaseT2HD - 2 pair category 3 UTP, half duplex mode 100BaseT2DF - 2 pair category 3 UTP, full duplex mode 1000BaseXHD - PCS/PMA,

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
			unknown PMD, half duplex mode 1000BaseXFD - PCS/PMA, unknown PMD, full duplex mode 1000BaseLXHD - Fiber over long-wavelength laser, half duplex mode 1000BaseLXFD - Fiber over long-wavelength laser, full duplex mode 1000BaseSXHD - Fiber over short-wavelength laser, half duplex mode 1000BaseSXFD - Fiber over short-wavelength laser, full duplex mode 1000BaseCXHD - Copper over 150-Ohm balanced cable, half duplex mode 1000BaseCXFD - Copper over 150-Ohm balanced

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
			cable, full duplex mode 1000BaseTHD - Four-pair Category 5 UTP, half duplex mode 1000BaseTFD - Four-pair Category 5 UTP, full duplex mode 10GigBaseX - X PCS/PMA, unknown PMD 10GigBaseLX4 - X fiber over WWDM optics 10GigBaseR - R PCS/PMA, unknown PMD 10GigBaseER - R fiber over 1550 nm optics 10GigBaseLR - R fiber over 1310 nm optics 10GigBaseSR - R fiber over 850 nm optics

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
			10GigBaseW - W PCS/PMA, unknown PMD 10GigBaseEW - W fiber over 1550 nm optics 10GigBaseLW - W fiber over 1310 nm optics 10GigBaseSW - W fiber over 850 nm optics. Possible values = Not Recieved, AUI, 10Base5, Foirl, 10Base2, 10BaseT, 10BaseFP, 10BaseFB, 10BaseFL, 10Broad36, 10BaseTHD, 10BaseTFD, 10BaseFLHD, 10BaseFLDF, 10BaseT4, 100BaseTXHD, 100BaseTXFD, 100BaseFXHD, 100BaseFXFD, 100BaseT2HD, 100BaseT2DF, 1000BaseXHD, 1000BaseXFD,

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
			1000BaseLXHD, 1000BaseLXFD, 1000BaseSXHD, 1000BaseSXFD, 1000BaseCXHD, 1000BaseCXFD, 1000BaseTHD, 1000BaseTFD, 10GigBaseX, 10GigBaseLX4, 10GigBaseR, 10GigBaseER, 10GigBaseLR, 10GigBaseSR, 10GigBaseW, 10GigBaseEW, 10GigBaseLW, 10GigBaseSW
mtu	<Double>	Read-only	MTU of Remote Device.
__count	<Double>	Read-only	count parameter

Operations

(click to see Properties)

CLEAR| GET (ALL)| GET| COUNT

Some options that you can use for each operations:

- **Getting warnings in response:**NITRO allows you to get warnings in an operation by specifying the "warning" query parameter as "yes". For example, to get warnings while connecting to the Citrix ADC appliance, the URL is as follows:

  http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/login?warning=yes

If any, the warnings are displayed in the response payload with the HTTP code "209 X-NITRO-WARNING".

- **Authenticated access for individual NITRO operations:**NITRO allows you to logon to the Citrix ADC appliance to perform individual operations. You can use this option instead of creating a NITRO session (using the login object) and then using that session to perform all operations,

  To do this, you must specify the username and password in the request header of the NITRO request as follows:

  X-NITRO-USER:<username>

  X-NITRO-PASS:<password>

  Note: In such cases, make sure that the request header DOES not include the following:

  Cookie:NITRO_AUTH_TOKEN=<tokenvalue>

`*Note:`*

Mandatory parameters are marked in redand placeholder content is marked in <green>.

clear

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/lldpneighbors?**action=clear**

HTTP Method:POST

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Content-Type:application/json

Request Payload:

```
{"lldpneighbors":{

}}
```

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error

get (all)

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/lldpneighbors

Query-parameters:

args

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/lldpneighbors?**args=ifnum:<String_value>,nodeid:<Double_value>**

Use this query-parameter to get lldpneighbors resources based on additional properties.

attrs

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/lldpneighbors?**attrs=property-name1,property-name2**

Use this query parameter to specify the resource details that you want to retrieve.

filter

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/lldpneighbors?**filter=property-name1:property-val1,property-name2:property-val2**

Use this query-parameter to get the filtered set of lldpneighbors resources configured on Citrix ADC. Filtering can be done on any of the properties of the resource.

view

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/lldpneighbors?**view=summary**

Note: By default, the retrieved results are displayed in detail view (?view=detail).

pagination

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/lldpneighbors?**pagesize=#no;pageno=#no**

Use this query-parameter to get the lldpneighbors resources in chunks.

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "lldpneighbors": [ {

ifnum:<String_value>,nodeid:<Double_value>"chassisidsubtype":<String_value
>,

"chassisid":<String_value>,

"portidsubtype":<String_value>,

"portid":<String_value>,

"ttl":<Double_value>,

"portdescription":<String_value>,

"sys":<String_value>,

"sysdesc":<String_value>,

"mgmtaddresssubtype":<String_value>,

"mgmtaddress":<String_value>,

"iftype":<String_value>,

"ifnumber":<Double_value>,

"vlan":<String_value>,
```

```
"vlanid":<Double_value>,

"portprotosupported":<Double_value>,

"portprotoenabled":<Double_value>,

"portprotoid":<Double_value>,

"portvlanid":<Double_value>,

"protocolid":<String_value>,

"linkaggrcapable":<String_value>,

"linkaggrenabled":<String_value>,

"linkaggrid":<Double_value>,

"flag":<Double_value>,

"syscapabilities":<String_value>,

"syscapenabled":<String_value>,

"autonegsupport":<String_value>,

"autonegenabled":<String_value>,

"autonegadvertised":<String_value>,

"autonegmautype":<String_value>,

"mtu":<Double_value>

}]}
```

get

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/lldpneighbors/ifnum_value<String>

Query-parameters:

attrs

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/lldpneighbors/ifnum_value<String>?**attrs=property-name1,property-name2**

Use this query parameter to specify the resource details that you want to retrieve.

view

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/lldpneighbors/ifnum_value<String>?**view=summary**

Note: By default, the retrieved results are displayed in detail view (?view=detail).

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "lldpneighbors": [ {

ifnum:<String_value>,nodeid:<Double_value>"chassisidsubtype":<String_value
>,

"chassisid":<String_value>,

"portidsubtype":<String_value>,

"portid":<String_value>,

"ttl":<Double_value>,
```

```
"portdescription":<String_value>,

"sys":<String_value>,

"sysdesc":<String_value>,

"mgmtaddresssubtype":<String_value>,

"mgmtaddress":<String_value>,

"iftype":<String_value>,

"ifnumber":<Double_value>,

"vlan":<String_value>,

"vlanid":<Double_value>,

"portprotosupported":<Double_value>,

"portprotoenabled":<Double_value>,

"portprotoid":<Double_value>,

"portvlanid":<Double_value>,

"protocolid":<String_value>,

"linkaggrcapable":<String_value>,

"linkaggrenabled":<String_value>,

"linkaggrid":<Double_value>,

"flag":<Double_value>,

"syscapabilities":<String_value>,

"syscapenabled":<String_value>,

"autonegsupport":<String_value>,

"autonegenabled":<String_value>,
```

```
"autonegadvertised":<String_value>,

"autonegmautype":<String_value>,

"mtu":<Double_value>

}]}
```

count

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/lldpneighbors?**count=yes**

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "lldpneighbors": [ { "__count": "#no"} ] }
```

lldpparam

Configuration for lldp params resource.

Properties

(click to see Operations)

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
holdtimetxmult	<Double>	Read-write	A multiplier for calculating the duration for which the receiving device stores the LLDP information in its database before discarding or removing it. The duration is calculated as the holdtimeTxMult (Holdtime Multiplier) parameter value multiplied by the timer (Timer) parameter value. Default value: 4 Minimum value = 1 Maximum value = 20
timer	<Double>	Read-write	Interval, in seconds, between LLDP packet data units (LLDPDUs). that the Citrix ADC sends to a directly connected device. Default value: 30 Minimum value = 1 Maximum value = 3000
mode	<String>	Read-write	Global mode of Link Layer Discovery Protocol (LLDP) on

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
			the Citrix ADC. The resultant LLDP mode of an interface depends on the LLDP mode configured at the global and the interface levels. Possible values = NONE, TRANSMITTER, RECEIVER, TRANSCEIVER

Operations

(click to see Properties)

UPDATE| UNSET| GET (ALL)

Some options that you can use for each operations:

- **Getting warnings in response:**NITRO allows you to get warnings in an operation by specifying the "warning" query parameter as "yes". For example, to get warnings while connecting to the Citrix ADC appliance, the URL is as follows:

  http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/login?warning=yes

  If any, the warnings are displayed in the response payload with the HTTP code "209 X-NITRO-WARNING".

- **Authenticated access for individual NITRO operations:**NITRO allows you to logon to the Citrix ADC appliance to perform individual operations. You can use this option instead of creating a NITRO session (using the login object) and then using that session to perform all operations,

  To do this, you must specify the username and password in the request header of the NITRO request as follows:

  X-NITRO-USER:<username>

X-NITRO-PASS:<password>

Note: In such cases, make sure that the request header DOES not include the following:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue>

`*Note:`*

Mandatory parameters are marked in redand placeholder content is marked in <green>.

update

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/lldpparam

HTTP Method:PUT

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Content-Type:application/json

Request Payload:

```
{"lldpparam":{

"holdtimetxmult":<Double_value>,

"timer":<Double_value>,

"mode":<String_value>

}}
```

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error

unset

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/lldpparam?**action=unset**

HTTP Method:POST

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Content-Type:application/json

Request Payload:

```
{"lldpparam":{

"holdtimetxmult":true,

"timer":true,

"mode":true

}}
```

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error

get (all)

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/lldpparam

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "lldpparam": [ {

"holdtimetxmult":<Double_value>,

"timer":<Double_value>,

"mode":<String_value>

}]}
```

Load Balancing

Load Balancing

Load Balancing configuration. The load balancing methods manage the selection of an appropriate physical server in a server farm.

NAME	DESCRIPTION
lbgroup	Configuration for LB group
lbgroup_binding	Binding object showing the resources that can be bound to lbgroup
lbgroup_lbvserver_binding	Binding object showing the lbvserver that can be bound to lbgroup
lbmetrictable	Configuration for metric table
lbmetrictable_binding	Binding object showing the resources that can be bound to lbmetrictable
lbmetrictable_metric_binding	Binding object showing the metric that can be bound to lbmetrictable
lbmonbindings	Configuration for monitro bindings
lbmonbindings_binding	Binding object showing the resources that can be bound to lbmonbindings

NAME	DESCRIPTION
lbmonbindings_gslbservicegroup_binding	Binding object showing the gslbservicegroup that can be bound to lbmonbindings
lbmonbindings_service_binding	Binding object showing the service that can be bound to lbmonbindings
lbmonbindings_servicegroup_binding	Binding object showing the servicegroup that can be bound to lbmonbindings
lbmonitor	Configuration for monitor
lbmonitor_binding	Binding object showing the resources that can be bound to lbmonitor
lbmonitor_metric_binding	Binding object showing the metric that can be bound to lbmonitor
lbmonitor_service_binding	Binding object showing the service that can be bound to lbmonitor
lbmonitor_servicegroup_binding	Binding object showing the servicegroup that can be bound to lbmonitor
lbmonitor_sslcertkey_binding	Binding object showing the sslcertkey that can be bound to lbmonitor
lbparameter	Configuration for LB parameter
lbpersistentsessions	Configuration for persistence session
lbprofile	Configuration for LB profile
lbroute	Configuration for LB route
lbroute6	Configuration for LB route6
lbsipparameters	Configuration for SIP parameters
lbvserver	Configuration for Load Balancing Virtual Server
lbvserver_analyticsprofile_binding	Binding object showing the analyticsprofile that can be bound to lbvserver

NAME	DESCRIPTION
lbvserver_appflowpolicy_binding	Binding object showing the appflowpolicy that can be bound to lbvserver
lbvserver_appfwpolicy_binding	Binding object showing the appfwpolicy that can be bound to lbvserver
lbvserver_appqoepolicy_binding	Binding object showing the appqoepolicy that can be bound to lbvserver
lbvserver_auditnslogpolicy_binding	Binding object showing the auditnslogpolicy that can be bound to lbvserver
lbvserver_auditsyslogpolicy_binding	Binding object showing the auditsyslogpolicy that can be bound to lbvserver
lbvserver_authorizationpolicy_binding	Binding object showing the authorizationpolicy that can be bound to lbvserver
lbvserver_binding	Binding object showing the resources that can be bound to lbvserver
lbvserver_cachepolicy_binding	Binding object showing the cachepolicy that can be bound to lbvserver
lbvserver_capolicy_binding	Binding object showing the capolicy that can be bound to lbvserver
lbvserver_cmppolicy_binding	Binding object showing the cmppolicy that can be bound to lbvserver
lbvserver_contentinspectionpolicy_binding	Binding object showing the contentinspectionpolicy that can be bound to lbvserver
lbvserver_csvserver_binding	Binding object showing the csvserver that can be bound to lbvserver
lbvserver_dnspolicy64_binding	Binding object showing the dnspolicy64 that can be bound to lbvserver
lbvserver_dospolicy_binding	Binding object showing the dospolicy that can be bound to lbvserver

NAME	DESCRIPTION
lbvserver_feopolicy_binding	Binding object showing the feopolicy that can be bound to lbvserver
lbvserver_filterpolicy_binding	Binding object showing the filterpolicy that can be bound to lbvserver
lbvserver_pqpolicy_binding	Binding object showing the pqpolicy that can be bound to lbvserver
lbvserver_responderpolicy_binding	Binding object showing the responderpolicy that can be bound to lbvserver
lbvserver_rewritepolicy_binding	Binding object showing the rewritepolicy that can be bound to lbvserver
lbvserver_scpolicy_binding	Binding object showing the scpolicy that can be bound to lbvserver
lbvserver_service_binding	Binding object showing the service that can be bound to lbvserver
lbvserver_servicegroup_binding	Binding object showing the servicegroup that can be bound to lbvserver
lbvserver_servicegroupmember_binding	Binding object showing the servicegroupmember that can be bound to lbvserver
lbvserver_spilloverpolicy_binding	Binding object showing the spilloverpolicy that can be bound to lbvserver
lbvserver_tmtrafficpolicy_binding	Binding object showing the tmtrafficpolicy that can be bound to lbvserver
lbvserver_transformpolicy_binding	Binding object showing the transformpolicy that can be bound to lbvserver
lbvserver_videooptimizationdetectionpolicy_binding	Binding object showing the videooptimizationdetectionpolicy that can be bound to lbvserver

NAME	DESCRIPTION
lbvserver_videooptimizationpacingpolicy_binding	Binding object showing the videooptimizationpacingpolicy that can be bound to lbvserver
lbwlm	Configuration for web log manager
lbwlm_binding	Binding object showing the resources that can be bound to lbwlm
lbwlm_lbvserver_binding	Binding object showing the lbvserver that can be bound to lbwlm

lbgroup

Configuration for LB group resource.

Properties

(click to see Operations)

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
name	<String>	Read-write	Name of the load balancing virtual server group. Minimum length = 1
persistencetype	<String>	Read-write	Type of persistence for the group. Available settings function as follows: * SOURCEIP - Create persistence sessions based on the client IP. * COOKIEINSERT - Create persistence sessions based on a cookie in client

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
			requests. The cookie is inserted by a Set-Cookie directive from the server, in its first response to a client. * RULE - Create persistence sessions based on a user defined rule. * NONE - Disable persistence for the group. Possible values = SOURCEIP, COOKIEINSERT, RULE, NONE
persistencebackup	<String>	Read-write	Type of backup persistence for the group. Possible values = SOURCEIP, NONE
backuppersistenceti meout	<Double>	Read-write	Time period, in minutes, for which backup persistence is in effect. Default value: 2 Minimum value = 2 Maximum value = 1440

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
persistmask	<String>	Read-write	Persistence mask to apply to source IPv4 addresses when creating source IP based persistence sessions. Minimum length = 1
cookiename	<String>	Read-write	Use this parameter to specify the cookie name for COOKIE peristence type. It specifies the name of cookie with a maximum of 32 characters. If not specified, cookie name is internally generated.
v6persistmasklen	<Double>	Read-write	Persistence mask to apply to source IPv6 addresses when creating source IP based persistence sessions. Default value: 128 Minimum value = 1 Maximum value = 128
cookiedomain	<String>	Read-write	Domain attribute for the HTTP cookie. Minimum length = 1
timeout	<Double>	Read-write	Time period for which a persistence session is in effect.

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
			Default value: 2 Minimum value = 0 Maximum value = 1440
rule	<String>	Read-write	Expression, or name of a named expression, against which traffic is evaluated. The following requirements apply only to the Citrix ADC CLI: * If the expression includes one or more spaces, enclose the entire expression in double quotation marks. * If the expression itself includes double quotation marks, escape the quotations by using the \ character. * Alternatively, you can use single quotation marks to enclose the rule, in which case you do not have to escape

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
			the double quotation marks. Default value: "None"
usevserverpersistency	<String>	Read-write	Use this parameter to enable vserver level persistence on group members. This allows member vservers to have their own persistence, but need to be compatible with other members persistence rules. When this setting is enabled persistence sessions created by any of the members can be shared by other member vservers. Default value: DISABLED Possible values = ENABLED, DISABLED
mastervserver	<String>	Read-write	When USE_VSERVER_PERSISTENCE is enabled, one can use this setting to designate a member vserver as master which is responsible to create the

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
			persistence sessions.
newname	<String>	Read-write	New name for the load balancing virtual server group. Minimum length = 1
td	<Double>	Read-only	Integer value that uniquely identifies the traffic domain in which you want to configure the entity. If you do not specify an ID, the entity becomes part of the default traffic domain, which has an ID of 0. Minimum value = 0 Maximum value = 4094
__count	<Double>	Read-only	count parameter

Operations

(click to see Properties)

ADD| UPDATE| UNSET| DELETE| RENAME| GET (ALL)| GET| COUNT

Some options that you can use for each operations:

- **Getting warnings in response:**NITRO allows you to get warnings in an operation by specifying the "warning" query parameter as "yes". For example, to get warnings while connecting to the Citrix ADC appliance, the URL is as follows:

  http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/login?warning=yes

  If any, the warnings are displayed in the response payload with the HTTP code "209 X-NITRO-WARNING".

- **Authenticated access for individual NITRO operations:**NITRO allows you to logon to the Citrix ADC appliance to perform individual operations. You can use this option instead of creating a NITRO session (using the login object) and then using that session to perform all operations,

  To do this, you must specify the username and password in the request header of the NITRO request as follows:

  X-NITRO-USER:<username>

  X-NITRO-PASS:<password>

  Note: In such cases, make sure that the request header DOES not include the following:

  Cookie:NITRO_AUTH_TOKEN=<tokenvalue>

`*Note:`*

Mandatory parameters are marked in redand placeholder content is marked in <green>.

add

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/lbgroup

HTTP Method:POST

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Content-Type:application/json

Request Payload:

```
{"lbgroup":{

<b>"name":<String_value>,

</b>"persistencetype":<String_value>,

"persistencebackup":<String_value>,

"backuppersistencetimeout":<Double_value>,

"persistmask":<String_value>,
```

```
"cookiename":<String_value>,

"v6persistmasklen":<Double_value>,

"cookiedomain":<String_value>,

"timeout":<Double_value>,

"rule":<String_value>,

"usevserverpersistency":<String_value>

}}
```

Response:

HTTP Status Code on Success: 201 Created HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error

update

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/lbgroup

HTTP Method:PUT

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Content-Type:application/json

Request Payload:

```
{"lbgroup":{

<b>"name":<String_value>,

</b>"persistencetype":<String_value>,

"persistencebackup":<String_value>,
```

```
"backuppersistencetimeout":<Double_value>,

"persistmask":<String_value>,

"cookiename":<String_value>,

"v6persistmasklen":<Double_value>,

"cookiedomain":<String_value>,

"timeout":<Double_value>,

"rule":<String_value>,

"usevserverpersistency":<String_value>,

"mastervserver":<String_value>

}}
```

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error

unset

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/lbgroup?**action=unset**

HTTP Method:POST

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Content-Type:application/json

Request Payload:

```
{"lbgroup":{
```

```
<b>"name":<String_value>,

</b>"persistencetype":true,

"persistencebackup":true,

"backuppersistencetimeout":true,

"persistmask":true,

"cookiename":true,

"v6persistmasklen":true,

"cookiedomain":true,

"timeout":true,

"rule":true,

"usevserverpersistency":true,

"mastervserver":true

}}
```

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error

delete

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/lbgroup/name_value<String>

HTTP Method:DELETE

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue>

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error

rename

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/lbgroup?**action=rename**

HTTP Method:POST

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Content-Type:application/json

Request Payload:

```
{"lbgroup":{

<b>"name":<String_value>,

</b><b>"newname":<String_value>

</b>}}
```

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error

get (all)

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/lbgroup

Query-parameters:

attrs

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/lbgroup?**attrs=property-name1,property-name2**

Use this query parameter to specify the resource details that you want to retrieve.

filter

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/lbgroup?**filter=property-name1:property-val1,property-name2:property-val2**

Use this query-parameter to get the filtered set of lbgroup resources configured on Citrix ADC. Filtering can be done on any of the properties of the resource.

view

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/lbgroup?**view=summary**

Note: By default, the retrieved results are displayed in detail view (?view=detail).

pagination

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/lbgroup?**pagesize=#no;pageno=#no**

Use this query-parameter to get the lbgroup resources in chunks.

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "lbgroup": [ {

"name":<String_value>,

"persistencetype":<String_value>,
```

```
"persistencebackup":<String_value>,

"backuppersistencetimeout":<Double_value>,

"persistmask":<String_value>,

"v6persistmasklen":<Double_value>,

"cookiename":<String_value>,

"cookiedomain":<String_value>,

"timeout":<Double_value>,

"rule":<String_value>,

"td":<Double_value>,

"usevserverpersistency":<String_value>,

"mastervserver":<String_value>

}]}
```

get

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/lbgroup/name_value<String>

Query-parameters:

attrs

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/lbgroup/name_value<String>?**attrs=property-name1,property-name2**

Use this query parameter to specify the resource details that you want to retrieve.

view

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/lbgroup/name_value<String>?**view=summary**

Note: By default, the retrieved results are displayed in detail view (?view=detail).

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "lbgroup": [ {

"name":<String_value>,

"persistencetype":<String_value>,

"persistencebackup":<String_value>,

"backuppersistencetimeout":<Double_value>,

"persistmask":<String_value>,

"v6persistmasklen":<Double_value>,

"cookiename":<String_value>,

"cookiedomain":<String_value>,

"timeout":<Double_value>,

"rule":<String_value>,

"td":<Double_value>,

"usevserverpersistency":<String_value>,
```

```
"mastervserver":<String_value>

}]}
```

count

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/lbgroup?**count=yes**

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "lbgroup": [ { "__count": "#no"} ] }
```

lbgroup_binding

Binding object which returns the resources bound to lbgroup.

Properties

(click to see Operations)

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
name	<String>	Read-write	Name of the load balancing virtual server group.

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
			Minimum length = 1
lbgroup_lbvserver_binding	<lbgroup_lbvserver_binding[]>	Read-only	lbvserver that can be bound to lbgroup.

Operations

(click to see Properties)

GET| GET (ALL)

Some options that you can use for each operations:

- **Getting warnings in response:**NITRO allows you to get warnings in an operation by specifying the "warning" query parameter as "yes". For example, to get warnings while connecting to the Citrix ADC appliance, the URL is as follows:

  http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/login?warning=yes

  If any, the warnings are displayed in the response payload with the HTTP code "209 X-NITRO-WARNING".

- **Authenticated access for individual NITRO operations:**NITRO allows you to logon to the Citrix ADC appliance to perform individual operations. You can use this option instead of creating a NITRO session (using the login object) and then using that session to perform all operations,

  To do this, you must specify the username and password in the request header of the NITRO request as follows:

  X-NITRO-USER:<username>

  X-NITRO-PASS:<password>

  Note: In such cases, make sure that the request header DOES not include the following:

  Cookie:NITRO_AUTH_TOKEN=<tokenvalue>

`*Note:`*

Mandatory parameters are marked in redand placeholder content is marked in <green>.

get

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/lbgroup_binding/name_value<String>

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "lbgroup_binding": [ {

"name":<String_value>,

"lbgroup_lbvserver_binding":<lbgroup_lbvserver_binding[]_value>

}]}
```

get (all)

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/lbgroup_binding

Query-parameters:

bulkbindings

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/lbgroup_binding?**bulkbindings=yes**

NITRO allows you to fetch bindings in bulk.

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "lbgroup_binding": [ {

"name":<String_value>,

"lbgroup_lbvserver_binding":<lbgroup_lbvserver_binding[]_value>

}]}
```

lbgroup_lbvserver_binding

Binding object showing the lbvserver that can be bound to lbgroup.

Properties

(click to see Operations)

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
name	<String>	Read-write	Name for the load balancing virtual server group. Must begin with an ASCII alphanumeric or underscore (_) character, and must contain only ASCII alphanumeric, underscore, hash

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
			(#), period (.), space, colon (:), at (@), equals (=), and hyphen (-) characters. Can be changed after the virtual server is created. CLI Users: If the name includes one or more spaces, enclose the name in double or single quotation marks (for example, "my lbgroup" or 'my lbgroup'). Minimum length = 1
vservername	<String>	Read-write	Virtual server name.
__count	<Double>	Read-write	count parameter

Operations

(click to see Properties)

ADD:| DELETE:| GET| GET (ALL)| COUNT

Some options that you can use for each operations:

- **Getting warnings in response:**NITRO allows you to get warnings in an operation by specifying the "warning" query parameter as "yes". For example, to get warnings while connecting to the Citrix ADC appliance, the URL is as follows:

  http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/login?warning=yes

  If any, the warnings are displayed in the response payload with the HTTP code "209 X-NITRO-WARNING".

- **Authenticated access for individual NITRO operations:**NITRO allows you to logon to the Citrix ADC appliance to perform individual operations. You can use this option instead of

creating a NITRO session (using the login object) and then using that session to perform all operations,

To do this, you must specify the username and password in the request header of the NITRO request as follows:

X-NITRO-USER:<username>

X-NITRO-PASS:<password>

Note: In such cases, make sure that the request header DOES not include the following:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue>

`*Note:`*

Mandatory parameters are marked in redand placeholder content is marked in <green>.

add

URL:http://<Citrix-ADC-IP-address(NSIP)/nitro/v1/config/lbgroup_lbvserver_binding

HTTP Method:PUT

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Content-Type:application/json

Request Payload:

```
{

"lbgroup_lbvserver_binding":{

<b>"name":<String_value>,

</b><b>"vservername":<String_value></b>

}}
```

Response:

HTTP Status Code on Success: 201 Created HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error

delete

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/lbgroup_lbvserver_binding/name_value<String>

Query-parameters:

args

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/lbgroup_lbvserver_binding/name_value<String>?**args=vservername:<String_value>**

HTTP Method:DELETE

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue>

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error

get

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/lbgroup_lbvserver_binding/name_value<String>

Query-parameters:

filter

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/lbgroup_lbvserver_binding/name_value<String>?**filter=property-name1:property-value1,property-name2:property-value2**

Use this query-parameter to get the filtered set of lbgroup_lbvserver_binding resources configured on Citrix ADC. Filtering can be done on any of the properties of the resource.

pagination

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/lbgroup_lbvserver_binding/name_value<String>?**pagesize=#no;pageno=#no**

Use this query-parameter to get the lbgroup_lbvserver_binding resources in chunks.

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "lbgroup_lbvserver_binding": [ {

"name":<String_value>,

"vservername":<String_value>

}]}
```

get (all)

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/lbgroup_lbvserver_binding

Query-parameters:

bulkbindings

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/lbgroup_lbvserver_binding?**bulkbindings=yes**

NITRO allows you to fetch bindings in bulk.

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "lbgroup_lbvserver_binding": [ {

"name":<String_value>,

"vservername":<String_value>

}]}
```

count

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/lbgroup_lbvserver_binding/name_value<String>?**count=yes**

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{"lbgroup_lbvserver_binding": [ { "__count": "#no"} ] }
```

lbmetrictable

Configuration for metric table resource.

Properties

(click to see Operations)

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
metrictable	<String>	Read-write	Name for the metric table. Must begin with an ASCII alphanumeric or underscore (_) character, and must contain only ASCII alphanumeric, underscore, hash (#), period (.), space, colon (:), at (@), equals (=), and hyphen (-) characters. CLI Users: If the name includes one or more spaces, enclose the name in double or single quotation marks (for example, "my

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
			metrictable" or 'my metrictable'). Minimum length = 1 Maximum length = 31
metric	<String>	Read-write	Name of the metric for which to change the SNMP OID. Minimum length = 1
Snmpoid	<String>	Read-write	New SNMP OID of the metric. Minimum length = 1
metrictype	<String>	Read-only	Indication if it is a configured or internal. Possible values = INTERNAL, CONFIGURED
__count	<Double>	Read-only	count parameter

Operations

(click to see Properties)

ADD| DELETE| UPDATE| GET (ALL)| GET| COUNT

Some options that you can use for each operations:

- **Getting warnings in response:**NITRO allows you to get warnings in an operation by specifying the "warning" query parameter as "yes". For example, to get warnings while connecting to the Citrix ADC appliance, the URL is as follows:

  http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/login?warning=yes

If any, the warnings are displayed in the response payload with the HTTP code "209 X-NITRO-WARNING".

- **Authenticated access for individual NITRO operations:**NITRO allows you to logon to the Citrix ADC appliance to perform individual operations. You can use this option instead of creating a NITRO session (using the login object) and then using that session to perform all operations,

  To do this, you must specify the username and password in the request header of the NITRO request as follows:

  X-NITRO-USER:<username>

  X-NITRO-PASS:<password>

  Note: In such cases, make sure that the request header DOES not include the following:

  Cookie:NITRO_AUTH_TOKEN=<tokenvalue>

`*Note:`*

Mandatory parameters are marked in redand placeholder content is marked in <green>.

add

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/lbmetrictable

HTTP Method:POST

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Content-Type:application/json

Request Payload:

```
{"lbmetrictable":{

<b>"metrictable":<String_value>

</b>}}
```

Response:

HTTP Status Code on Success: 201 Created HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error

delete

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/lbmetrictable/metrictable_value<String>

HTTP Method:DELETE

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue>

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error

update

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/lbmetrictable

HTTP Method:PUT

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Content-Type:application/json

Request Payload:

```
{"lbmetrictable":{

<b>"metrictable":<String_value>,

</b><b>"metric":<String_value>,

</b><b>"Snmpoid":<String_value>

</b>}}
```

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error

get (all)

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/lbmetrictable

Query-parameters:

attrs

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/lbmetrictable?**attrs=property-name1,property-name2**

Use this query parameter to specify the resource details that you want to retrieve.

filter

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/lbmetrictable?**filter=property-name1:property-val1,property-name2:property-val2**

Use this query-parameter to get the filtered set of lbmetrictable resources configured on Citrix ADC. Filtering can be done on any of the properties of the resource.

view

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/lbmetrictable?**view=summary**

Note: By default, the retrieved results are displayed in detail view (?view=detail).

pagination

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/lbmetrictable?**pagesize=#no;pageno=#no**

Use this query-parameter to get the lbmetrictable resources in chunks.

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "lbmetrictable": [ {

"metrictable":<String_value>,

"metric":<String_value>,

"Snmpoid":<String_value>,

"flags":<Boolean_value>,

"metrictype":<String_value>

}]}
```

get

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/lbmetrictable/metrictable_value<String>

Query-parameters:

attrs

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/lbmetrictable/metrictable_value<String>?**attrs=property-name1,property-name2**

Use this query parameter to specify the resource details that you want to retrieve.

view

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/lbmetrictable/metrictable_value<String>?**view=summary**

Note: By default, the retrieved results are displayed in detail view (?view=detail).

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "lbmetrictable": [ {

"metrictable":<String_value>,

"metric":<String_value>,

"Snmpoid":<String_value>,

"flags":<Boolean_value>,

"metrictype":<String_value>

}]}
```

count

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/lbmetrictable?**count=yes**

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "lbmetrictable": [ { "__count": "#no"} ] }
```

lbmetrictable_binding

Binding object which returns the resources bound to lbmetrictable.

Properties

(click to see Operations)

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
metrictable	<String>	Read-write	Name of the metric table. Minimum length = 1 Maximum length = 31
lbmetrictable_metric_binding	<lbmetrictable_metric_binding[]>	Read-only	metric that can be bound to lbmetrictable.

Operations

(click to see Properties)

GET| GET (ALL)

Some options that you can use for each operations:

- **Getting warnings in response:**NITRO allows you to get warnings in an operation by specifying the "warning" query parameter as "yes". For example, to get warnings while connecting to the Citrix ADC appliance, the URL is as follows:

  http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/login?warning=yes

  If any, the warnings are displayed in the response payload with the HTTP code "209 X-NITRO-WARNING".

- **Authenticated access for individual NITRO operations:**NITRO allows you to logon to the Citrix ADC appliance to perform individual operations. You can use this option instead of creating a NITRO session (using the login object) and then using that session to perform all operations,

  To do this, you must specify the username and password in the request header of the NITRO request as follows:

  X-NITRO-USER:<username>

  X-NITRO-PASS:<password>

  Note: In such cases, make sure that the request header DOES not include the following:

  Cookie:NITRO_AUTH_TOKEN=<tokenvalue>

`*Note:`*

Mandatory parameters are marked in redand placeholder content is marked in <green>.

get

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/lbmetrictable_binding/metrictable_value<String>

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "lbmetrictable_binding": [ {

"metrictable":<String_value>,

"lbmetrictable_metric_binding":<lbmetrictable_metric_binding[]_value>

}]}
```

get (all)

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/lbmetrictable_binding

Query-parameters:

bulkbindings

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/lbmetrictable_binding?**bulkbindings=yes**

NITRO allows you to fetch bindings in bulk.

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "lbmetrictable_binding": [ {

"metrictable":<String_value>,

"lbmetrictable_metric_binding":<lbmetrictable_metric_binding[]_value>

}]}
```

lbmetrictable_metric_binding

Binding object showing the metric that can be bound to lbmetrictable.

Properties

(click to see Operations)

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
metric	<String>	Read-write	Name of the metric for which to change the SNMP OID. Minimum length = 1
metrictable	<String>	Read-write	Name of the metric table. Minimum length = 1 Maximum length = 31
Snmpoid	<String>	Read-write	New SNMP OID of the metric. Minimum length = 1
metrictype	<String>	Read-only	Indication if it is a configured or internal.

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
			Possible values = INTERNAL, CONFIGURED
__count	<Double>	Read-only	count parameter

Operations

(click to see Properties)

ADD:| DELETE:| GET| GET (ALL)| COUNT

Some options that you can use for each operations:

- **Getting warnings in response:**NITRO allows you to get warnings in an operation by specifying the "warning" query parameter as "yes". For example, to get warnings while connecting to the Citrix ADC appliance, the URL is as follows:

  http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/login?warning=yes

  If any, the warnings are displayed in the response payload with the HTTP code "209 X-NITRO-WARNING".

- **Authenticated access for individual NITRO operations:**NITRO allows you to logon to the Citrix ADC appliance to perform individual operations. You can use this option instead of creating a NITRO session (using the login object) and then using that session to perform all operations,

  To do this, you must specify the username and password in the request header of the NITRO request as follows:

  X-NITRO-USER:<username>

  X-NITRO-PASS:<password>

  Note: In such cases, make sure that the request header DOES not include the following:

  Cookie:NITRO_AUTH_TOKEN=<tokenvalue>

`*Note:`*

Mandatory parameters are marked in redand placeholder content is marked in <green>.

add

URL:http://<Citrix-ADC-IP-address(NSIP)/nitro/v1/config/lbmetrictable_metric_binding

HTTP Method:PUT

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Content-Type:application/json

Request Payload:

```
{

"lbmetrictable_metric_binding":{

<b>"metrictable":<String_value>,

</b><b>"metric":<String_value>,

</b><b>"Snmpoid":<String_value></b>

}}
```

Response:

HTTP Status Code on Success: 201 Created HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error

delete

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/lbmetrictable_metric_binding/metrictable_value<String>

Query-parameters:

args

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/lbmetrictable_metric_binding/metrictable_value<String>?**args=metric:<String_value>**

HTTP Method:DELETE

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue>

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error

get

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/lbmetrictable_metric_binding/metrictable_value<String>

Query-parameters:

filter

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/lbmetrictable_metric_binding/metrictable_value<String>?**filter=property-name1:property-value1,property-name2:property-value2**

Use this query-parameter to get the filtered set of lbmetrictable_metric_binding resources configured on Citrix ADC. Filtering can be done on any of the properties of the resource.

pagination

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/lbmetrictable_metric_binding/metrictable_value<String>?**pagesize=#no;pageno=#no**

Use this query-parameter to get the lbmetrictable_metric_binding resources in chunks.

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "lbmetrictable_metric_binding": [ {

"metric":<String_value>,

"metrictable":<String_value>,

"Snmpoid":<String_value>,

"metrictype":<String_value>

}]}
```

get (all)

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/lbmetrictable_metric_binding

Query-parameters:

bulkbindings

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/lbmetrictable_metric_binding?**bulkbindings=yes**

NITRO allows you to fetch bindings in bulk.

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "lbmetrictable_metric_binding": [ {

"metric":<String_value>,

"metrictable":<String_value>,

"Snmpoid":<String_value>,

"metrictype":<String_value>

}]}
```

count

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/lbmetrictable_metric_binding/metrictable_value<String>?**count=yes**

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{"lbmetrictable_metric_binding": [ { "__count": "#no"} ] }
```

lbmonbindings

Configuration for monitro bindings resource.

Properties

(click to see Operations)

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
monitorname	<String>	Read-write	The name of the monitor. Minimum length = 1
type	<String>	Read-only	The type of monitor. Possible values = PING, TCP, HTTP, TCP-ECV, HTTP-ECV, UDP-ECV, DNS, FTP, LDNS-PING, LDNS-TCP, LDNS-DNS, RADIUS, USER, HTTP-INLINE, SIP-UDP, SIP-TCP, LOAD, FTP-EXTENDED, SMTP, SNMP, NNTP, MYSQL, MYSQL-ECV, MSSQL-ECV, ORACLE-ECV, LDAP, POP3, CITRIX-XML-SERVICE, CITRIX-WEB-INTERFACE,

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
			DNS-TCP, RTSP, ARP, CITRIX-AG, CITRIX-AAC-LOGINPAGE, CITRIX-AAC-LAS, CITRIX-XD-DDC, ND6, CITRIX-WI-EXTENDED, DIAMETER, RADIUS_ACCOUNTING, STOREFRONT, APPC, SMPP, CITRIX-XNC-ECV, CITRIX-XDM, CITRIX-STA-SERVICE, CITRIX-STA-SERVICE-NHOP
state	<String>	Read-only	The state of the monitor. Possible values = ENABLED, DISABLED
boundservicegroups vrstate	<String>	Read-only	The state of the servicegroup. Possible values = ENABLED, DISABLED
__count	<Double>	Read-only	count parameter

Operations

(click to see Properties)

GET (ALL)| GET| COUNT

Some options that you can use for each operations:

- **Getting warnings in response:**NITRO allows you to get warnings in an operation by specifying the "warning" query parameter as "yes". For example, to get warnings while connecting to the Citrix ADC appliance, the URL is as follows:

  http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/login?warning=yes

  If any, the warnings are displayed in the response payload with the HTTP code "209 X-NITRO-WARNING".

- **Authenticated access for individual NITRO operations:**NITRO allows you to logon to the Citrix ADC appliance to perform individual operations. You can use this option instead of creating a NITRO session (using the login object) and then using that session to perform all operations,

  To do this, you must specify the username and password in the request header of the NITRO request as follows:

  X-NITRO-USER:<username>

  X-NITRO-PASS:<password>

  Note: In such cases, make sure that the request header DOES not include the following:

  Cookie:NITRO_AUTH_TOKEN=<tokenvalue>

`*Note:`*

Mandatory parameters are marked in redand placeholder content is marked in <green>.

get (all)

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/lbmonbindings

Query-parameters:

args

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/lbmonbindings?
args=monitorname:<String_value>,

Use this query-parameter to get lbmonbindings resources based on additional properties.

attrs

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/lbmonbindings?**attrs=property-name1,property-name2**

Use this query parameter to specify the resource details that you want to retrieve.

filter

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/lbmonbindings?**filter=property-name1:property-val1,property-name2:property-val2**

Use this query-parameter to get the filtered set of lbmonbindings resources configured on Citrix ADC. Filtering can be done on any of the properties of the resource.

view

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/lbmonbindings?**view=summary**

Note: By default, the retrieved results are displayed in detail view (?view=detail).

pagination

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/lbmonbindings?**pagesize=#no;pageno=#no**

Use this query-parameter to get the lbmonbindings resources in chunks.

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "lbmonbindings": [ {

<b>monitorname:<String_value>,</b>"type":<String_value>,

"state":<String_value>,

"boundservicegroupsvrstate":<String_value>

}]}
```

get

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/lbmonbindings/monitorname_value<String>

Query-parameters:

attrs

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/lbmonbindings/monitorname_value<String>?**attrs=property-name1,property-name2**

Use this query parameter to specify the resource details that you want to retrieve.

view

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/lbmonbindings/monitorname_value<String>?**view=summary**

Note: By default, the retrieved results are displayed in detail view (?view=detail).

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "lbmonbindings": [ {

<b>monitorname:<String_value>,</b>"type":<String_value>,

"state":<String_value>,

"boundservicegroupsvrstate":<String_value>

}]}
```

count

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/lbmonbindings?**args=monitorname:<String_value>,;count=yes**

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "lbmonbindings": [ { "__count": "#no"} ] }
```

lbmonbindings_binding

Binding object which returns the resources bound to lbmonbindings.

Properties

(click to see Operations)

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
monitorname	<String>	Read-write	The name of the monitor. Minimum length = 1
lbmonbindings_gslbservicegroup_binding	<lbmonbindings_gslbservicegroup_binding[]>	Read-only	gslbservicegroup that can be bound to lbmonbindings.
lbmonbindings_servicegroup_binding	<lbmonbindings_servicegroup_binding[]>	Read-only	servicegroup that can be bound to lbmonbindings.
lbmonbindings_service_binding	<lbmonbindings_service_binding[]>	Read-only	service that can be bound to lbmonbindings.

Operations

(click to see Properties)

GET| GET (ALL)

Some options that you can use for each operations:

- **Getting warnings in response:**NITRO allows you to get warnings in an operation by specifying the "warning" query parameter as "yes". For example, to get warnings while connecting to the Citrix ADC appliance, the URL is as follows:

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/login?warning=yes

If any, the warnings are displayed in the response payload with the HTTP code "209 X-NITRO-WARNING".

- **Authenticated access for individual NITRO operations:**NITRO allows you to logon to the Citrix ADC appliance to perform individual operations. You can use this option instead of creating a NITRO session (using the login object) and then using that session to perform all operations,

  To do this, you must specify the username and password in the request header of the NITRO request as follows:

  X-NITRO-USER:<username>

  X-NITRO-PASS:<password>

  Note: In such cases, make sure that the request header DOES not include the following:

  Cookie:NITRO_AUTH_TOKEN=<tokenvalue>

`*Note:`*

Mandatory parameters are marked in redand placeholder content is marked in <green>.

get

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/lbmonbindings_binding/monitorname_value<String>

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "lbmonbindings_binding": [ {

"monitorname":<String_value>,

"lbmonbindings_service_binding":<lbmonbindings_service_binding[]_value>,

"lbmonbindings_servicegroup_binding":<lbmonbindings_servicegroup_binding[]
_value>,

"lbmonbindings_gslbservicegroup_binding":<lbmonbindings_gslbservicegroup_b
inding[]_value>

}]}
```

get (all)

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/lbmonbindings_binding

Query-parameters:

bulkbindings

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/lbmonbindings_binding?**bulkbindings=yes**

NITRO allows you to fetch bindings in bulk.

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "lbmonbindings_binding": [ {

"monitorname":<String_value>,

"lbmonbindings_service_binding":<lbmonbindings_service_binding[]_value>,

"lbmonbindings_servicegroup_binding":<lbmonbindings_servicegroup_binding[]
_value>,

"lbmonbindings_gslbservicegroup_binding":<lbmonbindings_gslbservicegroup_b
inding[]_value>

}]}
```

lbmonbindings_servicegroup_binding

Binding object showing the servicegroup that can be bound to lbmonbindings.

Properties

(click to see Operations)

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
monitorname	<String>	Read-write	The name of the monitor. Minimum length = 1
servicegroupname	<String>	Read-write	The name of the service group.
boundservicegroups vrstate	<String>	Read-only	The state of the servicegroup. Possible values = ENABLED, DISABLED
monstate	<String>	Read-only	The configured state (enable/disable) of

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
			Monitor on this service. Possible values = ENABLED, DISABLED
servicetype	<String>	Read-only	The type of service. Possible values = HTTP, FTP, TCP, UDP, SSL, SSL_BRIDGE, SSL_TCP, DTLS, NNTP, RPCSVR, DNS, ADNS, SNMP, RTSP, DHCPRA, ANY, SIP_UDP, SIP_TCP, SIP_SSL, DNS_TCP, ADNS_TCP, MYSQL, MSSQL, ORACLE, RADIUS, RADIUSListener, RDP, DIAMETER, SSL_DIAMETER, TFTP, SMPP, PPTP, GRE, SYSLOGTCP, SYSLOGUDP, FIX, SSL_FIX, USER_TCP, USER_SSL_TCP, QUIC, IPFIX, LOGSTREAM
__count	<Double>	Read-only	count parameter

Operations

(click to see Properties)

GET| GET (ALL)| COUNT

Some options that you can use for each operations:

- **Getting warnings in response:**NITRO allows you to get warnings in an operation by specifying the "warning" query parameter as "yes". For example, to get warnings while connecting to the Citrix ADC appliance, the URL is as follows:

  http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/login?warning=yes

  If any, the warnings are displayed in the response payload with the HTTP code "209 X-NITRO-WARNING".

- **Authenticated access for individual NITRO operations:**NITRO allows you to logon to the Citrix ADC appliance to perform individual operations. You can use this option instead of creating a NITRO session (using the login object) and then using that session to perform all operations,

  To do this, you must specify the username and password in the request header of the NITRO request as follows:

  X-NITRO-USER:<username>

  X-NITRO-PASS:<password>

  Note: In such cases, make sure that the request header DOES not include the following:

  Cookie:NITRO_AUTH_TOKEN=<tokenvalue>

`*Note:`*

Mandatory parameters are marked in redand placeholder content is marked in <green>.

get

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/lbmonbindings_servicegroup_binding/monitorname_value<String>

Query-parameters:

filter

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/lbmonbindings_servicegroup_binding/monitorname_value<String>?**filter=property-name1:property-value1,property-name2:property-value2**

Use this query-parameter to get the filtered set of lbmonbindings_servicegroup_binding resources configured on Citrix ADC. Filtering can be done on any of the properties of the resource.

pagination

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/lbmonbindings_servicegroup_binding/monitorname_value<String>?**pagesize=#no;pageno=#no**

Use this query-parameter to get the lbmonbindings_servicegroup_binding resources in chunks.

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "lbmonbindings_servicegroup_binding": [ {

"monitorname":<String_value>,

"servicegroupname":<String_value>,

"boundservicegroupsvrstate":<String_value>,

"monstate":<String_value>,

"servicetype":<String_value>
```

```
}]}
```

get (all)

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/lbmonbindings_servicegroup_binding

Query-parameters:

bulkbindings

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/lbmonbindings_servicegroup_binding?**bulkbindings=yes**

NITRO allows you to fetch bindings in bulk.

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "lbmonbindings_servicegroup_binding": [ {

"monitorname":<String_value>,

"servicegroupname":<String_value>,

"boundservicegroupsvrstate":<String_value>,

"monstate":<String_value>,
```

```
"servicetype":<String_value>

}]}
```

count

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/lbmonbindings_servicegroup_binding/monitorname_value<String>?**count=yes**

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{"lbmonbindings_servicegroup_binding": [ { "__count": "#no"} ] }
```

lbmonbindings_service_binding

Binding object showing the service that can be bound to lbmonbindings.

Properties

(click to see Operations)

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
servicename	<String>	Read-write	The name of the service.

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
monitorname	<String>	Read-write	The name of the monitor. Minimum length = 1
servicetype	<String>	Read-only	The type of service. Possible values = HTTP, FTP, TCP, UDP, SSL, SSL_BRIDGE, SSL_TCP, DTLS, NNTP, RPCSVR, DNS, ADNS, SNMP, RTSP, DHCPRA, ANY, SIP_UDP, SIP_TCP, SIP_SSL, DNS_TCP, ADNS_TCP, MYSQL, MSSQL, ORACLE, RADIUS, RADIUSListener, RDP, DIAMETER, SSL_DIAMETER, TFTP, SMPP, PPTP, GRE, SYSLOGTCP, SYSLOGUDP, FIX, SSL_FIX, USER_TCP, USER_SSL_TCP, QUIC, IPFIX, LOGSTREAM
svrstate	<String>	Read-only	The state of the service. Possible values = UP, DOWN, UNKNOWN, BUSY,

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
			OUT OF SERVICE, GOING OUT OF SERVICE, DOWN WHEN GOING OUT OF SERVICE, NS_EMPTY_STR, Unknown, DISABLED
monsvcstate	<String>	Read-only	The configured state (enable/disable) of Monitor on this service. Possible values = ENABLED, DISABLED
port	<Integer>	Read-only	The port of the service. Range 1 - 65535 * in CLI is represented as 65535 in NITRO API
ipaddress	<String>	Read-only	The IPAddress of the service.
__count	<Double>	Read-only	count parameter

Operations

(click to see Properties)

GET| GET (ALL)| COUNT

Some options that you can use for each operations:

- **Getting warnings in response:**NITRO allows you to get warnings in an operation by specifying the "warning" query parameter as "yes". For example, to get warnings while connecting to the Citrix ADC appliance, the URL is as follows:

  http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/login?warning=yes

  If any, the warnings are displayed in the response payload with the HTTP code "209 X-NITRO-WARNING".

- **Authenticated access for individual NITRO operations:**NITRO allows you to logon to the Citrix ADC appliance to perform individual operations. You can use this option instead of creating a NITRO session (using the login object) and then using that session to perform all operations,

  To do this, you must specify the username and password in the request header of the NITRO request as follows:

  X-NITRO-USER:<username>

  X-NITRO-PASS:<password>

  Note: In such cases, make sure that the request header DOES not include the following:

  Cookie:NITRO_AUTH_TOKEN=<tokenvalue>

`*Note:`*

Mandatory parameters are marked in redand placeholder content is marked in <green>.

get

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/lbmonbindings_service_binding/monitorname_value<String>

Query-parameters:

filter

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/lbmonbindings_service_binding/monitorname_value<String>?**filter=property-name1:property-value1,property-name2:property-value2**

Use this query-parameter to get the filtered set of lbmonbindings_service_binding resources configured on Citrix ADC. Filtering can be done on any of the properties of the resource.

pagination

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/lbmonbindings_service_binding/monitorname_value<String>?**pagesize=#no;pageno=#no**

Use this query-parameter to get the lbmonbindings_service_binding resources in chunks.

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "lbmonbindings_service_binding": [ {

"servicename":<String_value>,

"monitorname":<String_value>,

"servicetype":<String_value>,

"svrstate":<String_value>,

"monsvcstate":<String_value>,

"port":<Integer_value>,

"ipaddress":<String_value>

}]}
```

get (all)

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/lbmonbindings_service_binding

Query-parameters:

bulkbindings

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/lbmonbindings_service_binding?**bulkbindings=yes**

NITRO allows you to fetch bindings in bulk.

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "lbmonbindings_service_binding": [ {

"servicename":<String_value>,

"monitorname":<String_value>,

"servicetype":<String_value>,

"svrstate":<String_value>,

"monsvcstate":<String_value>,

"port":<Integer_value>,
```

```
"ipaddress":<String_value>

}]}
```

count

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/lbmonbindings_service_binding/monitorname_value<String>?**count=yes**

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{"lbmonbindings_service_binding": [ { "__count": "#no"} ] }
```

lbmonitor

Configuration for monitor resource.

Properties

(click to see Operations)

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
monitorname	<String>	Read-write	Name for the monitor. Must begin

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
			with an ASCII alphanumeric or underscore (_) character, and must contain only ASCII alphanumeric, underscore, hash (#), period (.), space, colon (:), at (@), equals (=), and hyphen (-) characters. CLI Users: If the name includes one or more spaces, enclose the name in double or single quotation marks (for example, "my monitor" or 'my monitor'). Minimum length = 1
type	<String>	Read-write	Type of monitor that you want to create. Possible values = PING, TCP, HTTP, TCP-ECV, HTTP-ECV, UDP-ECV, DNS, FTP, LDNS-PING, LDNS-TCP, LDNS-DNS, RADIUS, USER, HTTP-INLINE, SIP-UDP, SIP-TCP, LOAD, FTP-EXTENDED, SMTP,

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
			SNMP, NNTP, MYSQL, MYSQL-ECV, MSSQL-ECV, ORACLE-ECV, LDAP, POP3, CITRIX-XML-SERVICE, CITRIX-WEB-INTERFACE, DNS-TCP, RTSP, ARP, CITRIX-AG, CITRIX-AAC-LOGINPAGE, CITRIX-AAC-LAS, CITRIX-XD-DDC, ND6, CITRIX-WI-EXTENDED, DIAMETER, RADIUS_ACCOUNTING, STOREFRONT, APPC, SMPP, CITRIX-XNC-ECV, CITRIX-XDM, CITRIX-STA-SERVICE, CITRIX-STA-SERVICE-NHOP
action	<String>	Read-write	Action to perform when the response to an inline monitor (a monitor of type HTTP-INLINE) indicates that the service is down. A service monitored by an inline monitor is

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
			considered DOWN if the response code is not one of the codes that have been specified for the Response Code parameter. Available settings function as follows: * NONE - Do not take any action. However, the show service command and the show lb monitor command indicate the total number of responses that were checked and the number of consecutive error responses received after the last successful probe. * LOG - Log the event in NSLOG or SYSLOG. * DOWN - Mark the service as being down, and then do not direct any traffic to the service until the configured down time has expired. Persistent

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
			connections to the service are terminated as soon as the service is marked as DOWN. Also, log the event in NSLOG or SYSLOG. Default value: DOWN Possible values = NONE, LOG, DOWN
respcode	<String[]>	Read-write	Response codes for which to mark the service as UP. For any other response code, the action performed depends on the monitor type. HTTP monitors and RADIUS monitors mark the service as DOWN, while HTTP-INLINE monitors perform the action indicated by the Action parameter.
httprequest	<String>	Read-write	HTTP request to send to the server (for example, "HEAD /file.html").
rtsprequest	<String>	Read-write	RTSP request to send to the server

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
			(for example, "OPTIONS *").
customheaders	<String>	Read-write	Custom header string to include in the monitoring probes.
maxforwards	<Double>	Read-write	Maximum number of hops that the SIP request used for monitoring can traverse to reach the server. Applicable only to monitors of type SIP-UDP. Default value: 1 Minimum value = 0 Maximum value = 255
sipmethod	<String>	Read-write	SIP method to use for the query. Applicable only to monitors of type SIP-UDP. Possible values = OPTIONS, INVITE, REGISTER
sipuri	<String>	Read-write	SIP URI string to send to the service (for example, sip:sip.test). Applicable only to monitors of type SIP-UDP. Minimum length = 1

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
sipreguri	<String>	Read-write	SIP user to be registered. Applicable only if the monitor is of type SIP-UDP and the SIP Method parameter is set to REGISTER. Minimum length = 1
send	<String>	Read-write	String to send to the service. Applicable to TCP-ECV, HTTP-ECV, and UDP-ECV monitors.
recv	<String>	Read-write	String expected from the server for the service to be marked as UP. Applicable to TCP-ECV, HTTP-ECV, and UDP-ECV monitors.
query	<String>	Read-write	Domain name to resolve as part of monitoring the DNS service (for example, example.com).
querytype	<String>	Read-write	Type of DNS record for which to send monitoring queries. Set to Address for querying A records, AAAA for querying AAAA records, and Zone for querying the SOA record.

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
			Possible values = Address, Zone, AAAA
scriptname	<String>	Read-write	Path and name of the script to execute. The script must be available on the Citrix ADC, in the /nsconfig/monitors/ directory. Minimum length = 1
scriptargs	<String>	Read-write	String of arguments for the script. The string is copied verbatim into the request.
dispatcherip	<String>	Read-write	IP address of the dispatcher to which to send the probe.
dispatcherport	<Integer>	Read-write	Port number on which the dispatcher listens for the monitoring probe.
username	<String>	Read-write	User name with which to probe the RADIUS, NNTP, FTP, FTP-EXTENDED, MYSQL, MSSQL, POP3, CITRIX-AG, CITRIX-XD-DDC, CITRIX-WI-EXTENDED, CITRIX-XNC or CITRIX-XDM server.

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
			Minimum length = 1
password	<String>	Read-write	Password that is required for logging on to the RADIUS, NNTP, FTP, FTP-EXTENDED, MYSQL, MSSQL, POP3, CITRIX-AG, CITRIX-XD-DDC, CITRIX-WI-EXTENDED, CITRIX-XNC-ECV or CITRIX-XDM server. Used in conjunction with the user name specified for the User Name parameter. Minimum length = 1
secondarypassword	<String>	Read-write	Secondary password that users might have to provide to log on to the Access Gateway server. Applicable to CITRIX-AG monitors.
logonpointname	<String>	Read-write	Name of the logon point that is configured for the Citrix Access Gateway Advanced Access Control software. Required if you want to monitor the associated login page or Logon Agent. Applicable to CITRIX-AAC-LAS and CITRIX-AAC-

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
			LOGINPAGE monitors.
lasversion	<String>	Read-write	Version number of the Citrix Advanced Access Control Logon Agent. Required by the CITRIX-AAC-LAS monitor.
radkey	<String>	Read-write	Authentication key (shared secret text string) for RADIUS clients and servers to exchange. Applicable to monitors of type RADIUS and RADIUS_ACCOUNTING. Minimum length = 1
radnasid	<String>	Read-write	NAS-Identifier to send in the Access-Request packet. Applicable to monitors of type RADIUS. Minimum length = 1
radnasip	<String>	Read-write	Network Access Server (NAS) IP address to use as the source IP address when monitoring a RADIUS server. Applicable to monitors of type RADIUS and

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
			RADIUS_ACCOUNTING.
radaccounttype	<Double>	Read-write	Account Type to be used in Account Request Packet. Applicable to monitors of type RADIUS_ACCOUNTING. Default value: 1 Minimum value = 0 Maximum value = 15
radframedip	<String>	Read-write	Source ip with which the packet will go out . Applicable to monitors of type RADIUS_ACCOUNTING.
radapn	<String>	Read-write	Called Station Id to be used in Account Request Packet. Applicable to monitors of type RADIUS_ACCOUNTING. Minimum length = 1
radmsisdn	<String>	Read-write	Calling Stations Id to be used in Account Request Packet. Applicable to monitors of type RADIUS_ACCOUNTING. Minimum length = 1

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
radaccountsession	<String>	Read-write	Account Session ID to be used in Account Request Packet. Applicable to monitors of type RADIUS_ACCOUNTING. Minimum length = 1
lrtm	<String>	Read-write	Calculate the least response times for bound services. If this parameter is not enabled, the appliance does not learn the response times of the bound services. Also used for LRTM load balancing. Possible values = ENABLED, DISABLED
deviation	<Double>	Read-write	Time value added to the learned average response time in dynamic response time monitoring (DRTM). When a deviation is specified, the appliance learns the average response time of bound services and adds the deviation to the average. The final value is then continually adjusted

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
			to accommodate response time variations over time. Specified in milliseconds, seconds, or minutes. Minimum value = 0 Maximum value = 20939
units1	<String>	Read-write	Unit of measurement for the Deviation parameter. Cannot be changed after the monitor is created. Default value: SEC Possible values = SEC, MSEC, MIN
interval	<Integer>	Read-write	Time interval between two successive probes. Must be greater than the value of Response Time-out. Default value: 5 Minimum value = 1 Maximum value = 20940
units3	<String>	Read-write	monitor interval units. Default value: SEC Possible values = SEC, MSEC, MIN

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
resptimeout	<Integer>	Read-write	Amount of time for which the appliance must wait before it marks a probe as FAILED. Must be less than the value specified for the Interval parameter. Note: For UDP-ECV monitors for which a receive string is not configured, response timeout does not apply. For UDP-ECV monitors with no receive string, probe failure is indicated by an ICMP port unreachable error received from the service. Default value: 2 Minimum value = 1 Maximum value = 20939
units4	<String>	Read-write	monitor response timeout units. Default value: SEC Possible values = SEC, MSEC, MIN
resptimeoutthresh	<Double>	Read-write	Response time threshold, specified

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
			as a percentage of the Response Time-out parameter. If the response to a monitor probe has not arrived when the threshold is reached, the appliance generates an SNMP trap called monRespTimeoutAboveThresh. After the response time returns to a value below the threshold, the appliance generates a monRespTimeoutBelowThresh SNMP trap. For the traps to be generated, the "MONITOR-RTO-THRESHOLD" alarm must also be enabled. Minimum value = 0 Maximum value = 100
retries	<Integer>	Read-write	Maximum number of probes to send to establish the state of a service for which a monitoring probe failed. Default value: 3 Minimum value = 1

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
			Maximum value = 127
failureretries	<Integer>	Read-write	Number of retries that must fail, out of the number specified for the Retries parameter, for a service to be marked as DOWN. For example, if the Retries parameter is set to 10 and the Failure Retries parameter is set to 6, out of the ten probes sent, at least six probes must fail if the service is to be marked as DOWN. The default value of 0 means that all the retries must fail if the service is to be marked as DOWN. Minimum value = 0 Maximum value = 32
alertretries	<Integer>	Read-write	Number of consecutive probe failures after which the appliance generates an SNMP trap called monProbeFailed. Minimum value = 0

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
			Maximum value = 32
successretries	<Integer>	Read-write	Number of consecutive successful probes required to transition a service's state from DOWN to UP. Default value: 1 Minimum value = 1 Maximum value = 32
downtime	<Integer>	Read-write	Time duration for which to wait before probing a service that has been marked as DOWN. Expressed in milliseconds, seconds, or minutes. Default value: 30 Minimum value = 1 Maximum value = 20939
units2	<String>	Read-write	Unit of measurement for the Down Time parameter. Cannot be changed after the monitor is created. Default value: SEC Possible values = SEC, MSEC, MIN

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
destip	<String>	Read-write	IP address of the service to which to send probes. If the parameter is set to 0, the IP address of the server to which the monitor is bound is considered the destination IP address.
destport	<Integer>	Read-write	TCP or UDP port to which to send the probe. If the parameter is set to 0, the port number of the service to which the monitor is bound is considered the destination port. For a monitor of type USER, however, the destination port is the port number that is included in the HTTP request sent to the dispatcher. Does not apply to monitors of type PING.
state	<String>	Read-write	State of the monitor. The DISABLED setting disables not only the monitor being configured, but all monitors of the same type, until the parameter is set to ENABLED. If the monitor is bound to a

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
			service, the state of the monitor is not taken into account when the state of the service is determined. Default value: ENABLED Possible values = ENABLED, DISABLED
reverse	<String>	Read-write	Mark a service as DOWN, instead of UP, when probe criteria are satisfied, and as UP instead of DOWN when probe criteria are not satisfied. Default value: NO Possible values = YES, NO
transparent	<String>	Read-write	The monitor is bound to a transparent device such as a firewall or router. The state of a transparent device depends on the responsiveness of the services behind it. If a transparent device is being monitored, a destination IP address must be specified. The probe is sent to the

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
			specified IP address by using the MAC address of the transparent device. Default value: NO Possible values = YES, NO
iptunnel	<String>	Read-write	Send the monitoring probe to the service through an IP tunnel. A destination IP address must be specified. Default value: NO Possible values = YES, NO
tos	<String>	Read-write	Probe the service by encoding the destination IP address in the IP TOS (6) bits. Possible values = YES, NO
tosid	<Double>	Read-write	The TOS ID of the specified destination IP. Applicable only when the TOS parameter is set. Minimum value = 1 Maximum value = 63
secure	<String>	Read-write	Use a secure SSL connection when monitoring a service.

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
			Applicable only to TCP based monitors. The secure option cannot be used with a CITRIX-AG monitor, because a CITRIX-AG monitor uses a secure connection by default. Default value: NO Possible values = YES, NO
validatecred	<String>	Read-write	Validate the credentials of the Xen Desktop DDC server user. Applicable to monitors of type CITRIX-XD-DDC. Default value: NO Possible values = YES, NO
domain	<String>	Read-write	Domain in which the XenDesktop Desktop Delivery Controller (DDC) servers or Web Interface servers are present. Required by CITRIX-XD-DDC and CITRIX-WI-EXTENDED monitors for logging on to the DDC servers and Web

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
			Interface servers, respectively.
ipaddress	<String[]>	Read-write	Set of IP addresses expected in the monitoring response from the DNS server, if the record type is A or AAAA. Applicable to DNS monitors. Minimum length = 1
group	<String>	Read-write	Name of a newsgroup available on the NNTP service that is to be monitored. The appliance periodically generates an NNTP query for the name of the newsgroup and evaluates the response. If the newsgroup is found on the server, the service is marked as UP. If the newsgroup does not exist or if the search fails, the service is marked as DOWN. Applicable to NNTP monitors. Minimum length = 1
filename	<String>	Read-write	Name of a file on the FTP server. The appliance monitors the FTP service by periodically checking the existence of the

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
			file on the server. Applicable to FTP-EXTENDED monitors. Minimum length = 1
basedn	<String>	Read-write	The base distinguished name of the LDAP service, from where the LDAP server can begin the search for the attributes in the monitoring query. Required for LDAP service monitoring. Minimum length = 1
binddn	<String>	Read-write	The distinguished name with which an LDAP monitor can perform the Bind operation on the LDAP server. Optional. Applicable to LDAP monitors. Minimum length = 1
filter	<String>	Read-write	Filter criteria for the LDAP query. Optional. Minimum length = 1
attribute	<String>	Read-write	Attribute to evaluate when the LDAP server responds to the query. Success or failure of the monitoring probe depends on whether the attribute exists in

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
			the response. Optional. Minimum length = 1
database	<String>	Read-write	Name of the database to connect to during authentication. Minimum length = 1
oraclesid	<String>	Read-write	Name of the service identifier that is used to connect to the Oracle database during authentication. Minimum length = 1
sqlquery	<String>	Read-write	SQL query for a MYSQL-ECV or MSSQL-ECV monitor. Sent to the database server after the server authenticates the connection. Minimum length = 1
evalrule	<String>	Read-write	Expression that evaluates the database server's response to a MYSQL-ECV or MSSQL-ECV monitoring query. Must produce a Boolean result. The result determines the state of the server. If the expression

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
			returns TRUE, the probe succeeds. For example, if you want the appliance to evaluate the error message to determine the state of the server, use the rule MYSQL.RES.ROW(10) .TEXT_ELEM(2).EQ("MySQL").
mssqlprotocolversion	<String>	Read-write	Version of MSSQL server that is to be monitored. Default value: 70 Possible values = 70, 2000, 2000SP1, 2005, 2008, 2008R2, 2012, 2014
Snmpoid	<String>	Read-write	SNMP OID for SNMP monitors. Minimum length = 1
snmpcommunity	<String>	Read-write	Community name for SNMP monitors. Minimum length = 1
snmpthreshold	<String>	Read-write	Threshold for SNMP monitors. Minimum length = 1
snmpversion	<String>	Read-write	SNMP version to be used for SNMP monitors.

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
			Possible values = V1, V2
metrictable	<String>	Read-write	Metric table to which to bind metrics. Minimum length = 1 Maximum length = 99
application	<String>	Read-write	Name of the application used to determine the state of the service. Applicable to monitors of type CITRIX-XML-SERVICE. Minimum length = 1
sitepath	<String>	Read-write	URL of the logon page. For monitors of type CITRIX-WEB-INTERFACE, to monitor a dynamic page under the site path, terminate the site path with a slash (/). Applicable to CITRIX-WEB-INTERFACE, CITRIX-WI-EXTENDED and CITRIX-XDM monitors. Minimum length = 1
storename	<String>	Read-write	Store Name. For monitors of type STOREFRONT, STORENAME is an

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
			optional argument defining storefront service store name. Applicable to STOREFRONT monitors. Minimum length = 1
storefrontacctservice	<String>	Read-write	Enable/Disable probing for Account Service. Applicable only to Store Front monitors. For multi-tenancy configuration users my skip account service. Default value: YES Possible values = YES, NO
hostname	<String>	Read-write	Hostname in the FQDN format (Example: porche.cars.org). Applicable to STOREFRONT monitors. Minimum length = 1
netprofile	<String>	Read-write	Name of the network profile. Minimum length = 1 Maximum length = 127
originhost	<String>	Read-write	Origin-Host value for the Capabilities-Exchange-Request

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
			(CER) message to use for monitoring Diameter servers. Minimum length = 1
originrealm	<String>	Read-write	Origin-Realm value for the Capabilities-Exchange-Request (CER) message to use for monitoring Diameter servers. Minimum length = 1
hostipaddress	<String>	Read-write	Host-IP-Address value for the Capabilities-Exchange-Request (CER) message to use for monitoring Diameter servers. If Host-IP-Address is not specified, the appliance inserts the mapped IP (MIP) address or subnet IP (SNIP) address from which the CER request (the monitoring probe) is sent. Minimum length = 1
vendorid	<Double>	Read-write	Vendor-Id value for the Capabilities-Exchange-Request (CER) message to use for monitoring Diameter servers.
productname	<String>	Read-write	Product-Name value for the Capabilities-

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
			Exchange-Request (CER) message to use for monitoring Diameter servers. Minimum length = 1
firmwarerevision	<Double>	Read-write	Firmware-Revision value for the Capabilities-Exchange-Request (CER) message to use for monitoring Diameter servers.
authapplicationid	<Double[]>	Read-write	List of Auth-Application-Id attribute value pairs (AVPs) for the Capabilities-Exchange-Request (CER) message to use for monitoring Diameter servers. A maximum of eight of these AVPs are supported in a monitoring CER message. Minimum value = 0 Maximum value = 4294967295
acctapplicationid	<Double[]>	Read-write	List of Acct-Application-Id attribute value pairs (AVPs) for the Capabilities-Exchange-Request (CER) message to use for monitoring

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
			Diameter servers. A maximum of eight of these AVPs are supported in a monitoring message. Minimum value = 0 Maximum value = 4294967295
inbandsecurityid	<String>	Read-write	Inband-Security-Id for the Capabilities-Exchange-Request (CER) message to use for monitoring Diameter servers. Possible values = NO_INBAND_SECURITY, TLS
supportedvendorids	<Double[]>	Read-write	List of Supported-Vendor-Id attribute value pairs (AVPs) for the Capabilities-Exchange-Request (CER) message to use for monitoring Diameter servers. A maximum eight of these AVPs are supported in a monitoring message. Minimum value = 1 Maximum value = 4294967295
vendorspecificvendorid	<Double>	Read-write	Vendor-Id to use in the Vendor-Specific-Application-Id grouped attribute-

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
			value pair (AVP) in the monitoring CER message. To specify Auth-Application-Id or Acct-Application-Id in Vendor-Specific-Application-Id, use vendorSpecificAuthApplicationIds or vendorSpecificAcctApplicationIds, respectively. Only one Vendor-Id is supported for all the Vendor-Specific-Application-Id AVPs in a CER monitoring message. Minimum value = 1
vendorspecificauthapplicationids	<Double[]>	Read-write	List of Vendor-Specific-Auth-Application-Id attribute value pairs (AVPs) for the Capabilities-Exchange-Request (CER) message to use for monitoring Diameter servers. A maximum of eight of these AVPs are supported in a monitoring message. The specified value is combined with the value of vendorSpecificVendorId to obtain the Vendor-Specific-Application-Id AVP in

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
			the CER monitoring message. Minimum value = 0 Maximum value = 4294967295
vendorspecificacctapplicationids	<Double[]>	Read-write	List of Vendor-Specific-Acct-Application-Id attribute value pairs (AVPs) to use for monitoring Diameter servers. A maximum of eight of these AVPs are supported in a monitoring message. The specified value is combined with the value of vendorSpecificVendorId to obtain the Vendor-Specific-Application-Id AVP in the CER monitoring message. Minimum value = 0 Maximum value = 4294967295
kcdaccount	<String>	Read-write	KCD Account used by MSSQL monitor. Minimum length = 1 Maximum length = 32
storedb	<String>	Read-write	Store the database list populated with

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
			the responses to monitor probes. Used in database specific load balancing if MSSQL-ECV/MYSQL-ECV monitor is configured. Default value: DISABLED Possible values = ENABLED, DISABLED
storefrontcheckbackendservices	<String>	Read-write	This option will enable monitoring of services running on storefront server. Storefront services are monitored by probing to a Windows service that runs on the Storefront server and exposes details of which storefront services are running. Default value: NO Possible values = YES, NO
trofscode	<Double>	Read-write	Code expected when the server is under maintenance.
trofsstring	<String>	Read-write	String expected from the server for the service to be marked as trofs. Applicable

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
			to HTTP-ECV/TCP-ECV monitors.
sslprofile	<String>	Read-write	SSL Profile associated with the monitor. Minimum length = 1 Maximum length = 127
metric	<String>	Read-write	Metric name in the metric table, whose setting is changed. A value zero disables the metric and it will not be used for load calculation. Minimum length = 1 Maximum length = 37
metricthreshold	<Double>	Read-write	Threshold to be used for that metric.
metricweight	<Double>	Read-write	The weight for the specified service metric with respect to others. Minimum value = 1 Maximum value = 100
servicename	<String>	Read-write	The name of the service to which the monitor is bound. Minimum length = 1

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
servicegroupname	<String>	Read-write	The name of the service group to which the monitor is to be bound. Minimum length = 1
lrtmconf	<Integer>	Read-only	State of LRTM configuration on the monitor.
lrtmconfstr	<String>	Read-only	State of LRTM configuration on the monitor as STRING. Possible values = ENABLED, DISABLED
dynamicresponsetimeout	<Integer>	Read-only	Response timeout of the DRTM enabled monitor , calculated dynamically based on the history and current response time.
dynamicinterval	<Integer>	Read-only	Interval between monitoring probes for DRTM enabled monitor , calculated dynamically based monitor response time.
multimetrictable	<String[]>	Read-only	Metric table to which to bind metrics, to be used only for output purposes.
dup_state	<String>	Read-only	.

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
			Default value: ENABLED Possible values = ENABLED, DISABLED
dup_weight	<Double>	Read-only	. Default value: 1 Minimum value = 1 Maximum value = 100
weight	<Double>	Read-only	. Minimum value = 1 Maximum value = 100
__count	<Double>	Read-only	count parameter

Operations

(click to see Properties)

ADD| DELETE| UPDATE| UNSET| ENABLE| DISABLE| GET (ALL)| GET| COUNT

Some options that you can use for each operations:

- **Getting warnings in response:**NITRO allows you to get warnings in an operation by specifying the "warning" query parameter as "yes". For example, to get warnings while connecting to the Citrix ADC appliance, the URL is as follows:

  http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/login?warning=yes

  If any, the warnings are displayed in the response payload with the HTTP code "209 X-NITRO-WARNING".

- **Authenticated access for individual NITRO operations:**NITRO allows you to logon to the Citrix ADC appliance to perform individual operations. You can use this option instead of creating a NITRO session (using the login object) and then using that session to perform all operations,

  To do this, you must specify the username and password in the request header of the NITRO request as follows:

  X-NITRO-USER:<username>

  X-NITRO-PASS:<password>

  Note: In such cases, make sure that the request header DOES not include the following:

  Cookie:NITRO_AUTH_TOKEN=<tokenvalue>

`*Note:`*

Mandatory parameters are marked in redand placeholder content is marked in <green>.

add

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/lbmonitor

HTTP Method:POST

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Content-Type:application/json

Request Payload:

```
{"lbmonitor":{

<b>"monitorname":<String_value>,

</b><b>"type":<String_value>,

</b>"action":<String_value>,

"respcode":<String[]_value>,

"httprequest":<String_value>,
```

```
"rtsprequest":<String_value>,

"customheaders":<String_value>,

"maxforwards":<Double_value>,

"sipmethod":<String_value>,

"sipuri":<String_value>,

"sipreguri":<String_value>,

"send":<String_value>,

"recv":<String_value>,

"query":<String_value>,

"querytype":<String_value>,

"scriptname":<String_value>,

"scriptargs":<String_value>,

"dispatcherip":<String_value>,

"dispatcherport":<Integer_value>,

"username":<String_value>,

"password":<String_value>,

"secondarypassword":<String_value>,

"logonpointname":<String_value>,

"lasversion":<String_value>,

"radkey":<String_value>,

"radnasid":<String_value>,

"radnasip":<String_value>,
```

```
"radaccounttype":<Double_value>,

"radframedip":<String_value>,

"radapn":<String_value>,

"radmsisdn":<String_value>,

"radaccountsession":<String_value>,

"lrtm":<String_value>,

"deviation":<Double_value>,

"units1":<String_value>,

"interval":<Integer_value>,

"units3":<String_value>,

"resptimeout":<Integer_value>,

"units4":<String_value>,

"resptimeoutthresh":<Double_value>,

"retries":<Integer_value>,

"failureretries":<Integer_value>,

"alertretries":<Integer_value>,

"successretries":<Integer_value>,

"downtime":<Integer_value>,

"units2":<String_value>,

"destip":<String_value>,

"destport":<Integer_value>,

"state":<String_value>,
```

```
"reverse":<String_value>,

"transparent":<String_value>,

"iptunnel":<String_value>,

"tos":<String_value>,

"tosid":<Double_value>,

"secure":<String_value>,

"validatecred":<String_value>,

"domain":<String_value>,

"ipaddress":<String[]_value>,

"group":<String_value>,

"filename":<String_value>,

"basedn":<String_value>,

"binddn":<String_value>,

"filter":<String_value>,

"attribute":<String_value>,

"database":<String_value>,

"oraclesid":<String_value>,

"sqlquery":<String_value>,

"evalrule":<String_value>,

"mssqlprotocolversion":<String_value>,

"Snmpoid":<String_value>,

"snmpcommunity":<String_value>,
```

```
"snmpthreshold":<String_value>,

"snmpversion":<String_value>,

"metrictable":<String_value>,

"application":<String_value>,

"sitepath":<String_value>,

"storename":<String_value>,

"storefrontacctservice":<String_value>,

"hostname":<String_value>,

"netprofile":<String_value>,

"originhost":<String_value>,

"originrealm":<String_value>,

"hostipaddress":<String_value>,

"vendorid":<Double_value>,

"productname":<String_value>,

"firmwarerevision":<Double_value>,

"authapplicationid":<Double[]_value>,

"acctapplicationid":<Double[]_value>,

"inbandsecurityid":<String_value>,

"supportedvendorids":<Double[]_value>,

"vendorspecificvendorid":<Double_value>,

"vendorspecificauthapplicationids":<Double[]_value>,

"vendorspecificacctapplicationids":<Double[]_value>,
```

```
"kcdaccount":<String_value>,

"storedb":<String_value>,

"storefrontcheckbackendservices":<String_value>,

"trofscode":<Double_value>,

"trofsstring":<String_value>,

"sslprofile":<String_value>

}}
```

Response:

HTTP Status Code on Success: 201 Created HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error

delete

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/lbmonitor/monitorname_value<String>

HTTP Method:DELETE

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue>

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error

update

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/lbmonitor

HTTP Method:PUT

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Content-Type:application/json

Request Payload:

```
{"lbmonitor":{

<b>"monitorname":<String_value>,

</b><b>"type":<String_value>,

</b>"action":<String_value>,

"respcode":<String[]_value>,

"httprequest":<String_value>,

"rtsprequest":<String_value>,

"customheaders":<String_value>,

"maxforwards":<Double_value>,

"sipmethod":<String_value>,

"sipreguri":<String_value>,

"sipuri":<String_value>,

"send":<String_value>,

"recv":<String_value>,

"query":<String_value>,

"querytype":<String_value>,

"username":<String_value>,

"password":<String_value>,

"secondarypassword":<String_value>,
```

```
"logonpointname":<String_value>,

"lasversion":<String_value>,

"radkey":<String_value>,

"radnasid":<String_value>,

"radnasip":<String_value>,

"radaccounttype":<Double_value>,

"radframedip":<String_value>,

"radapn":<String_value>,

"radmsisdn":<String_value>,

"radaccountsession":<String_value>,

"lrtm":<String_value>,

"deviation":<Double_value>,

"units1":<String_value>,

"scriptname":<String_value>,

"scriptargs":<String_value>,

"validatecred":<String_value>,

"domain":<String_value>,

"dispatcherip":<String_value>,

"dispatcherport":<Integer_value>,

"interval":<Integer_value>,

"units3":<String_value>,

"resptimeout":<Integer_value>,
```

```
"units4":<String_value>,

"resptimeoutthresh":<Double_value>,

"retries":<Integer_value>,

"failureretries":<Integer_value>,

"alertretries":<Integer_value>,

"successretries":<Integer_value>,

"downtime":<Integer_value>,

"units2":<String_value>,

"destip":<String_value>,

"destport":<Integer_value>,

"state":<String_value>,

"reverse":<String_value>,

"transparent":<String_value>,

"iptunnel":<String_value>,

"tos":<String_value>,

"tosid":<Double_value>,

"secure":<String_value>,

"ipaddress":<String[]_value>,

"group":<String_value>,

"filename":<String_value>,

"basedn":<String_value>,

"binddn":<String_value>,
```

```
"filter":<String_value>,

"attribute":<String_value>,

"database":<String_value>,

"oraclesid":<String_value>,

"sqlquery":<String_value>,

"evalrule":<String_value>,

"Snmpoid":<String_value>,

"snmpcommunity":<String_value>,

"snmpthreshold":<String_value>,

"snmpversion":<String_value>,

"metrictable":<String_value>,

"metric":<String_value>,

"metricthreshold":<Double_value>,

"metricweight":<Double_value>,

"application":<String_value>,

"sitepath":<String_value>,

"storename":<String_value>,

"storefrontacctservice":<String_value>,

"storefrontcheckbackendservices":<String_value>,

"hostname":<String_value>,

"netprofile":<String_value>,

"mssqlprotocolversion":<String_value>,
```

```
"originhost":<String_value>,

"originrealm":<String_value>,

"hostipaddress":<String_value>,

"vendorid":<Double_value>,

"productname":<String_value>,

"firmwarerevision":<Double_value>,

"authapplicationid":<Double[]_value>,

"acctapplicationid":<Double[]_value>,

"inbandsecurityid":<String_value>,

"supportedvendorids":<Double[]_value>,

"vendorspecificvendorid":<Double_value>,

"vendorspecificauthapplicationids":<Double[]_value>,

"vendorspecificacctapplicationids":<Double[]_value>,

"kcdaccount":<String_value>,

"storedb":<String_value>,

"trofscode":<Double_value>,

"trofsstring":<String_value>,

"sslprofile":<String_value>

}}
```

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error

unset

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/lbmonitor?**action=unset**

HTTP Method:POST

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Content-Type:application/json

Request Payload:

```
{"lbmonitor":{

<b>"monitorname":<String_value>,

</b><b>"type":<String_value>,

</b>"ipaddress":true,

"scriptname":true,

"destport":true,

"netprofile":true,

"sslprofile":true,

"action":true,

"respcode":true,

"httprequest":true,

"rtsprequest":true,

"customheaders":true,
```

```
"maxforwards":true,

"sipmethod":true,

"sipreguri":true,

"send":true,

"recv":true,

"query":true,

"querytype":true,

"username":true,

"password":true,

"secondarypassword":true,

"logonpointname":true,

"lasversion":true,

"radkey":true,

"radnasid":true,

"radnasip":true,

"radaccounttype":true,

"radframedip":true,

"radapn":true,

"radmsisdn":true,

"radaccountsession":true,

"lrtm":true,

"deviation":true,
```

```
"scriptargs":true,

"validatecred":true,

"domain":true,

"dispatcherip":true,

"dispatcherport":true,

"interval":true,

"resptimeout":true,

"resptimeoutthresh":true,

"retries":true,

"failureretries":true,

"alertretries":true,

"successretries":true,

"downtime":true,

"destip":true,

"state":true,

"reverse":true,

"transparent":true,

"iptunnel":true,

"tos":true,

"tosid":true,

"secure":true,

"group":true,
```

```
"filename":true,
"basedn":true,
"binddn":true,
"filter":true,
"attribute":true,
"database":true,
"oraclesid":true,
"sqlquery":true,
"Snmpoid":true,
"snmpcommunity":true,
"snmpthreshold":true,
"snmpversion":true,
"metrictable":true,
"mssqlprotocolversion":true,
"originhost":true,
"originrealm":true,
"hostipaddress":true,
"vendorid":true,
"productname":true,
"firmwarerevision":true,
"authapplicationid":true,
"acctapplicationid":true,
```

```
"inbandsecurityid":true,

"supportedvendorids":true,

"vendorspecificvendorid":true,

"vendorspecificauthapplicationids":true,

"vendorspecificacctapplicationids":true,

"kcdaccount":true,

"storedb":true,

"trofscode":true,

"trofsstring":true

}}
```

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error

enable

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/lbmonitor?**action=enable**

HTTP Method:POST

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Content-Type:application/json

Request Payload:

```
{"lbmonitor":{
```

```
"servicename":<String_value>,

"servicegroupname":<String_value>,

"monitorname":<String_value>

}}
```

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error

disable

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/lbmonitor?**action=disable**

HTTP Method:POST

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Content-Type:application/json

Request Payload:

```
{"lbmonitor":{

"servicename":<String_value>,

"servicegroupname":<String_value>,

"monitorname":<String_value>

}}
```

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error

get (all)

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/lbmonitor

Query-parameters:

args

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/lbmonitor?**args=monitorname:<String_value>,type:<String_value>**

Use this query-parameter to get lbmonitor resources based on additional properties.

attrs

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/lbmonitor?**attrs=property-name1,property-name2**

Use this query parameter to specify the resource details that you want to retrieve.

filter

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/lbmonitor?**filter=property-name1:property-val1,property-name2:property-val2**

Use this query-parameter to get the filtered set of lbmonitor resources configured on Citrix ADC. Filtering can be done on any of the properties of the resource.

view

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/lbmonitor?**view=summary**

Note: By default, the retrieved results are displayed in detail view (?view=detail).

pagination

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/lbmonitor?**pagesize=#no;pageno=#no**

Use this query-parameter to get the lbmonitor resources in chunks.

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "lbmonitor": [ {

monitorname:<String_value>,type:<String_value>"interval":<Integer_value>,

"units3":<String_value>,

"resptimeout":<Integer_value>,

"resptimeoutthresh":<Double_value>,

"units4":<String_value>,

"retries":<Integer_value>,

"failureretries":<Integer_value>,

"alertretries":<Integer_value>,

"successretries":<Integer_value>,

"downtime":<Integer_value>,

"units2":<String_value>,

"destip":<String_value>,

"destport":<Integer_value>,
```

```
"state":<String_value>,

"reverse":<String_value>,

"transparent":<String_value>,

"iptunnel":<String_value>,

"tos":<String_value>,

"tosid":<Double_value>,

"secure":<String_value>,

"action":<String_value>,

"respcode":<String[]_value>,

"httprequest":<String_value>,

"rtsprequest":<String_value>,

"send":<String_value>,

"recv":<String_value>,

"query":<String_value>,

"querytype":<String_value>,

"username":<String_value>,

"password":<String_value>,

"secondarypassword":<String_value>,

"logonpointname":<String_value>,

"lasversion":<String_value>,

"validatecred":<String_value>,

"domain":<String_value>,
```

```
"radkey":<String_value>,

"radnasid":<String_value>,

"radnasip":<String_value>,

"radaccounttype":<Double_value>,

"radframedip":<String_value>,

"radapn":<String_value>,

"radmsisdn":<String_value>,

"radaccountsession":<String_value>,

"lrtm":<String_value>,

"lrtmconf":<Integer_value>,

"lrtmconfstr":<String_value>,

"deviation":<Double_value>,

"units1":<String_value>,

"dynamicresponsetimeout":<Integer_value>,

"dynamicinterval":<Integer_value>,

"scriptname":<String_value>,

"scriptargs":<String_value>,

"dispatcherip":<String_value>,

"dispatcherport":<Integer_value>,

"sipuri":<String_value>,

"sipmethod":<String_value>,

"maxforwards":<Double_value>,
```

```
"sipreguri":<String_value>,

"customheaders":<String_value>,

"ipaddress":<String[]_value>,

"group":<String_value>,

"filename":<String_value>,

"basedn":<String_value>,

"binddn":<String_value>,

"filter":<String_value>,

"attribute":<String_value>,

"database":<String_value>,

"oraclesid":<String_value>,

"sqlquery":<String_value>,

"evalrule":<String_value>,

"Snmpoid":<String_value>,

"snmpcommunity":<String_value>,

"snmpthreshold":<String_value>,

"snmpversion":<String_value>,

"metric":<String_value>,

"metrictable":<String_value>,

"multimetrictable":<String[]_value>,

"metricthreshold":<Double_value>,

"metricweight":<Double_value>,
```

```
"application":<String_value>,

"sitepath":<String_value>,

"storename":<String_value>,

"storefrontacctservice":<String_value>,

"storefrontcheckbackendservices":<String_value>,

"hostname":<String_value>,

"netprofile":<String_value>,

"mssqlprotocolversion":<String_value>,

"originhost":<String_value>,

"originrealm":<String_value>,

"hostipaddress":<String_value>,

"vendorid":<Double_value>,

"productname":<String_value>,

"firmwarerevision":<Double_value>,

"authapplicationid":<Double[]_value>,

"acctapplicationid":<Double[]_value>,

"inbandsecurityid":<String_value>,

"supportedvendorids":<Double[]_value>,

"vendorspecificvendorid":<Double_value>,

"vendorspecificauthapplicationids":<Double[]_value>,

"vendorspecificacctapplicationids":<Double[]_value>,

"servicename":<String_value>,
```

```
"dup_state":<String_value>,

"dup_weight":<Double_value>,

"servicegroupname":<String_value>,

"weight":<Double_value>,

"kcdaccount":<String_value>,

"storedb":<String_value>,

"trofscode":<Double_value>,

"trofsstring":<String_value>,

"sslprofile":<String_value>

}]}
```

get

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/lbmonitor/monitorname_value<String>

Query-parameters:

attrs

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/lbmonitor/monitorname_value<String>?**attrs=property-name1,property-name2**

Use this query parameter to specify the resource details that you want to retrieve.

view

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/lbmonitor/monitorname_value<String>?**view=summary**

Note: By default, the retrieved results are displayed in detail view (?view=detail).

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "lbmonitor": [ {

monitorname:<String_value>,type:<String_value>"interval":<Integer_value>,

"units3":<String_value>,

"resptimeout":<Integer_value>,

"resptimeoutthresh":<Double_value>,

"units4":<String_value>,

"retries":<Integer_value>,

"failureretries":<Integer_value>,

"alertretries":<Integer_value>,

"successretries":<Integer_value>,

"downtime":<Integer_value>,

"units2":<String_value>,

"destip":<String_value>,

"destport":<Integer_value>,

"state":<String_value>,
```

```
"reverse":<String_value>,

"transparent":<String_value>,

"iptunnel":<String_value>,

"tos":<String_value>,

"tosid":<Double_value>,

"secure":<String_value>,

"action":<String_value>,

"respcode":<String[]_value>,

"httprequest":<String_value>,

"rtsprequest":<String_value>,

"send":<String_value>,

"recv":<String_value>,

"query":<String_value>,

"querytype":<String_value>,

"username":<String_value>,

"password":<String_value>,

"secondarypassword":<String_value>,

"logonpointname":<String_value>,

"lasversion":<String_value>,

"validatecred":<String_value>,

"domain":<String_value>,

"radkey":<String_value>,
```

```
"radnasid":<String_value>,

"radnasip":<String_value>,

"radaccounttype":<Double_value>,

"radframedip":<String_value>,

"radapn":<String_value>,

"radmsisdn":<String_value>,

"radaccountsession":<String_value>,

"lrtm":<String_value>,

"lrtmconf":<Integer_value>,

"lrtmconfstr":<String_value>,

"deviation":<Double_value>,

"units1":<String_value>,

"dynamicresponsetimeout":<Integer_value>,

"dynamicinterval":<Integer_value>,

"scriptname":<String_value>,

"scriptargs":<String_value>,

"dispatcherip":<String_value>,

"dispatcherport":<Integer_value>,

"sipuri":<String_value>,

"sipmethod":<String_value>,

"maxforwards":<Double_value>,

"sipreguri":<String_value>,
```

```
"customheaders":<String_value>,

"ipaddress":<String[]_value>,

"group":<String_value>,

"filename":<String_value>,

"basedn":<String_value>,

"binddn":<String_value>,

"filter":<String_value>,

"attribute":<String_value>,

"database":<String_value>,

"oraclesid":<String_value>,

"sqlquery":<String_value>,

"evalrule":<String_value>,

"Snmpoid":<String_value>,

"snmpcommunity":<String_value>,

"snmpthreshold":<String_value>,

"snmpversion":<String_value>,

"metric":<String_value>,

"metrictable":<String_value>,

"multimetrictable":<String[]_value>,

"metricthreshold":<Double_value>,

"metricweight":<Double_value>,

"application":<String_value>,
```

```
"sitepath":<String_value>,

"storename":<String_value>,

"storefrontacctservice":<String_value>,

"storefrontcheckbackendservices":<String_value>,

"hostname":<String_value>,

"netprofile":<String_value>,

"mssqlprotocolversion":<String_value>,

"originhost":<String_value>,

"originrealm":<String_value>,

"hostipaddress":<String_value>,

"vendorid":<Double_value>,

"productname":<String_value>,

"firmwarerevision":<Double_value>,

"authapplicationid":<Double[]_value>,

"acctapplicationid":<Double[]_value>,

"inbandsecurityid":<String_value>,

"supportedvendorids":<Double[]_value>,

"vendorspecificvendorid":<Double_value>,

"vendorspecificauthapplicationids":<Double[]_value>,

"vendorspecificacctapplicationids":<Double[]_value>,

"servicename":<String_value>,

"dup_state":<String_value>,
```

```
"dup_weight":<Double_value>,

"servicegroupname":<String_value>,

"weight":<Double_value>,

"kcdaccount":<String_value>,

"storedb":<String_value>,

"trofscode":<Double_value>,

"trofsstring":<String_value>,

"sslprofile":<String_value>

}]}
```

count

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/lbmonitor?**count=yes**

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "lbmonitor": [ { "__count": "#no"} ] }
```

lbmonitor_binding

Binding object which returns the resources bound to lbmonitor.

Properties

(click to see Operations)

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
monitorname	<String>	Read-write	Name of the monitor. Minimum length = 1
lbmonitor_metric_binding	<lbmonitor_metric_binding[]>	Read-only	metric that can be bound to lbmonitor.
lbmonitor_sslcertkey_binding	<lbmonitor_sslcertkey_binding[]>	Read-only	sslcertkey that can be bound to lbmonitor.

Operations

(click to see Properties)

GET| GET (ALL)

Some options that you can use for each operations:

- **Getting warnings in response:**NITRO allows you to get warnings in an operation by specifying the "warning" query parameter as "yes". For example, to get warnings while connecting to the Citrix ADC appliance, the URL is as follows:

  http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/login?warning=yes

  If any, the warnings are displayed in the response payload with the HTTP code "209 X-NITRO-WARNING".
- **Authenticated access for individual NITRO operations:**NITRO allows you to logon to the Citrix ADC appliance to perform individual operations. You can use this option instead of creating a NITRO session (using the login object) and then using that session to perform all operations,

To do this, you must specify the username and password in the request header of the NITRO request as follows:

X-NITRO-USER:<username>

X-NITRO-PASS:<password>

Note: In such cases, make sure that the request header DOES not include the following:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue>

`*Note:`*

Mandatory parameters are marked in redand placeholder content is marked in <green>.

get

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/lbmonitor_binding/monitorname_value<String>

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "lbmonitor_binding": [ {

"monitorname":<String_value>,

"lbmonitor_metric_binding":<lbmonitor_metric_binding[]_value>,

"lbmonitor_sslcertkey_binding":<lbmonitor_sslcertkey_binding[]_value>
```

```
}]}
```

get (all)

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/lbmonitor_binding

Query-parameters:

bulkbindings

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/lbmonitor_binding?**bulkbindings=yes**

NITRO allows you to fetch bindings in bulk.

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "lbmonitor_binding": [ {

"monitorname":<String_value>,

"lbmonitor_metric_binding":<lbmonitor_metric_binding[]_value>,

"lbmonitor_sslcertkey_binding":<lbmonitor_sslcertkey_binding[]_value>

}]}
```

lbmonitor_metric_binding

Binding object showing the metric that can be bound to lbmonitor.

Properties

(click to see Operations)

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
monitorname	<String>	Read-write	Name of the monitor. Minimum length = 1
metric	<String>	Read-write	Metric name in the metric table, whose setting is changed. A value zero disables the metric and it will not be used for load calculation. Minimum length = 1 Maximum length = 37
metricthreshold	<Double>	Read-write	Threshold to be used for that metric.
metricweight	<Double>	Read-write	The weight for the specified service metric with respect to others. Minimum value = 1 Maximum value = 100
metrictable	<String>	Read-only	Metric table to which to bind metrics. Minimum length = 1 Maximum length = 99

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
metric_unit	<String>	Read-only	Giving the unit of the metric. Possible values = Bytes/s, ms, pkts/s, users
__count	<Double>	Read-only	count parameter

Operations

(click to see Properties)

ADD:| DELETE:| GET| GET (ALL)| COUNT

Some options that you can use for each operations:

- **Getting warnings in response:**NITRO allows you to get warnings in an operation by specifying the "warning" query parameter as "yes". For example, to get warnings while connecting to the Citrix ADC appliance, the URL is as follows:

  http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/login?warning=yes

  If any, the warnings are displayed in the response payload with the HTTP code "209 X-NITRO-WARNING".

- **Authenticated access for individual NITRO operations:**NITRO allows you to logon to the Citrix ADC appliance to perform individual operations. You can use this option instead of creating a NITRO session (using the login object) and then using that session to perform all operations,

  To do this, you must specify the username and password in the request header of the NITRO request as follows:

  X-NITRO-USER:<username>

  X-NITRO-PASS:<password>

  Note: In such cases, make sure that the request header DOES not include the following:

  Cookie:NITRO_AUTH_TOKEN=<tokenvalue>

`*Note:` *

Mandatory parameters are marked in redand placeholder content is marked in <green>.

add

URL:http://<Citrix-ADC-IP-address(NSIP)/nitro/v1/config/lbmonitor_metric_binding

HTTP Method:PUT

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Content-Type:application/json

Request Payload:

```
{

"lbmonitor_metric_binding":{

<b>"monitorname":<String_value>,

</b>"metric":<String_value>,

"metricthreshold":<Double_value>,

"metricweight":<Double_value>

}}
```

Response:

HTTP Status Code on Success: 201 Created HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error

delete

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/lbmonitor_metric_binding/monitorname_value<String>

Query-parameters:

args

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/lbmonitor_metric_binding/monitorname_value<String>?**args=metric:<String_value>**

HTTP Method:DELETE

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue>

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error

get

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/lbmonitor_metric_binding/monitorname_value<String>

Query-parameters:

filter

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/lbmonitor_metric_binding/monitorname_value<String>?**filter=property-name1:property-value1,property-name2:property-value2**

Use this query-parameter to get the filtered set of lbmonitor_metric_binding resources configured on Citrix ADC. Filtering can be done on any of the properties of the resource.

pagination

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/lbmonitor_metric_binding/monitorname_value<String>?**pagesize=#no;pageno=#no**

Use this query-parameter to get the lbmonitor_metric_binding resources in chunks.

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "lbmonitor_metric_binding": [ {

"monitorname":<String_value>,

"metric":<String_value>,

"metricthreshold":<Double_value>,

"metricweight":<Double_value>,

"metrictable":<String_value>,

"metric_unit":<String_value>

}]}
```

get (all)

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/lbmonitor_metric_binding

Query-parameters:

bulkbindings

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/lbmonitor_metric_binding?**bulkbindings=yes**

NITRO allows you to fetch bindings in bulk.

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "lbmonitor_metric_binding": [ {

"monitorname":<String_value>,

"metric":<String_value>,

"metricthreshold":<Double_value>,

"metricweight":<Double_value>,

"metrictable":<String_value>,

"metric_unit":<String_value>

}]}
```

count

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/lbmonitor_metric_binding/monitorname_value<String>?**count=yes**

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{"lbmonitor_metric_binding": [ { "__count": "#no"} ] }
```

lbmonitor_servicegroup_binding

Binding object showing the servicegroup that can be bound to lbmonitor.

Properties

(click to see Operations)

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
servicegroupname	<String>	Read-write	Name of the service group. Minimum length = 1
monitorname	<String>	Read-write	Name of the monitor. Minimum length = 1
weight	<Double>	Read-write	Weight to assign to the binding between the monitor and service. Minimum value = 1 Maximum value = 100
dup_state	<String>	Read-write	State of the monitor. The state setting for a monitor of a given type affects all

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
			monitors of that type. For example, if an HTTP monitor is enabled, all HTTP monitors on the appliance are (or remain) enabled. If an HTTP monitor is disabled, all HTTP monitors on the appliance are disabled. Default value: ENABLED Possible values = ENABLED, DISABLED
servicename	<String>	Read-write	Name of the service or service group. Minimum length = 1
state	<String>	Read-write	State of the monitor. The state setting for a monitor of a given type affects all monitors of that type. For example, if an HTTP monitor is enabled, all HTTP monitors on the appliance are (or remain) enabled. If an HTTP monitor is disabled, all HTTP monitors on the appliance are disabled.

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
			Default value: ENABLED Possible values = ENABLED, DISABLED
dup_weight	<Double>	Read-write	Weight to assign to the binding between the monitor and service. Default value: 1 Minimum value = 1 Maximum value = 100

Operations

(click to see Properties)

ADD:| DELETE:

Some options that you can use for each operations:

- **Getting warnings in response:**NITRO allows you to get warnings in an operation by specifying the "warning" query parameter as "yes". For example, to get warnings while connecting to the Citrix ADC appliance, the URL is as follows:

  http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/login?warning=yes

  If any, the warnings are displayed in the response payload with the HTTP code "209 X-NITRO-WARNING".

- **Authenticated access for individual NITRO operations:**NITRO allows you to logon to the Citrix ADC appliance to perform individual operations. You can use this option instead of creating a NITRO session (using the login object) and then using that session to perform all operations,

  To do this, you must specify the username and password in the request header of the NITRO request as follows:

X-NITRO-USER:<username>

X-NITRO-PASS:<password>

Note: In such cases, make sure that the request header DOES not include the following:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue>

`*Note:`*

Mandatory parameters are marked in redand placeholder content is marked in <green>.

add

URL:http://<Citrix-ADC-IP-address(NSIP)/nitro/v1/config/lbmonitor_servicegroup_binding

HTTP Method:PUT

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Content-Type:application/json

Request Payload:

```
{

"lbmonitor_servicegroup_binding":{

<b>"monitorname":<String_value>,

</b>"servicename":<String_value>,

"dup_state":<String_value>,

"dup_weight":<Double_value>,

"servicegroupname":<String_value>,

"state":<String_value>,

"weight":<Double_value>

}}
```

Response:

HTTP Status Code on Success: 201 Created HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error

delete

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/lbmonitor_servicegroup_binding/monitorname_value<String>

Query-parameters:

args

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/lbmonitor_servicegroup_binding/monitorname_value<String>?**args=servicename:<String_value>,servicegroupname:<String_value>**

HTTP Method:DELETE

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue>

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error

lbmonitor_service_binding

Binding object showing the service that can be bound to lbmonitor.

Properties

(click to see Operations)

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
servicegroupname	<String>	Read-write	Name of the service group. Minimum length = 1
monitorname	<String>	Read-write	Name of the monitor. Minimum length = 1
weight	<Double>	Read-write	Weight to assign to the binding between the monitor and service. Minimum value = 1 Maximum value = 100
dup_state	<String>	Read-write	State of the monitor. The state setting for a monitor of a given type affects all monitors of that type. For example, if an HTTP monitor is enabled, all HTTP monitors on the appliance are (or remain) enabled. If an HTTP monitor is disabled, all HTTP monitors on the appliance are disabled. Default value: ENABLED Possible values = ENABLED, DISABLED

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
servicename	<String>	Read-write	Name of the service or service group. Minimum length = 1
state	<String>	Read-write	State of the monitor. The state setting for a monitor of a given type affects all monitors of that type. For example, if an HTTP monitor is enabled, all HTTP monitors on the appliance are (or remain) enabled. If an HTTP monitor is disabled, all HTTP monitors on the appliance are disabled. Default value: ENABLED Possible values = ENABLED, DISABLED
dup_weight	<Double>	Read-write	Weight to assign to the binding between the monitor and service. Default value: 1 Minimum value = 1 Maximum value = 100

Operations

(click to see Properties)

ADD:| DELETE:

Some options that you can use for each operations:

- **Getting warnings in response:**NITRO allows you to get warnings in an operation by specifying the "warning" query parameter as "yes". For example, to get warnings while connecting to the Citrix ADC appliance, the URL is as follows:

  http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/login?warning=yes

  If any, the warnings are displayed in the response payload with the HTTP code "209 X-NITRO-WARNING".

- **Authenticated access for individual NITRO operations:**NITRO allows you to logon to the Citrix ADC appliance to perform individual operations. You can use this option instead of creating a NITRO session (using the login object) and then using that session to perform all operations,

  To do this, you must specify the username and password in the request header of the NITRO request as follows:

  X-NITRO-USER:<username>

  X-NITRO-PASS:<password>

  Note: In such cases, make sure that the request header DOES not include the following:

  Cookie:NITRO_AUTH_TOKEN=<tokenvalue>

`*Note:`*

Mandatory parameters are marked in redand placeholder content is marked in <green>.

add

URL:http://<Citrix-ADC-IP-address(NSIP)/nitro/v1/config/lbmonitor_service_binding

HTTP Method:PUT

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Content-Type:application/json

Request Payload:

```
{
"lbmonitor_service_binding":{
<b>"monitorname":<String_value>,
</b>"servicename":<String_value>,
"dup_state":<String_value>,
"dup_weight":<Double_value>,
"servicegroupname":<String_value>,
"state":<String_value>,
"weight":<Double_value>
}}
```

Response:

HTTP Status Code on Success: 201 Created HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error

delete

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/lbmonitor_service_binding/monitorname_value<String>

Query-parameters:

args

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/lbmonitor_service_binding/monitorname_value<String>?
args=servicename:<String_value>,servicegroupname:<String_value>

HTTP Method:DELETE

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue>

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error

lbmonitor_sslcertkey_binding

Binding object showing the sslcertkey that can be bound to lbmonitor.

Properties

(click to see Operations)

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
certkeyname	<String>	Read-write	The name of the certificate bound to the monitor.
monitorname	<String>	Read-write	Name of the monitor. Minimum length = 1
ocspcheck	<String>	Read-write	The state of the OCSP check parameter. (Mandatory/ Optional). Possible values = Mandatory, Optional
crlcheck	<String>	Read-write	The state of the CRL check parameter. (Mandatory/ Optional). Possible values = Mandatory, Optional

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
ca	<Boolean>	Read-write	The rule for use of CRL corresponding to this CA certificate during client authentication. If crlCheck is set to Mandatory, the system will deny all SSL clients if the CRL is missing, expired - NextUpdate date is in the past, or is incomplete with remote CRL refresh enabled. If crlCheck is set to optional, the system will allow SSL clients in the above error cases.However, in any case if the client certificate is revoked in the CRL, the SSL client will be denied access.
__count	<Double>	Read-write	count parameter

Operations

(click to see Properties)

ADD:| DELETE:| GET| GET (ALL)| COUNT

Some options that you can use for each operations:

- **Getting warnings in response:**NITRO allows you to get warnings in an operation by specifying the "warning" query parameter as "yes". For example, to get warnings while connecting to the Citrix ADC appliance, the URL is as follows:

  http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/login?warning=yes

If any, the warnings are displayed in the response payload with the HTTP code "209 X-NITRO-WARNING".

- **Authenticated access for individual NITRO operations:**NITRO allows you to logon to the Citrix ADC appliance to perform individual operations. You can use this option instead of creating a NITRO session (using the login object) and then using that session to perform all operations,

  To do this, you must specify the username and password in the request header of the NITRO request as follows:

  X-NITRO-USER:<username>

  X-NITRO-PASS:<password>

  Note: In such cases, make sure that the request header DOES not include the following:

  Cookie:NITRO_AUTH_TOKEN=<tokenvalue>

`*Note:`*

Mandatory parameters are marked in redand placeholder content is marked in <green>.

add

URL:http://<Citrix-ADC-IP-address(NSIP)/nitro/v1/config/lbmonitor_sslcertkey_binding

HTTP Method:PUT

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Content-Type:application/json

Request Payload:

```
{

"lbmonitor_sslcertkey_binding":{

<b>"monitorname":<String_value>,

</b>"certkeyname":<String_value>,
```

```
"ca":<Boolean_value>,

"crlcheck":<String_value>,

"ocspcheck":<String_value>

}}
```

Response:

HTTP Status Code on Success: 201 Created HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error

delete

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/lbmonitor_sslcertkey_binding/monitorname_value<String>

Query-parameters:

args

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/lbmonitor_sslcertkey_binding/monitorname_value<String>?**args=certkeyname:<String_value>,ca:<Boolean_value>**

HTTP Method:DELETE

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue>

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error

get

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/lbmonitor_sslcertkey_binding/monitorname_value<String>

Query-parameters:

filter

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/lbmonitor_sslcertkey_binding/monitorname_value<String>?**filter=property-name1:property-value1,property-name2:property-value2**

Use this query-parameter to get the filtered set of lbmonitor_sslcertkey_binding resources configured on Citrix ADC. Filtering can be done on any of the properties of the resource.

pagination

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/lbmonitor_sslcertkey_binding/monitorname_value<String>?**pagesize=#no;pageno=#no**

Use this query-parameter to get the lbmonitor_sslcertkey_binding resources in chunks.

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "lbmonitor_sslcertkey_binding": [ {

"certkeyname":<String_value>,

"monitorname":<String_value>,

"ocspcheck":<String_value>,
```

```
"crlcheck":<String_value>,

"ca":<Boolean_value>

}]}
```

get (all)

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/lbmonitor_sslcertkey_binding

Query-parameters:

bulkbindings

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/lbmonitor_sslcertkey_binding?**bulkbindings=yes**

NITRO allows you to fetch bindings in bulk.

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "lbmonitor_sslcertkey_binding": [ {

"certkeyname":<String_value>,

"monitorname":<String_value>,
```

```
"ocspcheck":<String_value>,

"crlcheck":<String_value>,

"ca":<Boolean_value>

}]}
```

count

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/lbmonitor_sslcertkey_binding/monitorname_value<String>?**count=yes**

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{"lbmonitor_sslcertkey_binding": [ { "__count": "#no"} ] }
```

lbparameter

Configuration for LB parameter resource.

Properties

(click to see Operations)

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
httponlycookieflag	<String>	Read-write	Include the HttpOnly attribute in persistence cookies. The HttpOnly attribute limits the scope of a cookie to HTTP requests and helps mitigate the risk of cross-site scripting attacks. Default value: ENABLED Possible values = ENABLED, DISABLED
usesecuredpersistencecookie	<String>	Read-write	Encode persistence cookie values using SHA2 hash. Default value: DISABLED Possible values = ENABLED, DISABLED
useencryptedpersistencecookie	<String>	Read-write	Encode persistence cookie values using SHA2 hash. Default value: DISABLED Possible values = ENABLED, DISABLED
cookiepassphrase	<String>	Read-write	Use this parameter to specify the passphrase used to

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
			generate secured persistence cookie value. It specifies the passphrase with a maximum of 31 characters.
consolidatedlconn	<String>	Read-write	To find the service with the fewest connections, the virtual server uses the consolidated connection statistics from all the packet engines. The NO setting allows consideration of only the number of connections on the packet engine that received the new connection. Default value: YES Possible values = YES, NO
useportforhashlb	<String>	Read-write	Include the port number of the service when creating a hash for hash based load balancing methods. With the NO setting, only the IP address of the service is considered when creating a hash. Default value: YES

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
			Possible values = YES, NO
preferdirectroute	<String>	Read-write	Perform route lookup for traffic received by the Citrix ADC, and forward the traffic according to configured routes. Do not set this parameter if you want a wildcard virtual server to direct packets received by the appliance to an intermediary device, such as a firewall, even if their destination is directly connected to the appliance. Route lookup is performed after the packets have been processed and returned by the intermediary device. Default value: YES Possible values = YES, NO
startuprrfactor	<Double>	Read-write	Number of requests, per service, for which to apply the round robin load balancing method before switching to the configured load balancing method,

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
			thus allowing services to ramp up gradually to full load. Until the specified number of requests is distributed, the Citrix ADC is said to be implementing the slow start mode (or startup round robin). Implemented for a virtual server when one of the following is true: * The virtual server is newly created. * One or more services are newly bound to the virtual server. * One or more services bound to the virtual server are enabled. * The load balancing method is changed. This parameter applies to all the load balancing virtual servers configured on the Citrix ADC, except for those virtual servers for which

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
			the virtual server-level slow start parameters (New Service Startup Request Rate and Increment Interval) are configured. If the global slow start parameter and the slow start parameters for a given virtual server are not set, the appliance implements a default slow start for the virtual server, as follows: * For a newly configured virtual server, the appliance implements slow start for the first 100 requests received by the virtual server. * For an existing virtual server, if one or more services are newly bound or newly enabled, or if the load balancing method is changed, the appliance dynamically

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
			computes the number of requests for which to implement startup round robin. It obtains this number by multiplying the request rate by the number of bound services (it includes services that are marked as DOWN). For example, if the current request rate is 20 requests/s and ten services are bound to the virtual server, the appliance performs startup round robin for 200 requests. Not applicable to a virtual server for which a hash based load balancing method is configured.
monitorskipmaxclient	<String>	Read-write	When a monitor initiates a connection to a service, do not check to determine whether the number of connections to the service has reached the limit specified by the service's Max

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
			Clients setting. Enables monitoring to continue even if the service has reached its connection limit. Default value: DISABLED Possible values = ENABLED, DISABLED
monitorconnectionclose	<String>	Read-write	Close monitoring connections by sending the service a connection termination message with the specified bit set. Default value: FIN Possible values = RESET, FIN
vserverspecificmac	<String>	Read-write	Allow a MAC-mode virtual server to accept traffic returned by an intermediary device, such as a firewall, to which the traffic was previously forwarded by another MAC-mode virtual server. The second virtual server can then distribute that traffic across the destination server farm. Also useful

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
			when load balancing Branch Repeater appliances. Note: The second virtual server can also send the traffic to another set of intermediary devices, such as another set of firewalls. If necessary, you can configure multiple MAC-mode virtual servers to pass traffic successively through multiple sets of intermediary devices. Default value: DISABLED Possible values = ENABLED, DISABLED
allowboundsvcremoval	<String>	Read-write	This is used, to enable/disable the option of svc/svcgroup removal, if it is bound to one or more vserver. If it is enabled, the svc/svcgroup can be removed, even if it bound to vservers. If disabled, an error will be thrown, when the

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
			user tries to remove a svc/svcgroup without unbinding from its vservers. Default value: ENABLED Possible values = ENABLED, DISABLED
retainservicestate	<String>	Read-write	This option is used to retain the original state of service or servicegroup member when an enable server command is issued. Default value: OFF Possible values = ON, OFF
dbsttl	<Double>	Read-write	Specify the TTL for DNS record for domain based service. The default value of ttl is 0 which indicates to use the TTL received in DNS response for monitors. Default value: 0
sessionsthreshold	<Double>	Read-only	This option is used to get the upper-limit on the number of persistent sessions set by the

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
			administrator for this system. Minimum value = 1

Operations

(click to see Properties)

UPDATE| UNSET| GET (ALL)

Some options that you can use for each operations:

- **Getting warnings in response:**NITRO allows you to get warnings in an operation by specifying the "warning" query parameter as "yes". For example, to get warnings while connecting to the Citrix ADC appliance, the URL is as follows:

  http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/login?warning=yes

  If any, the warnings are displayed in the response payload with the HTTP code "209 X-NITRO-WARNING".

- **Authenticated access for individual NITRO operations:**NITRO allows you to logon to the Citrix ADC appliance to perform individual operations. You can use this option instead of creating a NITRO session (using the login object) and then using that session to perform all operations,

  To do this, you must specify the username and password in the request header of the NITRO request as follows:

  X-NITRO-USER:<username>

  X-NITRO-PASS:<password>

  Note: In such cases, make sure that the request header DOES not include the following:

  Cookie:NITRO_AUTH_TOKEN=<tokenvalue>

`*Note:`*

Mandatory parameters are marked in redand placeholder content is marked in <green>.

update

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/lbparameter

HTTP Method:PUT

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Content-Type:application/json

Request Payload:

```
{"lbparameter":{

"httponlycookieflag":<String_value>,

"usesecuredpersistencecookie":<String_value>,

"useencryptedpersistencecookie":<String_value>,

"cookiepassphrase":<String_value>,

"consolidatedlconn":<String_value>,

"useportforhashlb":<String_value>,

"preferdirectroute":<String_value>,

"startuprrfactor":<Double_value>,

"monitorskipmaxclient":<String_value>,

"monitorconnectionclose":<String_value>,

"vserverspecificmac":<String_value>,

"allowboundsvcremoval":<String_value>,

"retainservicestate":<String_value>,

"dbsttl":<Double_value>

}}
```

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error

unset

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/lbparameter?**action=unset**

HTTP Method:POST

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Content-Type:application/json

Request Payload:

```
{"lbparameter":{

"httponlycookieflag":true,

"usesecuredpersistencecookie":true,

"useencryptedpersistencecookie":true,

"cookiepassphrase":true,

"consolidatedlconn":true,

"useportforhashlb":true,

"preferdirectroute":true,

"startuprrfactor":true,

"monitorskipmaxclient":true,

"monitorconnectionclose":true,

"vserverspecificmac":true,
```

```
"allowboundsvcremoval":true,

"retainservicestate":true,

"dbsttl":true

}}
```

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error

get (all)

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/lbparameter

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "lbparameter": [ {

"httponlycookieflag":<String_value>,

"usesecuredpersistencecookie":<String_value>,
```

```
"useencryptedpersistencecookie":<String_value>,

"cookiepassphrase":<String_value>,

"consolidatedlconn":<String_value>,

"useportforhashlb":<String_value>,

"preferdirectroute":<String_value>,

"startuprrfactor":<Double_value>,

"monitorskipmaxclient":<String_value>,

"monitorconnectionclose":<String_value>,

"vserverspecificmac":<String_value>,

"sessionsthreshold":<Double_value>,

"allowboundsvcremoval":<String_value>,

"retainservicestate":<String_value>,

"dbsttl":<Double_value>

}]}
```

lbpersistentsessions

Configuration for persistence session resource.

Properties

(click to see Operations)

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
vserver	<String>	Read-write	The name of the virtual server.

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
nodeid	<Double>	Read-write	Unique number that identifies the cluster node. Minimum value = 0 Maximum value = 31
persistenceparameter	<String>	Read-write	The persistence parameter whose persistence sessions are to be flushed.
type	<Double>	Read-only	Type of Persistence.
typestring	<String>	Read-only	Type of Persistence as String.
srcip	<String>	Read-only	SOURCE IP.
srcipv6	<String>	Read-only	SOURCE IPv6 ADDRESS.
destip	<String>	Read-only	DESTINATION IP.
destipv6	<String>	Read-only	DESTINATION IPv6 ADDRESS.
flags	<Boolean>	Read-only	IPv6 FLAGS.
destport	<Integer>	Read-only	Destination port. Range 1 - 65535 * in CLI is represented as 65535 in NITRO API
vservername	<String>	Read-only	Virtual server name.

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
timeout	<Double>	Read-only	Persistent Session timeout.
referencecount	<Double>	Read-only	Reference Count.
persistenceparam	<String>	Read-only	Specific persistence information . Callid in case of SIP_CALLID persistence entry , RTSP session id in case of RTSP_SESSIONID persistence entry.
cnamepersparam	<String>	Read-only	The cname that is selected incase of gslb service.
__count	<Double>	Read-only	count parameter

Operations

(click to see Properties)

CLEAR| GET (ALL)| COUNT

Some options that you can use for each operations:

- **Getting warnings in response:**NITRO allows you to get warnings in an operation by specifying the "warning" query parameter as "yes". For example, to get warnings while connecting to the Citrix ADC appliance, the URL is as follows:

  http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/login?warning=yes

  If any, the warnings are displayed in the response payload with the HTTP code "209 X-NITRO-WARNING".

- **Authenticated access for individual NITRO operations:**NITRO allows you to logon to the Citrix ADC appliance to perform individual operations. You can use this option instead of creating a NITRO session (using the login object) and then using that session to perform all operations,

To do this, you must specify the username and password in the request header of the NITRO request as follows:

X-NITRO-USER:<username>

X-NITRO-PASS:<password>

Note: In such cases, make sure that the request header DOES not include the following:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue>

`*Note:`*

Mandatory parameters are marked in redand placeholder content is marked in <green>.

clear

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/lbpersistentsessions?**action=clear**

HTTP Method:POST

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Content-Type:application/json

Request Payload:

```
{"lbpersistentsessions":{

"vserver":<String_value>,

"persistenceparameter":<String_value>

}}
```

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error

get (all)

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/lbpersistentsessions

Query-parameters:

args

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/lbpersistentsessions?**args=vserver:<String_value>,nodeid:<Double_value>**

Use this query-parameter to get lbpersistentsessions resources based on additional properties.

attrs

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/lbpersistentsessions?**attrs=property-name1,property-name2**

Use this query parameter to specify the resource details that you want to retrieve.

filter

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/lbpersistentsessions?**filter=property-name1:property-val1,property-name2:property-val2**

Use this query-parameter to get the filtered set of lbpersistentsessions resources configured on Citrix ADC. Filtering can be done on any of the properties of the resource.

view

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/lbpersistentsessions?**view=summary**

Note: By default, the retrieved results are displayed in detail view (?view=detail).

pagination

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/lbpersistentsessions?**pagesize=#no;pageno=#no**

Use this query-parameter to get the lbpersistentsessions resources in chunks.

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "lbpersistentsessions": [ {

vserver:<String_value>,nodeid:<Double_value>"type":<Double_value>,

"typestring":<String_value>,

"srcip":<String_value>,

"srcipv6":<String_value>,

"destip":<String_value>,

"destipv6":<String_value>,

"flags":<Boolean_value>,

"destport":<Integer_value>,

"vservername":<String_value>,

"timeout":<Double_value>,

"referencecount":<Double_value>,

"sipcallid":<String_value>,

"persistenceparam":<String_value>,

"cnamepersparam":<String_value>

}]}
```

count

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/lbpersistentsessions?**count=yes**

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "lbpersistentsessions": [ { "__count": "#no"} ] }
```

lbprofile

Configuration for LB profile resource.

Properties

(click to see Operations)

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
lbprofilename	<String>	Read-write	Name of the LB profile. Minimum length = 1
dbslb	<String>	Read-write	Enable database specific load balancing for MySQL

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
			and MSSQL service types. Default value: DISABLED Possible values = ENABLED, DISABLED
processlocal	<String>	Read-write	By turning on this option packets destined to a vserver in a cluster will not under go any steering. Turn this option for single pa cket request response mode or when the upstream device is performing a proper RSS for connection based distribution. Default value: DISABLED Possible values = ENABLED, DISABLED
httponlycookieflag	<String>	Read-write	Include the HttpOnly attribute in persistence cookies. The HttpOnly attribute limits the scope of a cookie to HTTP requests and helps mitigate the

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
			risk of cross-site scripting attacks. Default value: ENABLED Possible values = ENABLED, DISABLED
cookiepassphrase	<String>	Read-write	Use this parameter to specify the passphrase used to generate secured persistence cookie value. It specifies the passphrase with a maximum of 31 characters.
usesecuredpersistencecookie	<String>	Read-write	Encode persistence cookie values using SHA2 hash. Default value: DISABLED Possible values = ENABLED, DISABLED
useencryptedpersistencecookie	<String>	Read-write	Encode persistence cookie values using SHA2 hash. Default value: DISABLED Possible values = ENABLED, DISABLED

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
vsvrcount	<Double>	Read-only	Total number of vservers , the profile is bound to.
__count	<Double>	Read-only	count parameter

Operations

(click to see Properties)

ADD| DELETE| UPDATE| UNSET| GET (ALL)| GET| COUNT

Some options that you can use for each operations:

- **Getting warnings in response:**NITRO allows you to get warnings in an operation by specifying the "warning" query parameter as "yes". For example, to get warnings while connecting to the Citrix ADC appliance, the URL is as follows:

  http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/login?warning=yes

  If any, the warnings are displayed in the response payload with the HTTP code "209 X-NITRO-WARNING".

- **Authenticated access for individual NITRO operations:**NITRO allows you to logon to the Citrix ADC appliance to perform individual operations. You can use this option instead of creating a NITRO session (using the login object) and then using that session to perform all operations,

  To do this, you must specify the username and password in the request header of the NITRO request as follows:

  X-NITRO-USER:<username>

  X-NITRO-PASS:<password>

  Note: In such cases, make sure that the request header DOES not include the following:

  Cookie:NITRO_AUTH_TOKEN=<tokenvalue>

`*Note:`*

Mandatory parameters are marked in redand placeholder content is marked in <green>.

add

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/lbprofile

HTTP Method:POST

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Content-Type:application/json

Request Payload:

```
{"lbprofile":{

<b>"lbprofilename":<String_value>,

</b>"dbslb":<String_value>,

"processlocal":<String_value>,

"httponlycookieflag":<String_value>,

"cookiepassphrase":<String_value>,

"usesecuredpersistencecookie":<String_value>,

"useencryptedpersistencecookie":<String_value>

}}
```

Response:

HTTP Status Code on Success: 201 Created HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error

delete

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/lbprofile/lbprofilename_value<String>

HTTP Method:DELETE

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue>

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error

update

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/lbprofile

HTTP Method:PUT

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Content-Type:application/json

Request Payload:

```
{"lbprofile":{

<b>"lbprofilename":<String_value>,

</b>"dbslb":<String_value>,

"processlocal":<String_value>,

"httponlycookieflag":<String_value>,

"cookiepassphrase":<String_value>,

"usesecuredpersistencecookie":<String_value>,

"useencryptedpersistencecookie":<String_value>

}}
```

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error

unset

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/lbprofile?**action=unset**

HTTP Method:POST

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Content-Type:application/json

Request Payload:

```
{"lbprofile":{

<b>"lbprofilename":<String_value>,

</b>"dbslb":true,

"processlocal":true,

"httponlycookieflag":true,

"cookiepassphrase":true,

"usesecuredpersistencecookie":true,

"useencryptedpersistencecookie":true

}}
```

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error

get (all)

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/lbprofile

Query-parameters:

attrs

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/lbprofile?**attrs=property-name1,property-name2**

Use this query parameter to specify the resource details that you want to retrieve.

filter

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/lbprofile?**filter=property-name1:property-val1,property-name2:property-val2**

Use this query-parameter to get the filtered set of lbprofile resources configured on Citrix ADC. Filtering can be done on any of the properties of the resource.

view

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/lbprofile?**view=summary**

Note: By default, the retrieved results are displayed in detail view (?view=detail).

pagination

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/lbprofile?**pagesize=#no;pageno=#no**

Use this query-parameter to get the lbprofile resources in chunks.

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "lbprofile": [ {
"lbprofilename":<String_value>,
"dbslb":<String_value>,
"processlocal":<String_value>,
"httponlycookieflag":<String_value>,
"cookiepassphrase":<String_value>,
"usesecuredpersistencecookie":<String_value>,
"useencryptedpersistencecookie":<String_value>,
"vsvrcount":<Double_value>
}]}
```

get

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/lbprofile/lbprofilename_value<String>

Query-parameters:

attrs

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/lbprofile/lbprofilename_value<String>?**attrs=property-name1,property-name2**

Use this query parameter to specify the resource details that you want to retrieve.

view

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/lbprofile/lbprofilename_value<String>?**view=summary**

Note: By default, the retrieved results are displayed in detail view (?view=detail).

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "lbprofile": [ {

"lbprofilename":<String_value>,

"dbslb":<String_value>,

"processlocal":<String_value>,

"httponlycookieflag":<String_value>,

"cookiepassphrase":<String_value>,

"usesecuredpersistencecookie":<String_value>,

"useencryptedpersistencecookie":<String_value>,

"vsvrcount":<Double_value>

}]}
```

count

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/lbprofile?**count=yes**

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "lbprofile": [ { "__count": "#no"} ] }
```

lbroute

Configuration for LB route resource.

Properties

(click to see Operations)

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
network	<String>	Read-write	The IP address of the network to which the route belongs.
netmask	<String>	Read-write	The netmask to which the route belongs.
gatewayname	<String>	Read-write	The name of the route. Minimum length = 1
td	<Double>	Read-write	Integer value that uniquely identifies the traffic domain in

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
			which you want to configure the entity. If you do not specify an ID, the entity becomes part of the default traffic domain, which has an ID of 0. Default value: 0 Minimum value = 0 Maximum value = 4094
flags	<String>	Read-only	State of the configured gateway. Possible values = UP, DOWN, UNKNOWN, BUSY, OUT OF SERVICE, GOING OUT OF SERVICE, DOWN WHEN GOING OUT OF SERVICE, NS_EMPTY_STR, Unknown, DISABLED
__count	<Double>	Read-only	count parameter

Operations

(click to see Properties)

ADD| DELETE| GET (ALL)| COUNT

Some options that you can use for each operations:

- **Getting warnings in response:**NITRO allows you to get warnings in an operation by specifying the "warning" query parameter as "yes". For example, to get warnings while connecting to the Citrix ADC appliance, the URL is as follows:

  http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/login?warning=yes

  If any, the warnings are displayed in the response payload with the HTTP code "209 X-NITRO-WARNING".

- **Authenticated access for individual NITRO operations:**NITRO allows you to logon to the Citrix ADC appliance to perform individual operations. You can use this option instead of creating a NITRO session (using the login object) and then using that session to perform all operations,

  To do this, you must specify the username and password in the request header of the NITRO request as follows:

  X-NITRO-USER:<username>

  X-NITRO-PASS:<password>

  Note: In such cases, make sure that the request header DOES not include the following:

  Cookie:NITRO_AUTH_TOKEN=<tokenvalue>

`*Note:`*

Mandatory parameters are marked in redand placeholder content is marked in <green>.

add

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/lbroute

HTTP Method:POST

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Content-Type:application/json

Request Payload:

```
{"lbroute":{
```

```
<b>"network":<String_value>,

</b><b>"netmask":<String_value>,

</b><b>"gatewayname":<String_value>,

</b>"td":<Double_value>

}}
```

Response:

HTTP Status Code on Success: 201 Created HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error

delete

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/lbroute/network_value<String>

HTTP Method:DELETE

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue>

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error

get (all)

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/lbroute

Query-parameters:

attrs

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/lbroute?**attrs=property-name1,property-name2**

Use this query parameter to specify the resource details that you want to retrieve.

filter

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/lbroute?**filter=property-name1:property-val1,property-name2:property-val2**

Use this query-parameter to get the filtered set of lbroute resources configured on Citrix ADC. Filtering can be done on any of the properties of the resource.

view

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/lbroute?**view=summary**

Note: By default, the retrieved results are displayed in detail view (?view=detail).

pagination

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/lbroute?**pagesize=#no;pageno=#no**

Use this query-parameter to get the lbroute resources in chunks.

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "lbroute": [ {

"network":<String_value>,

"netmask":<String_value>,

"td":<Double_value>,
```

```
"gatewayname":<String_value>,

"flags":<String_value>

}]}
```

count

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/lbroute?**count=yes**

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "lbroute": [ { "__count": "#no"} ] }
```

lbroute6

Configuration for LB route6 resource.

Properties

(click to see Operations)

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
network	<String>	Read-write	The destination network.
gatewayname	<String>	Read-write	The name of the route. Minimum length = 1
td	<Double>	Read-write	Integer value that uniquely identifies the traffic domain in which you want to configure the entity. If you do not specify an ID, the entity becomes part of the default traffic domain, which has an ID of 0. Default value: 0 Minimum value = 0 Maximum value = 4094
flags	<String>	Read-only	State of the configured gateway. Possible values = UP, DOWN, UNKNOWN, BUSY, OUT OF SERVICE, GOING OUT OF SERVICE, DOWN WHEN GOING OUT OF SERVICE, NS_EMPTY_STR, Unknown, DISABLED

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
__count	<Double>	Read-only	count parameter

Operations

(click to see Properties)

ADD| DELETE| GET (ALL)| COUNT

Some options that you can use for each operations:

- **Getting warnings in response:**NITRO allows you to get warnings in an operation by specifying the "warning" query parameter as "yes". For example, to get warnings while connecting to the Citrix ADC appliance, the URL is as follows:

  http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/login?warning=yes

  If any, the warnings are displayed in the response payload with the HTTP code "209 X-NITRO-WARNING".
- **Authenticated access for individual NITRO operations:**NITRO allows you to logon to the Citrix ADC appliance to perform individual operations. You can use this option instead of creating a NITRO session (using the login object) and then using that session to perform all operations,

  To do this, you must specify the username and password in the request header of the NITRO request as follows:

  X-NITRO-USER:<username>

  X-NITRO-PASS:<password>

  Note: In such cases, make sure that the request header DOES not include the following:

  Cookie:NITRO_AUTH_TOKEN=<tokenvalue>

`*Note:`*

Mandatory parameters are marked in redand placeholder content is marked in <green>.

add

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/lbroute6

HTTP Method:POST

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Content-Type:application/json

Request Payload:

```
{"lbroute6":{

<b>"network":<String_value>,

</b><b>"gatewayname":<String_value>,

</b>"td":<Double_value>

}}
```

Response:

HTTP Status Code on Success: 201 Created HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error

delete

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/lbroute6/network_value<String>

HTTP Method:DELETE

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue>

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error

get (all)

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/lbroute6

Query-parameters:

attrs

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/lbroute6?**attrs=property-name1,property-name2**

Use this query parameter to specify the resource details that you want to retrieve.

filter

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/lbroute6?**filter=property-name1:property-val1,property-name2:property-val2**

Use this query-parameter to get the filtered set of lbroute6 resources configured on Citrix ADC. Filtering can be done on any of the properties of the resource.

view

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/lbroute6?**view=summary**

Note: By default, the retrieved results are displayed in detail view (?view=detail).

pagination

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/lbroute6?**pagesize=#no;pageno=#no**

Use this query-parameter to get the lbroute6 resources in chunks.

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "lbroute6": [ {

"network":<String_value>,

"td":<Double_value>,

"gatewayname":<String_value>,

"flags":<String_value>

}]}
```

count

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/lbroute6?**count=yes**

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "lbroute6": [ { "__count": "#no"} ] }
```

lbsipparameters

Configuration for SIP parameters resource.

Properties

(click to see Operations)

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
rnatsrcport	<Integer>	Read-write	Port number with which to match the source port in server-initiated SIP traffic. The rport parameter is added, without a value, to SIP packets that have a matching source port number, and CALL-ID based persistence is implemented for the responses received by the virtual server. Default value: 0
rnatdstport	<Integer>	Read-write	Port number with which to match the destination port in server-initiated SIP traffic. The rport parameter is added, without a value, to SIP packets that have a matching destination port number, and CALL-ID based persistence is implemented for the responses received by the virtual server. Default value: 0
retrydur	<Integer>	Read-write	Time, in seconds, for which a client must wait before initiating

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
			a connection after receiving a 503 Service Unavailable response from the SIP server. The time value is sent in the "Retry-After" header in the 503 response. Default value: 120 Minimum value = 1
addrportvip	<String>	Read-write	Add the rport parameter to the VIA headers of SIP requests that virtual servers receive from clients or servers. Default value: ENABLED Possible values = ENABLED, DISABLED
sip503ratethreshold	<Double>	Read-write	Maximum number of 503 Service Unavailable responses to generate, once every 10 milliseconds, when a SIP virtual server becomes unavailable. Default value: 100
rnatsecuresrcport	<Integer>	Read-write	Port number with which to match the source port in server-initiated SIP over SSL traffic. The rport

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
			parameter is added, without a value, to SIP packets that have a matching source port number, and CALL-ID based persistence is implemented for the responses received by the virtual server. Default value: 0 Range 1 - 65535 * in CLI is represented as 65535 in NITRO API
rnatsecuredstport	<Integer>	Read-write	Port number with which to match the destination port in server-initiated SIP over SSL traffic. The rport parameter is added, without a value, to SIP packets that have a matching destination port number, and CALL-ID based persistence is implemented for the responses received by the virtual server. Default value: 0 Range 1 - 65535 * in CLI is represented as

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
			65535 in NITRO API

Operations

(click to see Properties)

UPDATE| UNSET| GET (ALL)

Some options that you can use for each operations:

- **Getting warnings in response:**NITRO allows you to get warnings in an operation by specifying the "warning" query parameter as "yes". For example, to get warnings while connecting to the Citrix ADC appliance, the URL is as follows:

  http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/login?warning=yes

  If any, the warnings are displayed in the response payload with the HTTP code "209 X-NITRO-WARNING".

- **Authenticated access for individual NITRO operations:**NITRO allows you to logon to the Citrix ADC appliance to perform individual operations. You can use this option instead of creating a NITRO session (using the login object) and then using that session to perform all operations,

  To do this, you must specify the username and password in the request header of the NITRO request as follows:

  X-NITRO-USER:<username>

  X-NITRO-PASS:<password>

  Note: In such cases, make sure that the request header DOES not include the following:

  Cookie:NITRO_AUTH_TOKEN=<tokenvalue>

`*Note:`*

Mandatory parameters are marked in redand placeholder content is marked in <green>.

update

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/lbsipparameters

HTTP Method:PUT

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Content-Type:application/json

Request Payload:

```
{"lbsipparameters":{

"rnatsrcport":<Integer_value>,

"rnatdstport":<Integer_value>,

"retrydur":<Integer_value>,

"addrportvip":<String_value>,

"sip503ratethreshold":<Double_value>,

"rnatsecuresrcport":<Integer_value>,

"rnatsecuredstport":<Integer_value>

}}
```

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error

unset

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/lbsipparameters?**action=unset**

HTTP Method:POST

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Content-Type:application/json

Request Payload:

```
{"lbsipparameters":{

"rnatsrcport":true,

"rnatdstport":true,

"retrydur":true,

"addrportvip":true,

"sip503ratethreshold":true,

"rnatsecuresrcport":true,

"rnatsecuredstport":true

}}
```

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error

get (all)

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/lbsipparameters

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "lbsipparameters": [ {

"rnatsrcport":<Integer_value>,

"rnatdstport":<Integer_value>,

"retrydur":<Integer_value>,

"addrportvip":<String_value>,

"sip503ratethreshold":<Double_value>,

"rnatsecuresrcport":<Integer_value>,

"rnatsecuredstport":<Integer_value>

}]}
```

lbvserver

Configuration for Load Balancing Virtual Server resource.

Properties

(click to see Operations)

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
name	<String>	Read-write	Name for the virtual server. Must begin with an ASCII alphanumeric or underscore (_) character, and must contain only ASCII alphanumeric,

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
			underscore, hash (#), period (.), space, colon (:), at sign (@), equal sign (=), and hyphen (-) characters. Can be changed after the virtual server is created. CLI Users: If the name includes one or more spaces, enclose the name in double or single quotation marks (for example, "my vserver" or 'my vserver'). . Minimum length = 1
servicetype	<String>	Read-write	Protocol used by the service (also called the service type). Possible values = HTTP, FTP, TCP, UDP, SSL, SSL_BRIDGE, SSL_TCP, DTLS, NNTP, DNS, DHCPRA, ANY, SIP_UDP, SIP_TCP, SIP_SSL, DNS_TCP, RTSP, PUSH, SSL_PUSH, RADIUS, RDP, MYSQL, MSSQL, DIAMETER,

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
			SSL_DIAMETER, TFTP, ORACLE, SMPP, SYSLOGTCP, SYSLOGUDP, FIX, SSL_FIX, PROXY, USER_TCP, USER_SSL_TCP, QUIC, IPFIX, LOGSTREAM
ipv46	<String>	Read-write	IPv4 or IPv6 address to assign to the virtual server.
ippattern	<String>	Read-write	IP address pattern, in dotted decimal notation, for identifying packets to be accepted by the virtual server. The IP Mask parameter specifies which part of the destination IP address is matched against the pattern. Mutually exclusive with the IP Address parameter. For example, if the IP pattern assigned to the virtual server is 198.51.100.0 and the IP mask is 255.255.240.0 (a forward mask), the first 20 bits in the destination IP

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
			addresses are matched with the first 20 bits in the pattern. The virtual server accepts requests with IP addresses that range from 198.51.96.1 to 198.51.111.254. You can also use a pattern such as 0.0.2.2 and a mask such as 0.0.255.255 (a reverse mask). If a destination IP address matches more than one IP pattern, the pattern with the longest match is selected, and the associated virtual server processes the request. For example, if virtual servers vs1 and vs2 have the same IP pattern, 0.0.100.128, but different IP masks of 0.0.255.255 and 0.0.224.255, a destination IP address of

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
			198.51.100.128 has the longest match with the IP pattern of vs1. If a destination IP address matches two or more virtual servers to the same extent, the request is processed by the virtual server whose port number matches the port number in the request.
ipmask	<String>	Read-write	IP mask, in dotted decimal notation, for the IP Pattern parameter. Can have leading or trailing non-zero octets (for example, 255.255.240.0 or 0.0.255.255). Accordingly, the mask specifies whether the first n bits or the last n bits of the destination IP address in a client request are to be matched with the corresponding bits in the IP pattern. The former is called a forward mask. The

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
			latter is called a reverse mask.
port	<Integer>	Read-write	Port number for the virtual server. Range 1 - 65535 * in CLI is represented as 65535 in NITRO API
ipset	<String>	Read-write	The list of IPv4/IPv6 addresses bound to ipset would form a part of listening service on the current lb vserver. Minimum length = 1
range	<Double>	Read-write	Number of IP addresses that the appliance must generate and assign to the virtual server. The virtual server then functions as a network virtual server, accepting traffic on any of the generated IP addresses. The IP addresses are generated automatically, as follows: * For a range of n, the last octet of the address specified by the IP Address

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
			parameter increments n-1 times. * If the last octet exceeds 255, it rolls over to 0 and the third octet increments by 1. Note: The Range parameter assigns multiple IP addresses to one virtual server. To generate an array of virtual servers, each of which owns only one IP address, use brackets in the IP Address and Name parameters to specify the range. For example: add lb vserver my_vserver[1-3] HTTP 192.0.2.[1-3] 80. Default value: 1 Minimum value = 1 Maximum value = 254

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
persistencetype	<String>	Read-write	Type of persistence for the virtual server. Available settings function as follows: * SOURCEIP - Connections from the same client IP address belong to the same persistence session. * COOKIEINSERT - Connections that have the same HTTP Cookie, inserted by a Set-Cookie directive from a server, belong to the same persistence session. * SSLSESSION - Connections that have the same SSL Session ID belong to the same persistence session. * CUSTOMSERVERID - Connections with the same server ID form part of the same

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
			session. For this persistence type, set the Server ID (CustomServerID) parameter for each service and configure the Rule parameter to identify the server ID in a request. * RULE - All connections that match a user defined rule belong to the same persistence session. * URLPASSIVE - Requests that have the same server ID in the URL query belong to the same persistence session. The server ID is the hexadecimal representation of the IP address and port of the service to which the request must be forwarded. This persistence type requires a rule

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
			to identify the server ID in the request. * DESTIP - Connections to the same destination IP address belong to the same persistence session. * SRCIPDESTIP - Connections that have the same source IP address and destination IP address belong to the same persistence session. * CALLID - Connections that have the same CALL-ID SIP header belong to the same persistence session. * RTSPSID - Connections that have the same RTSP Session ID belong to the same

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
			persistence session. * FIXSESSION - Connections that have the same SenderCompID and TargetCompID values belong to the same persistence session. * USERSESSION - Persistence session is created based on the persistence parameter value provided from an extension. Possible values = SOURCEIP, COOKIEINSERT, SSLSESSION, RULE, URLPASSIVE, CUSTOMSERVERID, DESTIP, SRCIPDESTIP, CALLID, RTSPSID, DIAMETER, FIXSESSION, USERSESSION, NONE

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
timeout	<Double>	Read-write	Time period for which a persistence session is in effect. Default value: 2 Minimum value = 0 Maximum value = 1440
persistencebackup	<String>	Read-write	Backup persistence type for the virtual server. Becomes operational if the primary persistence mechanism fails. Possible values = SOURCEIP, NONE
backuppersistencetimeout	<Double>	Read-write	Time period for which backup persistence is in effect. Default value: 2 Minimum value = 2 Maximum value = 1440
lbmethod	<String>	Read-write	Load balancing method. The available settings function as follows: * ROUNDROBIN - Distribute requests in rotation, regardless of the load. Weights can be assigned to

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
			services to enforce weighted round robin distribution. * LEASTCONNECTION (default) - Select the service with the fewest connections. * LEASTRESPONSETIME - Select the service with the lowest average response time. * LEASTBANDWIDTH - Select the service currently handling the least traffic. * LEASTPACKETS - Select the service currently serving the lowest number of packets per second. * CUSTOMLOAD - Base service selection on the SNMP metrics obtained by custom load monitors.

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
			* LRTM - Select the service with the lowest response time. Response times are learned through monitoring probes. This method also takes the number of active connections into account. Also available are a number of hashing methods, in which the appliance extracts a predetermined portion of the request, creates a hash of the portion, and then checks whether any previous requests had the same hash value. If it finds a match, it forwards the request to the service that served those previous requests. Following are the hashing methods: * URLHASH - Create a hash of

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
			the request URL (or part of the URL). * DOMAINHASH - Create a hash of the domain name in the request (or part of the domain name). The domain name is taken from either the URL or the Host header. If the domain name appears in both locations, the URL is preferred. If the request does not contain a domain name, the load balancing method defaults to LEASTCONNECTION. * DESTINATIONIPHASH - Create a hash of the destination IP address in the IP header. * SOURCEIPHASH - Create a hash of the source IP address in the IP header.

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
			* TOKEN - Extract a token from the request, create a hash of the token, and then select the service to which any previous requests with the same token hash value were sent. * SRCIPDESTIPHASH - Create a hash of the string obtained by concatenating the source IP address and destination IP address in the IP header. * SRCIPSRCPORTHASH - Create a hash of the source IP address and source port in the IP header. * CALLIDHASH - Create a hash of the SIP Call-ID header.

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
			* USER_TOKEN - Same as TOKEN LB method but token needs to be provided from an extension. Default value: LEASTCONNECTION Possible values = ROUNDROBIN, LEASTCONNECTION, LEASTRESPONSETIME, URLHASH, DOMAINHASH, DESTINATIONIPHASH, SOURCEIPHASH, SRCIPDESTIPHASH, LEASTBANDWIDTH, LEASTPACKETS, TOKEN, SRCIPSRCPORTHASH, LRTM, CALLIDHASH, CUSTOMLOAD, LEASTREQUEST, AUDITLOGHASH, STATICPROXIMITY, USER_TOKEN

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
hashlength	<Double>	Read-write	Number of bytes to consider for the hash value used in the URLHASH and DOMAINHASH load balancing methods. Default value: 80 Minimum value = 1 Maximum value = 4096
netmask	<String>	Read-write	IPv4 subnet mask to apply to the destination IP address or source IP address when the load balancing method is DESTINATIONIPHASH or SOURCEIPHASH. Minimum length = 1
v6netmasklen	<Double>	Read-write	Number of bits to consider in an IPv6 destination or source IP address, for creating the hash that is required by the DESTINATIONIPHASH and SOURCEIPHASH load balancing methods. Default value: 128 Minimum value = 1

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
			Maximum value = 128
backuplbmethod	<String>	Read-write	Backup load balancing method. Becomes operational if the primary load balancing me thod fails or cannot be used. Valid only if the primary method is based on static proximity. Default value: ROUNDROBIN Possible values = ROUNDROBIN, LEASTCONNECTI ON, LEASTRESPONSE TIME, SOURCEIPHASH, LEASTBANDWIDT H, LEASTPACKETS, CUSTOMLOAD
cookiename	<String>	Read-write	Use this parameter to specify the cookie name for COOKIE peristence type. It specifies the name of cookie with a maximum of 32 characters. If not

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
			specified, cookie name is internally generated.
rule	<String>	Read-write	Expression, or name of a named expression, against which traffic is evaluated. The following requirements apply only to the Citrix ADC CLI: * If the expression includes one or more spaces, enclose the entire expression in double quotation marks. * If the expression itself includes double quotation marks, escape the quotations by using the \ character. * Alternatively, you can use single quotation marks to enclose the rule, in which case you do not have to escape the double quotation marks.

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
			Default value: "none"
listenpolicy	<String>	Read-write	Expression identifying traffic accepted by the virtual server. Can be either an expression (for example, CLIENT.IP.DST.IN_SUBNET(192.0.2.0/24) or the name of a named expression. In the above example, the virtual server accepts all requests whose destination IP address is in the 192.0.2.0/24 subnet. Default value: "NONE"
listenpriority	<Double>	Read-write	Integer specifying the priority of the listen policy. A higher number specifies a lower priority. If a request matches the listen policies of more than one virtual server the virtual server whose listen policy has the highest priority (the lowest priority number) accepts the request. Default value: 101

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
			Minimum value = 0 Maximum value = 101
resrule	<String>	Read-write	Expression specifying which part of a server's response to use for creating rule based persistence sessions (persistence type RULE). Can be either an expression or the name of a named expression. Example: HTTP.RES.HEADER("setcookie").VALUE(0).TYPECAST_NVLIST_T('=',';').VALUE("server1"). Default value: "none"
persistmask	<String>	Read-write	Persistence mask for IP based persistence types, for IPv4 virtual servers. Minimum length = 1
v6persistmasklen	<Double>	Read-write	Persistence mask for IP based persistence types, for IPv6 virtual servers. Default value: 128 Minimum value = 1

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
			Maximum value = 128
pq	<String>	Read-write	Use priority queuing on the virtual server. based persistence types, for IPv6 virtual servers. Default value: OFF Possible values = ON, OFF
sc	<String>	Read-write	Use SureConnect on the virtual server. Default value: OFF Possible values = ON, OFF
rtspnat	<String>	Read-write	Use network address translation (NAT) for RTSP data connections. Default value: OFF Possible values = ON, OFF
m	<String>	Read-write	Redirection mode for load balancing. Available settings function as follows: * IP - Before forwarding a request to a server, change the destination IP

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
			address to the server's IP address. * MAC - Before forwarding a request to a server, change the destination MAC address to the server's MAC address. The destination IP address is not changed. MAC-based redirection mode is used mostly in firewall load balancing deployments. * IPTUNNEL - Perform IP-in-IP encapsulation for client IP packets. In the outer IP headers, set the destination IP address to the IP address of the server and the source IP address to the subnet IP (SNIP). The client IP packets are not modified. Applicable

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
			to both IPv4 and IPv6 packets. * TOS - Encode the virtual server's TOS ID in the TOS field of the IP header. You can use either the IPTUNNEL or the TOS option to implement Direct Server Return (DSR). Default value: IP Possible values = IP, MAC, IPTUNNEL, TOS
tosid	<Double>	Read-write	TOS ID of the virtual server. Applicable only when the load balancing redirection mode is set to TOS. Minimum value = 1 Maximum value = 63
datalength	<Double>	Read-write	Length of the token to be extracted from the data segment of an incoming packet, for use in the token method of load balancing. The length of the token, specified in bytes,

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
			must not be greater than 24 KB. Applicable to virtual servers of type TCP. Minimum value = 1 Maximum value = 100
dataoffset	<Double>	Read-write	Offset to be considered when extracting a token from the TCP payload. Applicable to virtual servers, of type TCP, using the token method of load balancing. Must be within the first 24 KB of the TCP payload. Minimum value = 0 Maximum value = 25400
sessionless	<String>	Read-write	Perform load balancing on a per-packet basis, without establishing sessions. Recommended for load balancing of intrusion detection system (IDS) servers and scenarios involving direct server return (DSR), where session information is unnecessary.

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
			Default value: DISABLED Possible values = ENABLED, DISABLED
trofspersistence	<String>	Read-write	When value is ENABLED, Trofs persistence is honored. When value is DISABLED, Trofs persistence is not honored. Default value: ENABLED Possible values = ENABLED, DISABLED
state	<String>	Read-write	State of the load balancing virtual server. Default value: ENABLED Possible values = ENABLED, DISABLED
connfailover	<String>	Read-write	Mode in which the connection failover feature must operate for the virtual server. After a failover, established TCP connections and UDP packet flows are kept active and

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
			resumed on the secondary appliance. Clients remain connected to the same servers. Available settings function as follows: * STATEFUL - The primary appliance shares state information with the secondary appliance, in real time, resulting in some runtime processing overhead. * STATELESS - State information is not shared, and the new primary appliance tries to re-create the packet flow on the basis of the information contained in the packets it receives. * DISABLED - Connection failover does not occur. Default value: DISABLED Possible values = DISABLED,

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
			STATEFUL, STATELESS
redirurl	<String>	Read-write	URL to which to redirect traffic if the virtual server becomes unavailable. WARNING! Make sure that the domain in the URL does not match the domain specified for a content switching policy. If it does, requests are continuously redirected to the unavailable virtual server. Minimum length = 1
cacheable	<String>	Read-write	Route cacheable requests to a cache redirection virtual server. The load balancing virtual server can forward requests only to a transparent cache redirection virtual server that has an IP address and port combination of *:80, so such a cache redirection virtual server must be

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
			configured on the appliance. Default value: NO Possible values = YES, NO
clttimeout	<Double>	Read-write	Idle time, in seconds, after which a client connection is terminated. Minimum value = 0 Maximum value = 31536000
somethod	<String>	Read-write	Type of threshold that, when exceeded, triggers spillover. Available settings function as follows: * CONNECTION - Spillover occurs when the number of client connections exceeds the threshold. * DYNAMICCONNECTION - Spillover occurs when the number of client connections at the virtual server exceeds the sum of the maximum client (Max Clients)

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
			settings for bound services. Do not specify a spillover threshold for this setting, because the threshold is implied by the Max Clients settings of bound services. * BANDWIDTH - Spillover occurs when the bandwidth consumed by the virtual server's incoming and outgoing traffic exceeds the threshold. * HEALTH - Spillover occurs when the percentage of weights of the services that are UP drops below the threshold. For example, if services svc1, svc2, and svc3 are bound to a virtual server, with weights 1, 2, and 3, and the spillover threshold is 50%, spillover occurs if svc1 and svc3 or

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
			svc2 and svc3 transition to DOWN. * NONE - Spillover does not occur. Possible values = CONNECTION, DYNAMICCONNECTION, BANDWIDTH, HEALTH, NONE
sopersistence	<String>	Read-write	If spillover occurs, maintain source IP address based persistence for both primary and backup virtual servers. Default value: DISABLED Possible values = ENABLED, DISABLED
sopersistencetimeout	<Double>	Read-write	Timeout for spillover persistence, in minutes. Default value: 2 Minimum value = 2 Maximum value = 1440
healththreshold	<Double>	Read-write	Threshold in percent of active services below which vserver state is made down.

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
			If this threshold is 0, vserver state will be up even if one bound service is up. Default value: 0 Minimum value = 0 Maximum value = 100
sothreshold	<Double>	Read-write	Threshold at which spillover occurs. Specify an integer for the CONNECTION spillover method, a bandwidth value in kilobits per second for the BANDWIDTH method (do not enter the units), or a percentage for the HEALTH method (do not enter the percentage symbol). Minimum value = 1 Maximum value = 4294967287
sobackupaction	<String>	Read-write	Action to be performed if spillover is to take effect, but no backup chain to spillover is usable or exists. Possible values = DROP, ACCEPT, REDIRECT

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
redirectportrewrite	<String>	Read-write	Rewrite the port and change the protocol to ensure successful HTTP redirects from services. Default value: DISABLED Possible values = ENABLED, DISABLED
downstateflush	<String>	Read-write	Flush all active transactions associated with a virtual server whose state transitions from UP to DOWN. Do not enable this option for applications that must complete their transactions. Default value: ENABLED Possible values = ENABLED, DISABLED
backupvserver	<String>	Read-write	Name of the backup virtual server to which to forward requests if the primary virtual server goes DOWN or reaches its spillover threshold. Minimum length = 1

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
disableprimaryondown	<String>	Read-write	If the primary virtual server goes down, do not allow it to return to primary status until manually enabled. Default value: DISABLED Possible values = ENABLED, DISABLED
insertvserveripport	<String>	Read-write	Insert an HTTP header, whose value is the IP address and port number of the virtual server, before forwarding a request to the server. The format of the header is <vipHeader>: <virtual server IP address>_<port number >, where vipHeader is the name that you specify for the header. If the virtual server has an IPv6 address, the address in the header is enclosed in brackets ([and]) to separate it from the port number. If you have mapped an IPv4 address to a virtual server's IPv6 address, the value of

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
			this parameter determines which IP address is inserted in the header, as follows: * VIPADDR - Insert the IP address of the virtual server in the HTTP header regardless of whether the virtual server has an IPv4 address or an IPv6 address. A mapped IPv4 address, if configured, is ignored. * V6TOV4MAPPING - Insert the IPv4 address that is mapped to the virtual server's IPv6 address. If a mapped IPv4 address is not configured, insert the IPv6 address. * OFF - Disable header insertion. Possible values = OFF, VIPADDR, V6TOV4MAPPING

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
vipheader	<String>	Read-write	Name for the inserted header. The default name is vip-header. Minimum length = 1
authenticationhost	<String>	Read-write	Fully qualified domain name (FQDN) of the authentication virtual server to which the user must be redirected for authentication. Make sure that the Authentication parameter is set to ENABLED. Minimum length = 3 Maximum length = 252
authentication	<String>	Read-write	Enable or disable user authentication. Default value: OFF Possible values = ON, OFF
authn401	<String>	Read-write	Enable or disable user authentication with HTTP 401 responses. Default value: OFF Possible values = ON, OFF

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
authnvsname	<String>	Read-write	Name of an authentication virtual server with which to authenticate users. Minimum length = 1 Maximum length = 252
push	<String>	Read-write	Process traffic with the push virtual server that is bound to this load balancing virtual server. Default value: DISABLED Possible values = ENABLED, DISABLED
pushvserver	<String>	Read-write	Name of the load balancing virtual server, of type PUSH or SSL_PUSH, to which the server pushes updates received on the load balancing virtual server that you are configuring. Minimum length = 1
pushlabel	<String>	Read-write	Expression for extracting a label from the server's response. Can be either an expression or the name of a named expression.

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
			Default value: "none"
pushmulticlients	<String>	Read-write	Allow multiple Web 2.0 connections from the same client to connect to the virtual server and expect updates. Default value: NO Possible values = YES, NO
tcpprofilename	<String>	Read-write	Name of the TCP profile whose settings are to be applied to the virtual server. Minimum length = 1 Maximum length = 127
httpprofilename	<String>	Read-write	Name of the HTTP profile whose settings are to be applied to the virtual server. Minimum length = 1 Maximum length = 127
dbprofilename	<String>	Read-write	Name of the DB profile whose settings are to be applied to the virtual server. Minimum length = 1

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
			Maximum length = 127
comment	<String>	Read-write	Any comments that you might want to associate with the virtual server.
l2conn	<String>	Read-write	Use Layer 2 parameters (channel number, MAC address, and VLAN ID) in addition to the 4-tuple (<source IP>:<source port>::<destination IP>:<destination port>) that is used to identify a connection. Allows multiple TCP and non-TCP connections with the same 4-tuple to co-exist on the Citrix ADC. Possible values = ON, OFF
oracleserverversion	<String>	Read-write	Oracle server version. Default value: 10G Possible values = 10G, 11G
mssqlserverversion	<String>	Read-write	For a load balancing virtual server of type MSSQL, the Microsoft SQL Server version. Set this parameter if you

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
			expect some clients to run a version different from the version of the database. This setting provides compatibility between the client-side and server-side connections by ensuring that all communication conforms to the server's version. Default value: 2008R2 Possible values = 70, 2000, 2000SP1, 2005, 2008, 2008R2, 2012, 2014
mysqlprotocolversion	<Double>	Read-write	MySQL protocol version that the virtual server advertises to clients.
mysqlserverversion	<String>	Read-write	MySQL server version string that the virtual server advertises to clients. Minimum length = 1 Maximum length = 31
mysqlcharacterset	<Double>	Read-write	Character set that the virtual server advertises to clients.

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
mysqlservercapabilities	<Double>	Read-write	Server capabilities that the virtual server advertises to clients.
appflowlog	<String>	Read-write	Apply AppFlow logging to the virtual server. Default value: ENABLED Possible values = ENABLED, DISABLED
netprofile	<String>	Read-write	Name of the network profile to associate with the virtual server. If you set this parameter, the virtual server uses only the IP addresses in the network profile as source IP addresses when initiating connections with servers. Minimum length = 1 Maximum length = 127
icmpvsrresponse	<String>	Read-write	How the Citrix ADC responds to ping requests received for an IP address that is common to one or more virtual servers. Available settings function as follows:

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
			* If set to PASSIVE on all the virtual servers that share the IP address, the appliance always responds to the ping requests. * If set to ACTIVE on all the virtual servers that share the IP address, the appliance responds to the ping requests if at least one of the virtual servers is UP. Otherwise, the appliance does not respond. * If set to ACTIVE on some virtual servers and PASSIVE on the others, the appliance responds if at least one virtual server with the ACTIVE setting is UP. Otherwise, the appliance does not respond. Note: This parameter is available at the virtual server level.

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
			A similar parameter, ICMP Response, is available at the IP address level, for IPv4 addresses of type VIP. To set that parameter, use the add ip command in the CLI or the Create IP dialog box in the GUI. Default value: PASSIVE Possible values = PASSIVE, ACTIVE
rhistate	<String>	Read-write	Route Health Injection (RHI) functionality of the NetSaler appliance for advertising the route of the VIP address associated with the virtual server. When Vserver RHI Level (RHI) parameter is set to VSVR_CNTRLD, the following are different RHI behaviors for the VIP address on the basis of RHIstate (RHI STATE) settings on the virtual servers

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
			associated with the VIP address: * If you set RHI STATE to PASSIVE on all virtual servers, the Citrix ADC always advertises the route for the VIP address. * If you set RHI STATE to ACTIVE on all virtual servers, the Citrix ADC advertises the route for the VIP address if at least one of the associated virtual servers is in UP state. * If you set RHI STATE to ACTIVE on some and PASSIVE on others, the Citrix ADC advertises the route for the VIP address if at least one of the associated virtual servers, whose RHI STATE set to ACTIVE, is in UP state.

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
			Default value: PASSIVE Possible values = PASSIVE, ACTIVE
newservicerequest	<Double>	Read-write	Number of requests, or percentage of the load on existing services, by which to increase the load on a new service at each interval in slow-start mode. A non-zero value indicates that slow-start is applicable. A zero value indicates that the global RR startup parameter is applied. Changing the value to zero will cause services currently in slow start to take the full traffic as determined by the LB method. Subsequently, any new services added will use the global RR factor. Default value: 0
newservicerequestunit	<String>	Read-write	Units in which to increment load at each interval in slow-start mode. Default value: PER_SECOND

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
			Possible values = PER_SECOND, PERCENT
newservicerequestincrementinterval	<Double>	Read-write	Interval, in seconds, between successive increments in the load on a new service or a service whose state has just changed from DOWN to UP. A value of 0 (zero) specifies manual slow start. Default value: 0 Minimum value = 0 Maximum value = 3600
minautoscalemembers	<Double>	Read-write	Minimum number of members expected to be present when vserver is used in Autoscale. Default value: 0 Minimum value = 0 Maximum value = 5000
maxautoscalemembers	<Double>	Read-write	Maximum number of members expected to be present when vserver is used in Autoscale. Default value: 0

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
			Minimum value = 0 Maximum value = 5000
persistavpno	<Double[]>	Read-write	Persist AVP number for Diameter Persistency. In case this AVP is not defined in Base RFC 3588 and it is nested inside a Grouped AVP, define a sequence of AVP numbers (max 3) in order of parent to child. So say persist AVP number X is nested inside AVP Y which is nested in Z, then define the list as Z Y X. Minimum value = 1
skippersistency	<String>	Read-write	This argument decides the behavior incase the service which is selected from an existing persistence session has reached threshold. Default value: None

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
			Possible values = Bypass, ReLb, None
td	<Double>	Read-write	Integer value that uniquely identifies the traffic domain in which you want to configure the entity. If you do not specify an ID, the entity becomes part of the default traffic domain, which has an ID of 0. Minimum value = 0 Maximum value = 4094
authnprofile	<String>	Read-write	Name of the authentication profile to be used when authentication is turned on.
macmoderetainvlan	<String>	Read-write	This option is used to retain vlan information of incoming packet when macmode is enabled. Default value: DISABLED Possible values = ENABLED, DISABLED

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
dbslb	<String>	Read-write	Enable database specific load balancing for MySQL and MSSQL service types. Default value: DISABLED Possible values = ENABLED, DISABLED
dns64	<String>	Read-write	This argument is for enabling/disabling the dns64 on lbvserver. Possible values = ENABLED, DISABLED
bypassaaaa	<String>	Read-write	If this option is enabled while resolving DNS64 query AAAA queries are not sent to back end dns server. Default value: NO Possible values = YES, NO
recursionavailable	<String>	Read-write	When set to YES, this option causes the DNS replies from this vserver to have the RA bit turned on. Typically one would set this option to YES, when the vserver is load

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
			balancing a set of DNS servers thatsupport recursive queries. Default value: NO Possible values = YES, NO
processlocal	<String>	Read-write	By turning on this option packets destined to a vserver in a cluster will not under go any steering. Turn this option for single packet request response mode or when the upstream device is performing a proper RSS for connection based distribution. Default value: DISABLED Possible values = ENABLED, DISABLED
dnsprofilename	<String>	Read-write	Name of the DNS profile to be associated with the VServer. DNS profile properties will be applied to the transactions processed by a VServer. This parameter is valid

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
			only for DNS and DNS-TCP VServers. Minimum length = 1 Maximum length = 127
lbprofilename	<String>	Read-write	Name of the LB profile which is associated to the vserver.
redirectfromport	<Integer>	Read-write	Port number for the virtual server, from which we absorb the traffic for http redirect. Minimum value = 1 Range 1 - 65535 * in CLI is represented as 65535 in NITRO API
httpsredirecturl	<String>	Read-write	URL to which to redirect traffic if the traffic is recieved from redirect port.
retainconnectionsoncluster	<String>	Read-write	This option enables you to retain existing connections on a node joining a Cluster system or when a node is being configured for passive timeout. By default, this option is disabled.

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
			Default value: NO Possible values = YES, NO
adfsproxyprofile	<String>	Read-write	Name of the adfsProxy profile to be used to support ADFSPIP protocol for ADFS servers.
weight	<Double>	Read-write	Weight to assign to the specified service. Minimum value = 1 Maximum value = 100
servicename	<String>	Read-write	Service to bind to the virtual server. Minimum length = 1
redirurlflags	<Boolean>	Read-write	The redirect URL to be unset.
newname	<String>	Read-write	New name for the virtual server. Minimum length = 1
value	<String>	Read-only	SSL status. Possible values = Certkey not bound, SSL feature disabled
ipmapping	<String>	Read-only	The permanent mapping for the V6 Address.

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
ngname	<String>	Read-only	Nodegroup name to which this lbvsever belongs to.
type	<String>	Read-only	Type of LB vserver. Possible values = CONTENT, ADDRESS
curstate	<String>	Read-only	Current LB vserver state. Possible values = UP, DOWN, UNKNOWN, BUSY, OUT OF SERVICE, GOING OUT OF SERVICE, DOWN WHEN GOING OUT OF SERVICE, NS_EMPTY_STR, Unknown, DISABLED
effectivestate	<String>	Read-only	Effective state of the LB vserver , based on the state of backup vservers. Possible values = UP, DOWN, UNKNOWN, BUSY, OUT OF SERVICE, GOING OUT OF SERVICE, DOWN WHEN GOING OUT OF SERVICE, NS_EMPTY_STR,

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
			Unknown, DISABLED
status	<Integer>	Read-only	Current status of the lb vserver. During the initial phase if the configured lb method is not round robin , the vserver will adopt round robin to distribute traffic for a predefined number of requests.
lbrrreason	<Integer>	Read-only	Reason why a vserver is in RR. The following are the reasons: 1 - MEP is DOWN (GSLB) 2 - LB method has changed 3 - Bound service's state changed to UP 4 - A new service is bound 5 - Startup RR factor has changed 6 - LB feature is enabled 7 - Load monitor is not active on a service

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
			8 - Vserver is Enabled 9 - SSL feature is Enabled 10 - All bound services have reached threshold. Using effective state to load balance (GSLB) 11 - Primary state of bound services are not UP. Using effective state to load balance (GSLB) 12 - No LB decision can be made as all bound services have either reached threshold or are not UP (GSLB) 13 - All load monitors are active .
redirect	<String>	Read-only	Cache redirect type. Possible values = CACHE, POLICY, ORIGIN
precedence	<String>	Read-only	Precedence.

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
			Possible values = RULE, URL
homepage	<String>	Read-only	Home page.
dnsvservername	<String>	Read-only	DNS vserver name.
domain	<String>	Read-only	Domain.
policyname	<String>	Read-only	Name of the policy bound to the LB vserver.
cachevserver	<String>	Read-only	Cache virtual server.
health	<Double>	Read-only	Health of vserver based on percentage of weights of active svcs/all svcs. This does not consider administratively disabled svcs.
gotopriorityexpression	<String>	Read-only	Expression specifying the priority of the next policy which will get evaluated if the current policy rule evaluates to TRUE.
ruletype	<Double>	Read-only	Rule type.
groupname	<String>	Read-only	LB group to which the lb vserver is to be bound.
cookiedomain	<String>	Read-only	Domain name to be used in the set cookie header in

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
			case of cookie persistence.
map	<String>	Read-only	Map. Possible values = ON, OFF
gt2gb	<String>	Read-only	Allow for greater than 2 GB transactions on this vserver. Default value: DISABLED Possible values = ENABLED, DISABLED
consolidatedlconn	<String>	Read-only	Use consolidated stats for LeastConnection. Default value: GLOBAL Possible values = GLOBAL, NO, YES
consolidatedlconngbl	<String>	Read-only	Fetches Global setting. Possible values = YES, NO
thresholdvalue	<Integer>	Read-only	Tells whether threshold exceeded for this service participating in CUSTOMLB.

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
bindpoint	<String>	Read-only	The bindpoint to which the policy is bound. Possible values = REQUEST, RESPONSE
invoke	<Boolean>	Read-only	Invoke policies bound to a virtual server or policy label.
labeltype	<String>	Read-only	The invocation type. Possible values = reqvserver, resvserver, policylabel
labelname	<String>	Read-only	Name of the label invoked.
version	<Integer>	Read-only	Cookie version.
totalservices	<Double>	Read-only	Total number of services bound to the vserver.
activeservices	<Double>	Read-only	Total number of active services bound to the vserver.
statechangetimesec	<String>	Read-only	Time when last state change happened. Seconds part.
statechangetimeseconds	<Double>	Read-only	Time when last state change happened. Seconds part.
statechangetimemsec	<Double>	Read-only	Time at which last state change

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
			happened. Milliseconds part.
tickssincelaststatechange	<Double>	Read-only	Time in 10 millisecond ticks since the last state change.
isgslb	<Boolean>	Read-only	This field is set to true if it is a GSLBVserver.
vsvrdynconnsothreshold	<Double>	Read-only	Spillover threshold for dynamic connection.
backupvserverstatus	<String>	Read-only	Staus of BackUp Vserver .
nodefaultbindings	<String>	Read-only	to determine if the configuration will have default ssl CIPHER and ECC curve bindings. Default value: NO Possible values = YES, NO
__count	<Double>	Read-only	count parameter

Operations

(click to see Properties)

ADD| DELETE| UPDATE| UNSET| ENABLE| DISABLE| RENAME| GET (ALL)| GET| COUNT

Some options that you can use for each operations:

- **Getting warnings in response:**NITRO allows you to get warnings in an operation by specifying the "warning" query parameter as "yes". For example, to get warnings while connecting to the Citrix ADC appliance, the URL is as follows:

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/login?warning=yes

If any, the warnings are displayed in the response payload with the HTTP code "209 X-NITRO-WARNING".

- **Authenticated access for individual NITRO operations:**NITRO allows you to logon to the Citrix ADC appliance to perform individual operations. You can use this option instead of creating a NITRO session (using the login object) and then using that session to perform all operations,

  To do this, you must specify the username and password in the request header of the NITRO request as follows:

  X-NITRO-USER:<username>

  X-NITRO-PASS:<password>

  Note: In such cases, make sure that the request header DOES not include the following:

  Cookie:NITRO_AUTH_TOKEN=<tokenvalue>

`*Note:`*

Mandatory parameters are marked in redand placeholder content is marked in <green>.

add

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/lbvserver

HTTP Method:POST

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Content-Type:application/json

Request Payload:

```
{"lbvserver":{

<b>"name":<String_value>,

</b><b>"servicetype":<String_value>,

</b>"ipv46":<String_value>,
```

```
"ippattern":<String_value>,

"ipmask":<String_value>,

"port":<Integer_value>,

"ipset":<String_value>,

"range":<Double_value>,

"persistencetype":<String_value>,

"timeout":<Double_value>,

"persistencebackup":<String_value>,

"backuppersistencetimeout":<Double_value>,

"lbmethod":<String_value>,

"hashlength":<Double_value>,

"netmask":<String_value>,

"v6netmasklen":<Double_value>,

"backuplbmethod":<String_value>,

"cookiename":<String_value>,

"rule":<String_value>,

"listenpolicy":<String_value>,

"listenpriority":<Double_value>,

"resrule":<String_value>,

"persistmask":<String_value>,

"v6persistmasklen":<Double_value>,

"pq":<String_value>,
```

```
"sc":<String_value>,

"rtspnat":<String_value>,

"m":<String_value>,

"tosid":<Double_value>,

"datalength":<Double_value>,

"dataoffset":<Double_value>,

"sessionless":<String_value>,

"trofspersistence":<String_value>,

"state":<String_value>,

"connfailover":<String_value>,

"redirurl":<String_value>,

"cacheable":<String_value>,

"clttimeout":<Double_value>,

"somethod":<String_value>,

"sopersistence":<String_value>,

"sopersistencetimeout":<Double_value>,

"healththreshold":<Double_value>,

"sothreshold":<Double_value>,

"sobackupaction":<String_value>,

"redirectportrewrite":<String_value>,

"downstateflush":<String_value>,

"backupvserver":<String_value>,
```

```
"disableprimaryondown":<String_value>,

"insertvserveripport":<String_value>,

"vipheader":<String_value>,

"authenticationhost":<String_value>,

"authentication":<String_value>,

"authn401":<String_value>,

"authnvsname":<String_value>,

"push":<String_value>,

"pushvserver":<String_value>,

"pushlabel":<String_value>,

"pushmulticlients":<String_value>,

"tcpprofilename":<String_value>,

"httpprofilename":<String_value>,

"dbprofilename":<String_value>,

"comment":<String_value>,

"l2conn":<String_value>,

"oracleserverversion":<String_value>,

"mssqlserverversion":<String_value>,

"mysqlprotocolversion":<Double_value>,

"mysqlserverversion":<String_value>,

"mysqlcharacterset":<Double_value>,

"mysqlservercapabilities":<Double_value>,
```

```
"appflowlog":<String_value>,

"netprofile":<String_value>,

"icmpvsrresponse":<String_value>,

"rhistate":<String_value>,

"newservicerequest":<Double_value>,

"newservicerequestunit":<String_value>,

"newservicerequestincrementinterval":<Double_value>,

"minautoscalemembers":<Double_value>,

"maxautoscalemembers":<Double_value>,

"persistavpno":<Double[]_value>,

"skippersistency":<String_value>,

"td":<Double_value>,

"authnprofile":<String_value>,

"macmoderetainvlan":<String_value>,

"dbslb":<String_value>,

"dns64":<String_value>,

"bypassaaaa":<String_value>,

"recursionavailable":<String_value>,

"processlocal":<String_value>,

"dnsprofilename":<String_value>,

"lbprofilename":<String_value>,

"redirectfromport":<Integer_value>,
```

```
"httpsredirecturl":<String_value>,

"retainconnectionsoncluster":<String_value>,

"adfsproxyprofile":<String_value>

}}
```

Response:

HTTP Status Code on Success: 201 Created HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error

delete

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/lbvserver/name_value<String>

HTTP Method:DELETE

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue>

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error

update

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/lbvserver

HTTP Method:PUT

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Content-Type:application/json

Request Payload:

```
{"lbvserver":{
<b>"name":<String_value>,
</b>"ipv46":<String_value>,
"ipset":<String_value>,
"ippattern":<String_value>,
"ipmask":<String_value>,
"weight":<Double_value>,
"servicename":<String_value>,
"persistencetype":<String_value>,
"timeout":<Double_value>,
"persistencebackup":<String_value>,
"backuppersistencetimeout":<Double_value>,
"lbmethod":<String_value>,
"hashlength":<Double_value>,
"netmask":<String_value>,
"v6netmasklen":<Double_value>,
"backuplbmethod":<String_value>,
"rule":<String_value>,
"cookiename":<String_value>,
"resrule":<String_value>,
"persistmask":<String_value>,
```

```
"v6persistmasklen":<Double_value>,

"pq":<String_value>,

"sc":<String_value>,

"rtspnat":<String_value>,

"m":<String_value>,

"tosid":<Double_value>,

"datalength":<Double_value>,

"dataoffset":<Double_value>,

"sessionless":<String_value>,

"trofspersistence":<String_value>,

"connfailover":<String_value>,

"backupvserver":<String_value>,

"redirurl":<String_value>,

"cacheable":<String_value>,

"clttimeout":<Double_value>,

"somethod":<String_value>,

"sothreshold":<Double_value>,

"sopersistence":<String_value>,

"sopersistencetimeout":<Double_value>,

"healththreshold":<Double_value>,

"sobackupaction":<String_value>,

"redirectportrewrite":<String_value>,
```

```
"downstateflush":<String_value>,

"insertvserveripport":<String_value>,

"vipheader":<String_value>,

"disableprimaryondown":<String_value>,

"authenticationhost":<String_value>,

"authentication":<String_value>,

"authn401":<String_value>,

"authnvsname":<String_value>,

"push":<String_value>,

"pushvserver":<String_value>,

"pushlabel":<String_value>,

"pushmulticlients":<String_value>,

"listenpolicy":<String_value>,

"listenpriority":<Double_value>,

"tcpprofilename":<String_value>,

"httpprofilename":<String_value>,

"dbprofilename":<String_value>,

"comment":<String_value>,

"l2conn":<String_value>,

"oracleserverversion":<String_value>,

"mssqlserverversion":<String_value>,

"mysqlprotocolversion":<Double_value>,
```

```
"mysqlserverversion":<String_value>,

"mysqlcharacterset":<Double_value>,

"mysqlservercapabilities":<Double_value>,

"appflowlog":<String_value>,

"netprofile":<String_value>,

"icmpvsrresponse":<String_value>,

"rhistate":<String_value>,

"newservicerequest":<Double_value>,

"newservicerequestunit":<String_value>,

"newservicerequestincrementinterval":<Double_value>,

"minautoscalemembers":<Double_value>,

"maxautoscalemembers":<Double_value>,

"persistavpno":<Double[]_value>,

"skippersistency":<String_value>,

"authnprofile":<String_value>,

"macmoderetainvlan":<String_value>,

"dbslb":<String_value>,

"dns64":<String_value>,

"bypassaaaa":<String_value>,

"recursionavailable":<String_value>,

"processlocal":<String_value>,

"dnsprofilename":<String_value>,
```

```
"lbprofilename":<String_value>,

"redirectfromport":<Integer_value>,

"httpsredirecturl":<String_value>,

"retainconnectionsoncluster":<String_value>,

"adfsproxyprofile":<String_value>

}}
```

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error

unset

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/lbvserver?**action=unset**

HTTP Method:POST

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Content-Type:application/json

Request Payload:

```
{"lbvserver":{

<b>"name":<String_value>,

</b>"backupvserver":true,

"clttimeout":true,

"redirurl":true,
```

```
"redirurlflags":true,

"authn401":true,

"authentication":true,

"authenticationhost":true,

"authnvsname":true,

"pushvserver":true,

"pushlabel":true,

"tcpprofilename":true,

"httpprofilename":true,

"dbprofilename":true,

"rule":true,

"sothreshold":true,

"l2conn":true,

"mysqlprotocolversion":true,

"mysqlserverversion":true,

"mysqlcharacterset":true,

"mysqlservercapabilities":true,

"appflowlog":true,

"netprofile":true,

"icmpvsrresponse":true,

"skippersistency":true,

"minautoscalemembers":true,
```

```
"maxautoscalemembers":true,

"authnprofile":true,

"macmoderetainvlan":true,

"dbslb":true,

"dnsprofilename":true,

"lbprofilename":true,

"redirectfromport":true,

"httpsredirecturl":true,

"adfsproxyprofile":true,

"ipset":true,

"servicename":true,

"persistencetype":true,

"timeout":true,

"persistencebackup":true,

"backuppersistencetimeout":true,

"lbmethod":true,

"hashlength":true,

"netmask":true,

"v6netmasklen":true,

"backuplbmethod":true,

"cookiename":true,

"resrule":true,
```

```
"persistmask":true,

"v6persistmasklen":true,

"pq":true,

"sc":true,

"rtspnat":true,

"m":true,

"tosid":true,

"datalength":true,

"dataoffset":true,

"sessionless":true,

"trofspersistence":true,

"connfailover":true,

"cacheable":true,

"somethod":true,

"sopersistence":true,

"sopersistencetimeout":true,

"healththreshold":true,

"sobackupaction":true,

"redirectportrewrite":true,

"downstateflush":true,

"insertvserveripport":true,

"vipheader":true,
```

```
"disableprimaryondown":true,

"push":true,

"pushmulticlients":true,

"listenpolicy":true,

"listenpriority":true,

"comment":true,

"oracleserverversion":true,

"mssqlserverversion":true,

"rhistate":true,

"newservicerequest":true,

"newservicerequestunit":true,

"newservicerequestincrementinterval":true,

"persistavpno":true,

"recursionavailable":true,

"retainconnectionsoncluster":true

}}
```


Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error

enable

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/lbvserver?**action=enable**

HTTP Method:POST

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Content-Type:application/json

Request Payload:

```
{"lbvserver":{

<b>"name":<String_value>

</b>}}
```

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error

disable

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/lbvserver?**action=disable**

HTTP Method:POST

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Content-Type:application/json

Request Payload:

```
{"lbvserver":{

<b>"name":<String_value>

</b>}}
```

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error

rename

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/lbvserver?**action=rename**

HTTP Method:POST

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Content-Type:application/json

Request Payload:

```
{"lbvserver":{

<b>"name":<String_value>,

</b><b>"newname":<String_value>

</b>}}
```

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error

get (all)

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/lbvserver

Query-parameters:

attrs

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/lbvserver?**attrs=property-name1,property-name2**

Use this query parameter to specify the resource details that you want to retrieve.

filter

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/lbvserver?**filter=property-name1:property-val1,property-name2:property-val2**

Use this query-parameter to get the filtered set of lbvserver resources configured on Citrix ADC. Filtering can be done on any of the properties of the resource.

view

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/lbvserver?**view=summary**

Note: By default, the retrieved results are displayed in detail view (?view=detail).

pagination

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/lbvserver?**pagesize=#no;pageno=#no**

Use this query-parameter to get the lbvserver resources in chunks.

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "lbvserver": [ {

"name":<String_value>,

"insertvserveripport":<String_value>,
```

```
"vipheader":<String_value>,

"value":<String_value>,

"ip":<String_value>,

"ipv46":<String_value>,

"ippattern":<String_value>,

"ipmask":<String_value>,

"listenpolicy":<String_value>,

"listenpriority":<Double_value>,

"ipmapping":<String_value>,

"port":<Integer_value>,

"ipset":<String_value>,

"range":<Double_value>,

"servicetype":<String_value>,

"ngname":<String_value>,

"type":<String_value>,

"curstate":<String_value>,

"effectivestate":<String_value>,

"status":<Integer_value>,

"lbrrreason":<Integer_value>,

"redirect":<String_value>,

"precedence":<String_value>,

"redirurl":<String_value>,
```

```
"authentication":<String_value>,

"authn401":<String_value>,

"authnvsname":<String_value>,

"homepage":<String_value>,

"dnsvservername":<String_value>,

"domain":<String_value>,

"policyname":<String_value>,

"servicename":<String_value>,

"weight":<Double_value>,

"cachevserver":<String_value>,

"backupvserver":<String_value>,

"clttimeout":<Double_value>,

"somethod":<String_value>,

"sopersistence":<String_value>,

"sopersistencetimeout":<Double_value>,

"healththreshold":<Double_value>,

"sothreshold":<Double_value>,

"sobackupaction":<String_value>,

"lbmethod":<String_value>,

"backuplbmethod":<String_value>,

"hashlength":<Double_value>,

"dataoffset":<Double_value>,
```

```
"health":<Double_value>,

"datalength":<Double_value>,

"netmask":<String_value>,

"v6netmasklen":<Double_value>,

"rule":<String_value>,

"resrule":<String_value>,

"gotopriorityexpression":<String_value>,

"ruletype":<Double_value>,

"groupname":<String_value>,

"m":<String_value>,

"tosid":<Double_value>,

"persistencetype":<String_value>,

"timeout":<Double_value>,

"cookiedomain":<String_value>,

"persistmask":<String_value>,

"v6persistmasklen":<Double_value>,

"persistencebackup":<String_value>,

"backuppersistencetimeout":<Double_value>,

"cacheable":<String_value>,

"pq":<String_value>,

"sc":<String_value>,

"rtspnat":<String_value>,
```

```
"sessionless":<String_value>,

"trofspersistence":<String_value>,

"map":<String_value>,

"connfailover":<String_value>,

"redirectportrewrite":<String_value>,

"downstateflush":<String_value>,

"disableprimaryondown":<String_value>,

"gt2gb":<String_value>,

"consolidatedlconn":<String_value>,

"consolidatedlconngbl":<String_value>,

"thresholdvalue":<Integer_value>,

"bindpoint":<String_value>,

"invoke":<Boolean_value>,

"labeltype":<String_value>,

"labelname":<String_value>,

"cookiename":<String_value>,

"version":<Integer_value>,

"totalservices":<Double_value>,

"activeservices":<Double_value>,

"statechangetimesec":<String_value>,

"statechangetimeseconds":<Double_value>,

"statechangetimemsec":<Double_value>,
```

```
"tickssincelaststatechange":<Double_value>,

"authenticationhost":<String_value>,

"push":<String_value>,

"pushvserver":<String_value>,

"pushlabel":<String_value>,

"pushmulticlients":<String_value>,

"tcpprofilename":<String_value>,

"httpprofilename":<String_value>,

"dbprofilename":<String_value>,

"comment":<String_value>,

"l2conn":<String_value>,

"oracleserverversion":<String_value>,

"mssqlserverversion":<String_value>,

"mysqlprotocolversion":<Double_value>,

"mysqlserverversion":<String_value>,

"mysqlcharacterset":<Double_value>,

"mysqlservercapabilities":<Double_value>,

"appflowlog":<String_value>,

"netprofile":<String_value>,

"isgslb":<Boolean_value>,

"icmpvsrresponse":<String_value>,

"rhistate":<String_value>,
```

```
"newservicerequest":<Double_value>,

"newservicerequestunit":<String_value>,

"newservicerequestincrementinterval":<Double_value>,

"persistavpno":<Double[]_value>,

"skippersistency":<String_value>,

"td":<Double_value>,

"minautoscalemembers":<Double_value>,

"maxautoscalemembers":<Double_value>,

"authnprofile":<String_value>,

"macmoderetainvlan":<String_value>,

"dbslb":<String_value>,

"dns64":<String_value>,

"bypassaaaa":<String_value>,

"recursionavailable":<String_value>,

"processlocal":<String_value>,

"vsvrdynconnsothreshold":<Double_value>,

"backupvserverstatus":<String_value>,

"dnsprofilename":<String_value>,

"lbprofilename":<String_value>,

"redirectfromport":<Integer_value>,

"httpsredirecturl":<String_value>,

"retainconnectionsoncluster":<String_value>,
```

```
"nodefaultbindings":<String_value>,

"adfsproxyprofile":<String_value>

}]}
```

get

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/lbvserver/name_value<String>

Query-parameters:

attrs

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/lbvserver/name_value<String>?**attrs=property-name1,property-name2**

Use this query parameter to specify the resource details that you want to retrieve.

view

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/lbvserver/name_value<String>?**view=summary**

Note: By default, the retrieved results are displayed in detail view (?view=detail).

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "lbvserver": [ {

"name":<String_value>,

"insertvserveripport":<String_value>,

"vipheader":<String_value>,

"value":<String_value>,

"ip":<String_value>,

"ipv46":<String_value>,

"ippattern":<String_value>,

"ipmask":<String_value>,

"listenpolicy":<String_value>,

"listenpriority":<Double_value>,

"ipmapping":<String_value>,

"port":<Integer_value>,

"ipset":<String_value>,

"range":<Double_value>,

"servicetype":<String_value>,

"ngname":<String_value>,

"type":<String_value>,

"curstate":<String_value>,

"effectivestate":<String_value>,

"status":<Integer_value>,
```

```
"lbrrreason":<Integer_value>,

"redirect":<String_value>,

"precedence":<String_value>,

"redirurl":<String_value>,

"authentication":<String_value>,

"authn401":<String_value>,

"authnvsname":<String_value>,

"homepage":<String_value>,

"dnsvservername":<String_value>,

"domain":<String_value>,

"policyname":<String_value>,

"servicename":<String_value>,

"weight":<Double_value>,

"cachevserver":<String_value>,

"backupvserver":<String_value>,

"clttimeout":<Double_value>,

"somethod":<String_value>,

"sopersistence":<String_value>,

"sopersistencetimeout":<Double_value>,

"healththreshold":<Double_value>,

"sothreshold":<Double_value>,

"sobackupaction":<String_value>,
```

```
"lbmethod":<String_value>,

"backuplbmethod":<String_value>,

"hashlength":<Double_value>,

"dataoffset":<Double_value>,

"health":<Double_value>,

"datalength":<Double_value>,

"netmask":<String_value>,

"v6netmasklen":<Double_value>,

"rule":<String_value>,

"resrule":<String_value>,

"gotopriorityexpression":<String_value>,

"ruletype":<Double_value>,

"groupname":<String_value>,

"m":<String_value>,

"tosid":<Double_value>,

"persistencetype":<String_value>,

"timeout":<Double_value>,

"cookiedomain":<String_value>,

"persistmask":<String_value>,

"v6persistmasklen":<Double_value>,

"persistencebackup":<String_value>,

"backuppersistencetimeout":<Double_value>,
```

```
"cacheable":<String_value>,

"pq":<String_value>,

"sc":<String_value>,

"rtspnat":<String_value>,

"sessionless":<String_value>,

"trofspersistence":<String_value>,

"map":<String_value>,

"connfailover":<String_value>,

"redirectportrewrite":<String_value>,

"downstateflush":<String_value>,

"disableprimaryondown":<String_value>,

"gt2gb":<String_value>,

"consolidatedlconn":<String_value>,

"consolidatedlconngbl":<String_value>,

"thresholdvalue":<Integer_value>,

"bindpoint":<String_value>,

"invoke":<Boolean_value>,

"labeltype":<String_value>,

"labelname":<String_value>,

"cookiename":<String_value>,

"version":<Integer_value>,

"totalservices":<Double_value>,
```

```
"activeservices":<Double_value>,

"statechangetimesec":<String_value>,

"statechangetimeseconds":<Double_value>,

"statechangetimemsec":<Double_value>,

"tickssincelaststatechange":<Double_value>,

"authenticationhost":<String_value>,

"push":<String_value>,

"pushvserver":<String_value>,

"pushlabel":<String_value>,

"pushmulticlients":<String_value>,

"tcpprofilename":<String_value>,

"httpprofilename":<String_value>,

"dbprofilename":<String_value>,

"comment":<String_value>,

"l2conn":<String_value>,

"oracleserverversion":<String_value>,

"mssqlserverversion":<String_value>,

"mysqlprotocolversion":<Double_value>,

"mysqlserverversion":<String_value>,

"mysqlcharacterset":<Double_value>,

"mysqlservercapabilities":<Double_value>,

"appflowlog":<String_value>,
```

```
"netprofile":<String_value>,

"isgslb":<Boolean_value>,

"icmpvsrresponse":<String_value>,

"rhistate":<String_value>,

"newservicerequest":<Double_value>,

"newservicerequestunit":<String_value>,

"newservicerequestincrementinterval":<Double_value>,

"persistavpno":<Double[]_value>,

"skippersistency":<String_value>,

"td":<Double_value>,

"minautoscalemembers":<Double_value>,

"maxautoscalemembers":<Double_value>,

"authnprofile":<String_value>,

"macmoderetainvlan":<String_value>,

"dbslb":<String_value>,

"dns64":<String_value>,

"bypassaaaa":<String_value>,

"recursionavailable":<String_value>,

"processlocal":<String_value>,

"vsvrdynconnsothreshold":<Double_value>,

"backupvserverstatus":<String_value>,

"dnsprofilename":<String_value>,
```

```
"lbprofilename":<String_value>,

"redirectfromport":<Integer_value>,

"httpsredirecturl":<String_value>,

"retainconnectionsoncluster":<String_value>,

"nodefaultbindings":<String_value>,

"adfsproxyprofile":<String_value>

}]}
```

count

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/lbvserver?**count=yes**

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "lbvserver": [ { "__count": "#no"} ] }
```

lbvserver_analyticsprofile_binding

Binding object showing the analyticsprofile that can be bound to lbvserver.

Properties

(click to see Operations)

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
name	<String>	Read-write	Name for the virtual server. Must begin with an ASCII alphanumeric or underscore (_) character, and must contain only ASCII alphanumeric, underscore, hash (#), period (.), space, colon (:), at sign (@), equal sign (=), and hyphen (-) characters. Can be changed after the virtual server is created. CLI Users: If the name includes one or more spaces, enclose the name in double or single quotation marks (for example, "my vserver" or 'my vserver'). . Minimum length = 1
analyticsprofile	<String>	Read-write	Name of the analytics profile bound to the LB vserver.
__count	<Double>	Read-write	count parameter

Operations

(click to see Properties)

ADD:| DELETE:| GET| GET (ALL)| COUNT

Some options that you can use for each operations:

- **Getting warnings in response:**NITRO allows you to get warnings in an operation by specifying the "warning" query parameter as "yes". For example, to get warnings while connecting to the Citrix ADC appliance, the URL is as follows:

  http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/login?warning=yes

  If any, the warnings are displayed in the response payload with the HTTP code "209 X-NITRO-WARNING".

- **Authenticated access for individual NITRO operations:**NITRO allows you to logon to the Citrix ADC appliance to perform individual operations. You can use this option instead of creating a NITRO session (using the login object) and then using that session to perform all operations,

  To do this, you must specify the username and password in the request header of the NITRO request as follows:

  X-NITRO-USER:<username>

  X-NITRO-PASS:<password>

  Note: In such cases, make sure that the request header DOES not include the following:

  Cookie:NITRO_AUTH_TOKEN=<tokenvalue>

`*Note:`*

Mandatory parameters are marked in redand placeholder content is marked in <green>.

add

URL:http://<Citrix-ADC-IP-address(NSIP)/nitro/v1/config/lbvserver_analyticsprofile_binding

HTTP Method:PUT

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Content-Type:application/json

Request Payload:

```
{

"lbvserver_analyticsprofile_binding":{

<b>"name":<String_value>,

</b>"analyticsprofile":<String_value>

}}
```

Response:

HTTP Status Code on Success: 201 Created HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error

delete

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/lbvserver_analyticsprofile_binding/name_value<String>

Query-parameters:

args

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/lbvserver_analyticsprofile_binding/name_value<String>?**args=analyticsprofile:<String_value>**

HTTP Method:DELETE

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue>

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error

get

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/lbvserver_analyticsprofile_binding/name_value<String>

Query-parameters:

filter

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/lbvserver_analyticsprofile_binding/name_value<String>?**filter=property-name1:property-value1,property-name2:property-value2**

Use this query-parameter to get the filtered set of lbvserver_analyticsprofile_binding resources configured on Citrix ADC. Filtering can be done on any of the properties of the resource.

pagination

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/lbvserver_analyticsprofile_binding/name_value<String>?**pagesize=#no;pageno=#no**

Use this query-parameter to get the lbvserver_analyticsprofile_binding resources in chunks.

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "lbvserver_analyticsprofile_binding": [ {

"name":<String_value>,

"analyticsprofile":<String_value>
```

```
}]}
```

get (all)

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/lbvserver_analyticsprofile_binding

Query-parameters:

bulkbindings

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/lbvserver_analyticsprofile_binding?**bulkbindings=yes**

NITRO allows you to fetch bindings in bulk.

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "lbvserver_analyticsprofile_binding": [ {

"name":<String_value>,

"analyticsprofile":<String_value>

}]}
```

count

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/lbvserver_analyticsprofile_binding/name_value<String>?**count=yes**

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{"lbvserver_analyticsprofile_binding": [ { "__count": "#no"} ] }
```

lbvserver_appflowpolicy_binding

Binding object showing the appflowpolicy that can be bound to lbvserver.

Properties

(click to see Operations)

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
priority	<Double>	Read-write	Priority.
bindpoint	<String>	Read-write	The bindpoint to which the policy is bound. Possible values = REQUEST, RESPONSE

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
policyname	<String>	Read-write	Name of the policy bound to the LB vserver.
labelname	<String>	Read-write	Name of the label invoked.
name	<String>	Read-write	Name for the virtual server. Must begin with an ASCII alphanumeric or underscore (_) character, and must contain only ASCII alphanumeric, underscore, hash (#), period (.), space, colon (:), at sign (@), equal sign (=), and hyphen (-) characters. Can be changed after the virtual server is created. CLI Users: If the name includes one or more spaces, enclose the name in double or single quotation marks (for example, "my vserver" or 'my vserver'). . Minimum length = 1
gotopriorityexpression	<String>	Read-write	Expression specifying the priority of the next policy which will get evaluated if the

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
			current policy rule evaluates to TRUE.
invoke	<Boolean>	Read-write	Invoke policies bound to a virtual server or policy label.
labeltype	<String>	Read-write	The invocation type. Possible values = reqvserver, resvserver, policylabel
__count	<Double>	Read-write	count parameter

Operations

(click to see Properties)

ADD:| DELETE:| GET| GET (ALL)| COUNT

Some options that you can use for each operations:

- **Getting warnings in response:**NITRO allows you to get warnings in an operation by specifying the "warning" query parameter as "yes". For example, to get warnings while connecting to the Citrix ADC appliance, the URL is as follows:

  http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/login?warning=yes

  If any, the warnings are displayed in the response payload with the HTTP code "209 X-NITRO-WARNING".

- **Authenticated access for individual NITRO operations:**NITRO allows you to logon to the Citrix ADC appliance to perform individual operations. You can use this option instead of creating a NITRO session (using the login object) and then using that session to perform all operations,

  To do this, you must specify the username and password in the request header of the NITRO request as follows:

  X-NITRO-USER:<username>

X-NITRO-PASS:<password>

Note: In such cases, make sure that the request header DOES not include the following:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue>

`*Note:`*

Mandatory parameters are marked in redand placeholder content is marked in <green>.

add

URL:http://<Citrix-ADC-IP-address(NSIP)/nitro/v1/config/lbvserver_appflowpolicy_binding

HTTP Method:PUT

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Content-Type:application/json

Request Payload:

```
{
"lbvserver_appflowpolicy_binding":{
<b>"name":<String_value>,
</b>"policyname":<String_value>,
"priority":<Double_value>,
"gotopriorityexpression":<String_value>,
"bindpoint":<String_value>,
"invoke":<Boolean_value>,
"labeltype":<String_value>,
"labelname":<String_value>
}}
```

Response:

HTTP Status Code on Success: 201 Created HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error

delete

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/lbvserver_appflowpolicy_binding/name_value<String>

Query-parameters:

args

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/lbvserver_appflowpolicy_binding/name_value<String>?**args=policyname:<String_value>,bindpoint:<String_value>,priority:<Double_value>**

HTTP Method:DELETE

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue>

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error

get

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/lbvserver_appflowpolicy_binding/name_value<String>

Query-parameters:

filter

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/lbvserver_appflowpolicy_binding/name_value<String>?**filter=property-name1:property-value1,property-name2:property-value2**

Use this query-parameter to get the filtered set of lbvserver_appflowpolicy_binding resources configured on Citrix ADC. Filtering can be done on any of the properties of the resource.

pagination

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/lbvserver_appflowpolicy_binding/name_value<String>?**pagesize=#no;pageno=#no**

Use this query-parameter to get the lbvserver_appflowpolicy_binding resources in chunks.

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "lbvserver_appflowpolicy_binding": [ {

"priority":<Double_value>,

"bindpoint":<String_value>,

"policyname":<String_value>,

"labelname":<String_value>,

"name":<String_value>,

"gotopriorityexpression":<String_value>,

"invoke":<Boolean_value>,
```

```
"labeltype":<String_value>

}]}
```

get (all)

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/lbvserver_appflowpolicy_binding

Query-parameters:

bulkbindings

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/lbvserver_appflowpolicy_binding?**bulkbindings=yes**

NITRO allows you to fetch bindings in bulk.

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "lbvserver_appflowpolicy_binding": [ {

"priority":<Double_value>,

"bindpoint":<String_value>,

"policyname":<String_value>,
```

```
"labelname":<String_value>,

"name":<String_value>,

"gotopriorityexpression":<String_value>,

"invoke":<Boolean_value>,

"labeltype":<String_value>

}]}
```

count

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/lbvserver_appflowpolicy_binding/name_value<String>?**count=yes**

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{"lbvserver_appflowpolicy_binding": [ { "__count": "#no"} ] }
```

lbvserver_appfwpolicy_binding

Binding object showing the appfwpolicy that can be bound to lbvserver.

Properties

(click to see Operations)

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
priority	<Double>	Read-write	Priority.
bindpoint	<String>	Read-write	The bindpoint to which the policy is bound. Possible values = REQUEST, RESPONSE
policyname	<String>	Read-write	Name of the policy bound to the LB vserver.
labelname	<String>	Read-write	Name of the label invoked.
name	<String>	Read-write	Name for the virtual server. Must begin with an ASCII alphanumeric or underscore (_) character, and must contain only ASCII alphanumeric, underscore, hash (#), period (.), space, colon (:), at sign (@), equal sign (=), and hyphen (-) characters. Can be changed after the virtual server is created. CLI Users: If the name includes one or more spaces, enclose the name in double or single

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
			quotation marks (for example, "my vserver" or 'my vserver'). . Minimum length = 1
gotopriorityexpression	<String>	Read-write	Expression specifying the priority of the next policy which will get evaluated if the current policy rule evaluates to TRUE.
invoke	<Boolean>	Read-write	Invoke policies bound to a virtual server or policy label.
labeltype	<String>	Read-write	The invocation type. Possible values = reqvserver, resvserver, policylabel
sc	<String>	Read-only	Use SureConnect on the virtual server. Default value: OFF Possible values = ON, OFF
__count	<Double>	Read-only	count parameter

Operations

(click to see Properties)

ADD:| DELETE:| GET| GET (ALL)| COUNT

Some options that you can use for each operations:

- **Getting warnings in response:**NITRO allows you to get warnings in an operation by specifying the "warning" query parameter as "yes". For example, to get warnings while connecting to the Citrix ADC appliance, the URL is as follows:

  http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/login?warning=yes

  If any, the warnings are displayed in the response payload with the HTTP code "209 X-NITRO-WARNING".

- **Authenticated access for individual NITRO operations:**NITRO allows you to logon to the Citrix ADC appliance to perform individual operations. You can use this option instead of creating a NITRO session (using the login object) and then using that session to perform all operations,

  To do this, you must specify the username and password in the request header of the NITRO request as follows:

  X-NITRO-USER:<username>

  X-NITRO-PASS:<password>

  Note: In such cases, make sure that the request header DOES not include the following:

  Cookie:NITRO_AUTH_TOKEN=<tokenvalue>

`*Note:`*

Mandatory parameters are marked in redand placeholder content is marked in <green>.

add

URL:http://<Citrix-ADC-IP-address(NSIP)/nitro/v1/config/lbvserver_appfwpolicy_binding

HTTP Method:PUT

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Content-Type:application/json

Request Payload:

```
{
```

```
"lbvserver_appfwpolicy_binding":{

<b>"name":<String_value>,

</b>"policyname":<String_value>,

"priority":<Double_value>,

"gotopriorityexpression":<String_value>,

"bindpoint":<String_value>,

"invoke":<Boolean_value>,

"labeltype":<String_value>,

"labelname":<String_value>

}}
```

Response:

HTTP Status Code on Success: 201 Created HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error

delete

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/lbvserver_appfwpolicy_binding/name_value<String>

Query-parameters:

args

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/lbvserver_appfwpolicy_binding/name_value<String>?
args=policyname:<String_value>,bindpoint:<String_value>,priority:<Double_value>

HTTP Method:DELETE

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue>

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error

get

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/lbvserver_appfwpolicy_binding/name_value<String>

Query-parameters:

filter

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/lbvserver_appfwpolicy_binding/name_value<String>?**filter=property-name1:property-value1,property-name2:property-value2**

Use this query-parameter to get the filtered set of lbvserver_appfwpolicy_binding resources configured on Citrix ADC. Filtering can be done on any of the properties of the resource.

pagination

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/lbvserver_appfwpolicy_binding/name_value<String>?**pagesize=#no;pageno=#no**

Use this query-parameter to get the lbvserver_appfwpolicy_binding resources in chunks.

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "lbvserver_appfwpolicy_binding": [ {

"priority":<Double_value>,

"bindpoint":<String_value>,

"policyname":<String_value>,

"labelname":<String_value>,

"name":<String_value>,

"gotopriorityexpression":<String_value>,

"invoke":<Boolean_value>,

"labeltype":<String_value>,

"sc":<String_value>

}]}
```

get (all)

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/lbvserver_appfwpolicy_binding

Query-parameters:

bulkbindings

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/lbvserver_appfwpolicy_binding?**bulkbindings=yes**

NITRO allows you to fetch bindings in bulk.

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "lbvserver_appfwpolicy_binding": [ {

"priority":<Double_value>,

"bindpoint":<String_value>,

"policyname":<String_value>,

"labelname":<String_value>,

"name":<String_value>,

"gotopriorityexpression":<String_value>,

"invoke":<Boolean_value>,

"labeltype":<String_value>,

"sc":<String_value>

}]}
```

count

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/lbvserver_appfwpolicy_binding/name_value<String>?**count=yes**

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{"lbvserver_appfwpolicy_binding": [ { "__count": "#no"} ] }
```

lbvserver_appqoepolicy_binding

Binding object showing the appqoepolicy that can be bound to lbvserver.

Properties

(click to see Operations)

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
priority	<Double>	Read-write	Priority.
bindpoint	<String>	Read-write	The bindpoint to which the policy is bound. Possible values = REQUEST, RESPONSE
policyname	<String>	Read-write	Name of the policy bound to the LB vserver.
labelname	<String>	Read-write	Name of the virtual server or user-defined policy label

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
			to invoke if the policy evaluates to TRUE.
name	<String>	Read-write	Name for the virtual server. Must begin with an ASCII alphanumeric or underscore (_) character, and must contain only ASCII alphanumeric, underscore, hash (#), period (.), space, colon (:), at sign (@), equal sign (=), and hyphen (-) characters. Can be changed after the virtual server is created. CLI Users: If the name includes one or more spaces, enclose the name in double or single quotation marks (for example, "my vserver" or 'my vserver'). . Minimum length = 1
gotopriorityexpression	<String>	Read-write	Expression specifying the priority of the next policy which will get evaluated if the current policy rule evaluates to TRUE.
invoke	<Boolean>	Read-write	Invoke policies bound to a virtual server or policy label.

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
labeltype	<String>	Read-write	Type of policy label to invoke. Applicable only to rewrite, videooptimization and cache policies. Available settings function as follows: * reqvserver - Evaluate the request against the request-based policies bound to the specified virtual server. * resvserver - Evaluate the response against the response-based policies bound to the specified virtual server. * policylabel - invoke the request or response against the specified user-defined policy label. Possible values = reqvserver, resvserver, policylabel
__count	<Double>	Read-write	count parameter

Operations

(click to see Properties)

ADD:| DELETE:| GET| GET (ALL)| COUNT

Some options that you can use for each operations:

- **Getting warnings in response:**NITRO allows you to get warnings in an operation by specifying the "warning" query parameter as "yes". For example, to get warnings while connecting to the Citrix ADC appliance, the URL is as follows:

  http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/login?warning=yes

  If any, the warnings are displayed in the response payload with the HTTP code "209 X-NITRO-WARNING".

- **Authenticated access for individual NITRO operations:**NITRO allows you to logon to the Citrix ADC appliance to perform individual operations. You can use this option instead of creating a NITRO session (using the login object) and then using that session to perform all operations,

  To do this, you must specify the username and password in the request header of the NITRO request as follows:

  X-NITRO-USER:<username>

  X-NITRO-PASS:<password>

  Note: In such cases, make sure that the request header DOES not include the following:

  Cookie:NITRO_AUTH_TOKEN=<tokenvalue>

`*Note:`*

Mandatory parameters are marked in redand placeholder content is marked in <green>.

add

URL:http://<Citrix-ADC-IP-address(NSIP)/nitro/v1/config/lbvserver_appqoepolicy_binding

HTTP Method:PUT

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Content-Type:application/json

Request Payload:

```
{
```

```
"lbvserver_appqoepolicy_binding":{

<b>"name":<String_value>,

</b>"policyname":<String_value>,

"priority":<Double_value>,

"gotopriorityexpression":<String_value>,

"bindpoint":<String_value>,

"invoke":<Boolean_value>,

"labeltype":<String_value>,

"labelname":<String_value>

}}
```

Response:

HTTP Status Code on Success: 201 Created HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error

delete

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/lbvserver_appqoepolicy_binding/name_value<String>

Query-parameters:

args

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/lbvserver_appqoepolicy_binding/name_value<String>?
args=policyname:<String_value>,bindpoint:<String_value>,priority:<Double_value>

HTTP Method:DELETE

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue>

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error

get

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/lbvserver_appqoepolicy_binding/name_value<String>

Query-parameters:

filter

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/lbvserver_appqoepolicy_binding/name_value<String>?**filter=property-name1:property-value1,property-name2:property-value2**

Use this query-parameter to get the filtered set of lbvserver_appqoepolicy_binding resources configured on Citrix ADC. Filtering can be done on any of the properties of the resource.

pagination

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/lbvserver_appqoepolicy_binding/name_value<String>?**pagesize=#no;pageno=#no**

Use this query-parameter to get the lbvserver_appqoepolicy_binding resources in chunks.

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "lbvserver_appqoepolicy_binding": [ {

"priority":<Double_value>,

"bindpoint":<String_value>,

"policyname":<String_value>,

"labelname":<String_value>,

"name":<String_value>,

"gotopriorityexpression":<String_value>,

"invoke":<Boolean_value>,

"labeltype":<String_value>

}]}
```

get (all)

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/lbvserver_appqoepolicy_binding

Query-parameters:

bulkbindings

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/lbvserver_appqoepolicy_binding?**bulkbindings=yes**

NITRO allows you to fetch bindings in bulk.

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "lbvserver_appqoepolicy_binding": [ {

"priority":<Double_value>,

"bindpoint":<String_value>,

"policyname":<String_value>,

"labelname":<String_value>,

"name":<String_value>,

"gotopriorityexpression":<String_value>,

"invoke":<Boolean_value>,

"labeltype":<String_value>

}]}
```

count

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/lbvserver_appqoepolicy_binding/name_value<String>?**count=yes**

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{"lbvserver_appqoepolicy_binding": [ { "__count": "#no"} ] }
```

lbvserver_auditnslogpolicy_binding

Binding object showing the auditnslogpolicy that can be bound to lbvserver.

Properties

(click to see Operations)

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
priority	<Double>	Read-write	Priority.
bindpoint	<String>	Read-write	Bind point to which to bind the policy. Applicable only to compression, rewrite, videooptimization and cache policies. Possible values = REQUEST, RESPONSE
policyname	<String>	Read-write	Name of the policy bound to the LB vserver.
labelname	<String>	Read-write	Name of the virtual server or user-defined policy label

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
			to invoke if the policy evaluates to TRUE.
name	<String>	Read-write	Name for the virtual server. Must begin with an ASCII alphanumeric or underscore (_) character, and must contain only ASCII alphanumeric, underscore, hash (#), period (.), space, colon (:), at sign (@), equal sign (=), and hyphen (-) characters. Can be changed after the virtual server is created. CLI Users: If the name includes one or more spaces, enclose the name in double or single quotation marks (for example, "my vserver" or 'my vserver'). . Minimum length = 1
gotopriorityexpression	<String>	Read-write	Expression or other value specifying the next policy to be evaluated if the current policy evaluates to TRUE. Specify one of the following values: * NEXT - Evaluate the policy with the next higher priority

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
			number. * END - End policy evaluation. * USE_INVOCATION_RESULT - Applicable if this policy invokes another policy label. If the final goto in the invoked policy label has a value of END, the evaluation stops. If the final goto is anything other than END, the current policy label performs a NEXT. * An expression that evaluates to a number. If you specify an expression, the number to which it evaluates determines the next policy to evaluate, as follows: * If the expression evaluates to a higher numbered priority, the policy with that priority is evaluated next. * If the expression evaluates to the priority of the current policy, the policy with the next higher numbered priority is evaluated next. * If the expression evaluates to a priority number that is numerically

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
			higher than the highest numbered priority, policy evaluation ends. An UNDEF event is triggered if: * The expression is invalid. * The expression evaluates to a priority number that is numerically lower than the current policy's priority. * The expression evaluates to a priority number that is between the current policy's priority number (say, 30) and the highest priority number (say, 100), but does not match any configured priority number (for example, the expression evaluates to the number 85). This example assumes that the priority number increments by 10 for every successive policy, and therefore a priority number of 85 does not exist in the policy label.
invoke	<Boolean>	Read-write	Invoke policies bound to a virtual server or policy label.

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
labeltype	<String>	Read-write	Type of policy label to invoke. Applicable only to rewrite, videooptimization and cache policies. Available settings function as follows: * reqvserver - Evaluate the request against the request-based policies bound to the specified virtual server. * resvserver - Evaluate the response against the response-based policies bound to the specified virtual server. * policylabel - invoke the request or response against the specified user-defined policy label. Possible values = reqvserver, resvserver, policylabel
sc	<String>	Read-only	Use SureConnect on the virtual server. Default value: OFF Possible values = ON, OFF
__count	<Double>	Read-only	count parameter

Operations

(click to see Properties)

ADD:| DELETE:| GET| GET (ALL)| COUNT

Some options that you can use for each operations:

- **Getting warnings in response:**NITRO allows you to get warnings in an operation by specifying the "warning" query parameter as "yes". For example, to get warnings while connecting to the Citrix ADC appliance, the URL is as follows:

  http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/login?warning=yes

  If any, the warnings are displayed in the response payload with the HTTP code "209 X-NITRO-WARNING".

- **Authenticated access for individual NITRO operations:**NITRO allows you to logon to the Citrix ADC appliance to perform individual operations. You can use this option instead of creating a NITRO session (using the login object) and then using that session to perform all operations,

  To do this, you must specify the username and password in the request header of the NITRO request as follows:

  X-NITRO-USER:<username>

  X-NITRO-PASS:<password>

  Note: In such cases, make sure that the request header DOES not include the following:

  Cookie:NITRO_AUTH_TOKEN=<tokenvalue>

`*Note:`*

Mandatory parameters are marked in redand placeholder content is marked in <green>.

add

URL:http://<Citrix-ADC-IP-address(NSIP)/nitro/v1/config/lbvserver_auditnslogpolicy_binding

HTTP Method:PUT

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Content-Type:application/json

Request Payload:

```
{

"lbvserver_auditnslogpolicy_binding":{

<b>"name":<String_value>,

</b>"policyname":<String_value>,

"priority":<Double_value>,

"gotopriorityexpression":<String_value>,

"bindpoint":<String_value>,

"invoke":<Boolean_value>,

"labeltype":<String_value>,

"labelname":<String_value>

}}
```

Response:

HTTP Status Code on Success: 201 Created HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error

delete

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/lbvserver_auditnslogpolicy_binding/name_value<String>

Query-parameters:

args

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/lbvserver_auditnslogpolicy_binding/name_value<String>?
args=policyname:<String_value>,bindpoint:<String_value>,priority:<Double_value>

HTTP Method:DELETE

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue>

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error

get

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/lbvserver_auditnslogpolicy_binding/name_value<String>

Query-parameters:

filter

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/lbvserver_auditnslogpolicy_binding/name_value<String>?**filter=property-name1:property-value1,property-name2:property-value2**

Use this query-parameter to get the filtered set of lbvserver_auditnslogpolicy_binding resources configured on Citrix ADC. Filtering can be done on any of the properties of the resource.

pagination

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/lbvserver_auditnslogpolicy_binding/name_value<String>?**pagesize=#no;pageno=#no**

Use this query-parameter to get the lbvserver_auditnslogpolicy_binding resources in chunks.

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "lbvserver_auditnslogpolicy_binding": [ {

"priority":<Double_value>,

"bindpoint":<String_value>,

"policyname":<String_value>,

"labelname":<String_value>,

"name":<String_value>,

"gotopriorityexpression":<String_value>,

"invoke":<Boolean_value>,

"labeltype":<String_value>,

"sc":<String_value>

}]}
```

get (all)

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/lbvserver_auditnslogpolicy_binding

Query-parameters:

bulkbindings

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/lbvserver_auditnslogpolicy_binding?**bulkbindings=yes**

NITRO allows you to fetch bindings in bulk.

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "lbvserver_auditnslogpolicy_binding": [ {

"priority":<Double_value>,

"bindpoint":<String_value>,

"policyname":<String_value>,

"labelname":<String_value>,

"name":<String_value>,

"gotopriorityexpression":<String_value>,

"invoke":<Boolean_value>,

"labeltype":<String_value>,

"sc":<String_value>

}]}
```

count

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/lbvserver_auditnslogpolicy_binding/name_value<String>?**count=yes**

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{"lbvserver_auditnslogpolicy_binding": [ { "__count": "#no"} ] }
```

lbvserver_auditsyslogpolicy_binding

Binding object showing the auditsyslogpolicy that can be bound to lbvserver.

Properties

(click to see Operations)

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
priority	<Double>	Read-write	Priority.
bindpoint	<String>	Read-write	Bind point to which to bind the policy. Applicable only to compression, rewrite, videooptimization and cache policies. Possible values = REQUEST, RESPONSE

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
policyname	<String>	Read-write	Name of the policy bound to the LB vserver.
labelname	<String>	Read-write	Name of the virtual server or user-defined policy label to invoke if the policy evaluates to TRUE.
name	<String>	Read-write	Name for the virtual server. Must begin with an ASCII alphanumeric or underscore (_) character, and must contain only ASCII alphanumeric, underscore, hash (#), period (.), space, colon (:), at sign (@), equal sign (=), and hyphen (-) characters. Can be changed after the virtual server is created. CLI Users: If the name includes one or more spaces, enclose the name in double or single quotation marks (for example, "my vserver" or 'my vserver'). . Minimum length = 1
gotopriorityexpression	<String>	Read-write	Expression or other value specifying the next policy to be evaluated if the

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
			current policy evaluates to TRUE. Specify one of the following values: * NEXT - Evaluate the policy with the next higher priority number. * END - End policy evaluation. * USE_INVOCATION_RESULT - Applicable if this policy invokes another policy label. If the final goto in the invoked policy label has a value of END, the evaluation stops. If the final goto is anything other than END, the current policy label performs a NEXT. * An expression that evaluates to a number. If you specify an expression, the number to which it evaluates determines the next policy to evaluate, as follows: * If the expression evaluates to a higher numbered priority, the policy with that priority is evaluated next. * If the expression evaluates to the priority of the current policy, the

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
			policy with the next higher numbered priority is evaluated next. * If the expression evaluates to a priority number that is numerically higher than the highest numbered priority, policy evaluation ends. An UNDEF event is triggered if: * The expression is invalid. * The expression evaluates to a priority number that is numerically lower than the current policy's priority. * The expression evaluates to a priority number that is between the current policy's priority number (say, 30) and the highest priority number (say, 100), but does not match any configured priority number (for example, the expression evaluates to the number 85). This example assumes that the priority number increments by 10 for every successive policy, and therefore a priority number of

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
			85 does not exist in the policy label.
invoke	<Boolean>	Read-write	Invoke policies bound to a virtual server or policy label.
labeltype	<String>	Read-write	Type of policy label to invoke. Applicable only to rewrite, videooptimization and cache policies. Available settings function as follows: * reqvserver - Evaluate the request against the request-based policies bound to the specified virtual server. * resvserver - Evaluate the response against the response-based policies bound to the specified virtual server. * policylabel - invoke the request or response against the specified user-defined policy label. Possible values = reqvserver, resvserver, policylabel
sc	<String>	Read-only	Use SureConnect on the virtual server. Default value: OFF

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
			Possible values = ON, OFF
__count	<Double>	Read-only	count parameter

Operations

(click to see Properties)

ADD:| DELETE:| GET| GET (ALL)| COUNT

Some options that you can use for each operations:

- **Getting warnings in response:**NITRO allows you to get warnings in an operation by specifying the "warning" query parameter as "yes". For example, to get warnings while connecting to the Citrix ADC appliance, the URL is as follows:

  http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/login?warning=yes

  If any, the warnings are displayed in the response payload with the HTTP code "209 X-NITRO-WARNING".

- **Authenticated access for individual NITRO operations:**NITRO allows you to logon to the Citrix ADC appliance to perform individual operations. You can use this option instead of creating a NITRO session (using the login object) and then using that session to perform all operations,

  To do this, you must specify the username and password in the request header of the NITRO request as follows:

  X-NITRO-USER:<username>

  X-NITRO-PASS:<password>

  Note: In such cases, make sure that the request header DOES not include the following:

  Cookie:NITRO_AUTH_TOKEN=<tokenvalue>

`*Note:`*

Mandatory parameters are marked in redand placeholder content is marked in <green>.

add

URL:http://<Citrix-ADC-IP-address(NSIP)/nitro/v1/config/lbvserver_auditsyslogpolicy_binding

HTTP Method:PUT

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Content-Type:application/json

Request Payload:

```
{

"lbvserver_auditsyslogpolicy_binding":{

<b>"name":<String_value>,

</b>"policyname":<String_value>,

"priority":<Double_value>,

"gotopriorityexpression":<String_value>,

"bindpoint":<String_value>,

"invoke":<Boolean_value>,

"labeltype":<String_value>,

"labelname":<String_value>

}}
```

Response:

HTTP Status Code on Success: 201 Created HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error

delete

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/lbvserver_auditsyslogpolicy_binding/name_value<String>

Query-parameters:

args

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/lbvserver_auditsyslogpolicy_binding/name_value<String>?**args=policyname:<String_value>,bindpoint:<String_value>,priority:<Double_value>**

HTTP Method:DELETE

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue>

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error

get

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/lbvserver_auditsyslogpolicy_binding/name_value<String>

Query-parameters:

filter

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/lbvserver_auditsyslogpolicy_binding/name_value<String>?**filter=property-name1:property-value1,property-name2:property-value2**

Use this query-parameter to get the filtered set of lbvserver_auditsyslogpolicy_binding resources configured on Citrix ADC. Filtering can be done on any of the properties of the resource.

pagination

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/lbvserver_auditsyslogpolicy_binding/name_value<String>?**pagesize=#no;pageno=#no**

Use this query-parameter to get the lbvserver_auditsyslogpolicy_binding resources in chunks.

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "lbvserver_auditsyslogpolicy_binding": [ {

"priority":<Double_value>,

"bindpoint":<String_value>,

"policyname":<String_value>,

"labelname":<String_value>,

"name":<String_value>,

"gotopriorityexpression":<String_value>,

"invoke":<Boolean_value>,

"labeltype":<String_value>,

"sc":<String_value>

}]}
```

get (all)

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/lbvserver_auditsyslogpolicy_binding

Query-parameters:

bulkbindings

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/lbvserver_auditsyslogpolicy_binding?**bulkbindings=yes**

NITRO allows you to fetch bindings in bulk.

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "lbvserver_auditsyslogpolicy_binding": [ {

"priority":<Double_value>,

"bindpoint":<String_value>,

"policyname":<String_value>,

"labelname":<String_value>,

"name":<String_value>,

"gotopriorityexpression":<String_value>,

"invoke":<Boolean_value>,

"labeltype":<String_value>,
```

```
"sc":<String_value>

}]}
```

count

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/lbvserver_auditsyslogpolicy_binding/name_value<String>?**count=yes**

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{"lbvserver_auditsyslogpolicy_binding": [ { "__count": "#no"} ] }
```

lbvserver_authorizationpolicy_binding

Binding object showing the authorizationpolicy that can be bound to lbvserver.

Properties

(click to see Operations)

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
priority	<Double>	Read-write	Priority.

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
bindpoint	<String>	Read-write	The bindpoint to which the policy is bound. Possible values = REQUEST, RESPONSE
policyname	<String>	Read-write	Name of the policy bound to the LB vserver.
labelname	<String>	Read-write	Name of the label invoked.
name	<String>	Read-write	Name for the virtual server. Must begin with an ASCII alphanumeric or underscore (_) character, and must contain only ASCII alphanumeric, underscore, hash (#), period (.), space, colon (:), at sign (@), equal sign (=), and hyphen (-) characters. Can be changed after the virtual server is created. CLI Users: If the name includes one or more spaces, enclose the name in double or single quotation marks (for example, "my vserver" or 'my vserver'). . Minimum length = 1

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
gotopriorityexpression	<String>	Read-write	Expression specifying the priority of the next policy which will get evaluated if the current policy rule evaluates to TRUE.
invoke	<Boolean>	Read-write	Invoke policies bound to a virtual server or policy label.
labeltype	<String>	Read-write	The invocation type. Possible values = reqvserver, resvserver, policylabel
sc	<String>	Read-only	Use SureConnect on the virtual server. Default value: OFF Possible values = ON, OFF
__count	<Double>	Read-only	count parameter

Operations

(click to see Properties)

ADD:| DELETE:| GET| GET (ALL)| COUNT

Some options that you can use for each operations:

- **Getting warnings in response:**NITRO allows you to get warnings in an operation by specifying the "warning" query parameter as "yes". For example, to get warnings while connecting to the Citrix ADC appliance, the URL is as follows:

  http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/login?warning=yes

If any, the warnings are displayed in the response payload with the HTTP code "209 X-NITRO-WARNING".

- **Authenticated access for individual NITRO operations:**NITRO allows you to logon to the Citrix ADC appliance to perform individual operations. You can use this option instead of creating a NITRO session (using the login object) and then using that session to perform all operations,

  To do this, you must specify the username and password in the request header of the NITRO request as follows:

  X-NITRO-USER:<username>

  X-NITRO-PASS:<password>

  Note: In such cases, make sure that the request header DOES not include the following:

  Cookie:NITRO_AUTH_TOKEN=<tokenvalue>

`*Note:`*

Mandatory parameters are marked in redand placeholder content is marked in <green>.

add

URL:http://<Citrix-ADC-IP-address(NSIP)/nitro/v1/config/lbvserver_authorizationpolicy_binding

HTTP Method:PUT

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Content-Type:application/json

Request Payload:

```
{

"lbvserver_authorizationpolicy_binding":{

<b>"name":<String_value>,

</b>"policyname":<String_value>,
```

```
"priority":<Double_value>,

"gotopriorityexpression":<String_value>,

"bindpoint":<String_value>,

"invoke":<Boolean_value>,

"labeltype":<String_value>,

"labelname":<String_value>

}}
```

Response:

HTTP Status Code on Success: 201 Created HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error

delete

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/lbvserver_authorizationpolicy_binding/name_value<String>

Query-parameters:

args

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/lbvserver_authorizationpolicy_binding/name_value<String>?
args=policyname:<String_value>,bindpoint:<String_value>,priority:<Double_value>

HTTP Method:DELETE

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue>

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error

get

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/lbvserver_authorizationpolicy_binding/name_value<String>

Query-parameters:

filter

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/lbvserver_authorizationpolicy_binding/name_value<String>?**filter=property-name1:property-value1,property-name2:property-value2**

Use this query-parameter to get the filtered set of lbvserver_authorizationpolicy_binding resources configured on Citrix ADC. Filtering can be done on any of the properties of the resource.

pagination

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/lbvserver_authorizationpolicy_binding/name_value<String>?**pagesize=#no;pageno=#no**

Use this query-parameter to get the lbvserver_authorizationpolicy_binding resources in chunks.

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "lbvserver_authorizationpolicy_binding": [ {
```

```
"priority":<Double_value>,

"bindpoint":<String_value>,

"policyname":<String_value>,

"labelname":<String_value>,

"name":<String_value>,

"gotopriorityexpression":<String_value>,

"invoke":<Boolean_value>,

"labeltype":<String_value>,

"sc":<String_value>

}]}
```

get (all)

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/lbvserver_authorizationpolicy_binding

Query-parameters:

bulkbindings

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/lbvserver_authorizationpolicy_binding?**bulkbindings=yes**

NITRO allows you to fetch bindings in bulk.

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "lbvserver_authorizationpolicy_binding": [ {

"priority":<Double_value>,

"bindpoint":<String_value>,

"policyname":<String_value>,

"labelname":<String_value>,

"name":<String_value>,

"gotopriorityexpression":<String_value>,

"invoke":<Boolean_value>,

"labeltype":<String_value>,

"sc":<String_value>

}]}
```

count

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/lbvserver_authorizationpolicy_binding/name_value<String>?**count=yes**

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{"lbvserver_authorizationpolicy_binding": [ { "__count": "#no"} ] }
```

lbvserver_binding

Binding object which returns the resources bound to lbvserver.

Properties

(click to see Operations)

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
name	<String>	Read-write	Name of the virtual server. If no name is provided, statistical data of all configured virtual servers is displayed. Minimum length = 1
lbvserver_spilloverpolicy_binding	<lbvserver_spilloverpolicy_binding[]>	Read-only	spilloverpolicy that can be bound to lbvserver.
lbvserver_csvserver_binding	<lbvserver_csvserver_binding[]>	Read-only	csvserver that can be bound to lbvserver.
lbvserver_capolicy_binding	<lbvserver_capolicy_binding[]>	Read-only	capolicy that can be bound to lbvserver.

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
lbvserver_auditnslogpolicy_binding	<lbvserver_auditnslogpolicy_binding[]>	Read-only	auditnslogpolicy that can be bound to lbvserver.
lbvserver_filterpolicy_binding	<lbvserver_filterpolicy_binding[]>	Read-only	filterpolicy that can be bound to lbvserver.
lbvserver_analyticsprofile_binding	<lbvserver_analyticsprofile_binding[]>	Read-only	analyticsprofile that can be bound to lbvserver.
lbvserver_cmppolicy_binding	<lbvserver_cmppolicy_binding[]>	Read-only	cmppolicy that can be bound to lbvserver.
lbvserver_appflowpolicy_binding	<lbvserver_appflowpolicy_binding[]>	Read-only	appflowpolicy that can be bound to lbvserver.
lbvserver_responderpolicy_binding	<lbvserver_responderpolicy_binding[]>	Read-only	responderpolicy that can be bound to lbvserver.
lbvserver_videooptimizationdetectionpolicy_binding	<lbvserver_videooptimizationdetectionpolicy_binding[]>	Read-only	videooptimizationdetectionpolicy that can be bound to lbvserver.
lbvserver_transformpolicy_binding	<lbvserver_transformpolicy_binding[]>	Read-only	transformpolicy that can be bound to lbvserver.
lbvserver_dospolicy_binding	<lbvserver_dospolicy_binding[]>	Read-only	dospolicy that can be bound to lbvserver.
lbvserver_contentinspectionpolicy_binding	<lbvserver_contentinspectionpolicy_binding[]>	Read-only	contentinspectionpolicy that can be bound to lbvserver.

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
lbvserver_servicegroupmember_binding	<lbvserver_servicegroupmember_binding[]>	Read-only	servicegroupmember that can be bound to lbvserver.
lbvserver_feopolicy_binding	<lbvserver_feopolicy_binding[]>	Read-only	feopolicy that can be bound to lbvserver.
lbvserver_videooptimizationpacingpolicy_binding	<lbvserver_videooptimizationpacingpolicy_binding[]>	Read-only	videooptimizationpacingpolicy that can be bound to lbvserver.
lbvserver_authorizationpolicy_binding	<lbvserver_authorizationpolicy_binding[]>	Read-only	authorizationpolicy that can be bound to lbvserver.
lbvserver_dnspolicy64_binding	<lbvserver_dnspolicy64_binding[]>	Read-only	dnspolicy64 that can be bound to lbvserver.
lbvserver_cachepolicy_binding	<lbvserver_cachepolicy_binding[]>	Read-only	cachepolicy that can be bound to lbvserver.
lbvserver_rewritepolicy_binding	<lbvserver_rewritepolicy_binding[]>	Read-only	rewritepolicy that can be bound to lbvserver.
lbvserver_scpolicy_binding	<lbvserver_scpolicy_binding[]>	Read-only	scpolicy that can be bound to lbvserver.
lbvserver_pqpolicy_binding	<lbvserver_pqpolicy_binding[]>	Read-only	pqpolicy that can be bound to lbvserver.
lbvserver_service_binding	<lbvserver_service_binding[]>	Read-only	service that can be bound to lbvserver.
lbvserver_servicegroup_binding	<lbvserver_servicegroup_binding[]>	Read-only	servicegroup that can be bound to lbvserver.

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
lbvserver_tmtrafficpolicy_binding	<lbvserver_tmtrafficpolicy_binding[]>	Read-only	tmtrafficpolicy that can be bound to lbvserver.
lbvserver_auditsyslogpolicy_binding	<lbvserver_auditsyslogpolicy_binding[]>	Read-only	auditsyslogpolicy that can be bound to lbvserver.
lbvserver_appqoepolicy_binding	<lbvserver_appqoepolicy_binding[]>	Read-only	appqoepolicy that can be bound to lbvserver.
lbvserver_appfwpolicy_binding	<lbvserver_appfwpolicy_binding[]>	Read-only	appfwpolicy that can be bound to lbvserver.

Operations

(click to see Properties)

GET| GET (ALL)

Some options that you can use for each operations:

- **Getting warnings in response:**NITRO allows you to get warnings in an operation by specifying the "warning" query parameter as "yes". For example, to get warnings while connecting to the Citrix ADC appliance, the URL is as follows:

  http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/login?warning=yes

  If any, the warnings are displayed in the response payload with the HTTP code "209 X-NITRO-WARNING".

- **Authenticated access for individual NITRO operations:**NITRO allows you to logon to the Citrix ADC appliance to perform individual operations. You can use this option instead of creating a NITRO session (using the login object) and then using that session to perform all operations,

  To do this, you must specify the username and password in the request header of the NITRO request as follows:

  X-NITRO-USER:<username>

X-NITRO-PASS:<password>

Note: In such cases, make sure that the request header DOES not include the following:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue>

`*Note:`*

Mandatory parameters are marked in redand placeholder content is marked in <green>.

get

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/lbvserver_binding/name_value<String>

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "lbvserver_binding": [ {

"name":<String_value>,

"lbvserver_auditsyslogpolicy_binding":<lbvserver_auditsyslogpolicy_binding
[]_value>,

"lbvserver_pqpolicy_binding":<lbvserver_pqpolicy_binding[]_value>,

"lbvserver_dnspolicy64_binding":<lbvserver_dnspolicy64_binding[]_value>,

"lbvserver_appfwpolicy_binding":<lbvserver_appfwpolicy_binding[]_value>,
```

```
"lbvserver_rewritepolicy_binding":<lbvserver_rewritepolicy_binding[]_value
>,

"lbvserver_videooptimizationpacingpolicy_binding":<lbvserver_videooptimiza
tionpacingpolicy_binding[]_value>,

"lbvserver_contentinspectionpolicy_binding":<lbvserver_contentinspectionpo
licy_binding[]_value>,

"lbvserver_spilloverpolicy_binding":<lbvserver_spilloverpolicy_binding[]_v
alue>,

"lbvserver_auditnslogpolicy_binding":<lbvserver_auditnslogpolicy_binding[]
_value>,

"lbvserver_appqoepolicy_binding":<lbvserver_appqoepolicy_binding[]_value>,

"lbvserver_transformpolicy_binding":<lbvserver_transformpolicy_binding[]_v
alue>,

"lbvserver_filterpolicy_binding":<lbvserver_filterpolicy_binding[]_value>,

"lbvserver_scpolicy_binding":<lbvserver_scpolicy_binding[]_value>,

"lbvserver_feopolicy_binding":<lbvserver_feopolicy_binding[]_value>,

"lbvserver_csvserver_binding":<lbvserver_csvserver_binding[]_value>,

"lbvserver_appflowpolicy_binding":<lbvserver_appflowpolicy_binding[]_value
>,

"lbvserver_analyticsprofile_binding":<lbvserver_analyticsprofile_binding[]
_value>,

"lbvserver_videooptimizationdetectionpolicy_binding":<lbvserver_videooptim
izationdetectionpolicy_binding[]_value>,

"lbvserver_authorizationpolicy_binding":<lbvserver_authorizationpolicy_bin
ding[]_value>,

"lbvserver_capolicy_binding":<lbvserver_capolicy_binding[]_value>,

"lbvserver_servicegroup_binding":<lbvserver_servicegroup_binding[]_value>,
```

```
"lbvserver_cachepolicy_binding":<lbvserver_cachepolicy_binding[]_value>,

"lbvserver_service_binding":<lbvserver_service_binding[]_value>,

"lbvserver_servicegroupmember_binding":<lbvserver_servicegroupmember_bindi
ng[]_value>,

"lbvserver_responderpolicy_binding":<lbvserver_responderpolicy_binding[]_v
alue>,

"lbvserver_dospolicy_binding":<lbvserver_dospolicy_binding[]_value>,

"lbvserver_tmtrafficpolicy_binding":<lbvserver_tmtrafficpolicy_binding[]_v
alue>,

"lbvserver_cmppolicy_binding":<lbvserver_cmppolicy_binding[]_value>

}]}
```

get (all)

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/lbvserver_binding

Query-parameters:

bulkbindings

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/lbvserver_binding?**bulkbindings=yes**

NITRO allows you to fetch bindings in bulk.

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "lbvserver_binding": [ {

"name":<String_value>,

"lbvserver_auditsyslogpolicy_binding":<lbvserver_auditsyslogpolicy_binding
[]_value>,

"lbvserver_pqpolicy_binding":<lbvserver_pqpolicy_binding[]_value>,

"lbvserver_dnspolicy64_binding":<lbvserver_dnspolicy64_binding[]_value>,

"lbvserver_appfwpolicy_binding":<lbvserver_appfwpolicy_binding[]_value>,

"lbvserver_rewritepolicy_binding":<lbvserver_rewritepolicy_binding[]_value
>,

"lbvserver_videooptimizationpacingpolicy_binding":<lbvserver_videooptimiza
tionpacingpolicy_binding[]_value>,

"lbvserver_contentinspectionpolicy_binding":<lbvserver_contentinspectionpo
licy_binding[]_value>,

"lbvserver_spilloverpolicy_binding":<lbvserver_spilloverpolicy_binding[]_v
alue>,

"lbvserver_auditnslogpolicy_binding":<lbvserver_auditnslogpolicy_binding[]
_value>,

"lbvserver_appqoepolicy_binding":<lbvserver_appqoepolicy_binding[]_value>,

"lbvserver_transformpolicy_binding":<lbvserver_transformpolicy_binding[]_v
alue>,

"lbvserver_filterpolicy_binding":<lbvserver_filterpolicy_binding[]_value>,

"lbvserver_scpolicy_binding":<lbvserver_scpolicy_binding[]_value>,
```

```
"lbvserver_feopolicy_binding":<lbvserver_feopolicy_binding[]_value>,

"lbvserver_csvserver_binding":<lbvserver_csvserver_binding[]_value>,

"lbvserver_appflowpolicy_binding":<lbvserver_appflowpolicy_binding[]_value
>,

"lbvserver_analyticsprofile_binding":<lbvserver_analyticsprofile_binding[]
_value>,

"lbvserver_videooptimizationdetectionpolicy_binding":<lbvserver_videooptim
izationdetectionpolicy_binding[]_value>,

"lbvserver_authorizationpolicy_binding":<lbvserver_authorizationpolicy_bin
ding[]_value>,

"lbvserver_capolicy_binding":<lbvserver_capolicy_binding[]_value>,

"lbvserver_servicegroup_binding":<lbvserver_servicegroup_binding[]_value>,

"lbvserver_cachepolicy_binding":<lbvserver_cachepolicy_binding[]_value>,

"lbvserver_service_binding":<lbvserver_service_binding[]_value>,

"lbvserver_servicegroupmember_binding":<lbvserver_servicegroupmember_bindi
ng[]_value>,

"lbvserver_responderpolicy_binding":<lbvserver_responderpolicy_binding[]_v
alue>,

"lbvserver_dospolicy_binding":<lbvserver_dospolicy_binding[]_value>,

"lbvserver_tmtrafficpolicy_binding":<lbvserver_tmtrafficpolicy_binding[]_v
alue>,

"lbvserver_cmppolicy_binding":<lbvserver_cmppolicy_binding[]_value>

}]}
```

lbvserver_cachepolicy_binding

Binding object showing the cachepolicy that can be bound to lbvserver.

Properties

(click to see Operations)

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
priority	<Double>	Read-write	Priority.
bindpoint	<String>	Read-write	The bindpoint to which the policy is bound. Possible values = REQUEST, RESPONSE
policyname	<String>	Read-write	Name of the policy bound to the LB vserver.
labelname	<String>	Read-write	Name of the label invoked.
name	<String>	Read-write	Name for the virtual server. Must begin with an ASCII alphanumeric or underscore (_) character, and must contain only ASCII alphanumeric, underscore, hash (#), period (.), space, colon (:), at sign (@), equal sign (=), and hyphen (-) characters. Can be changed after the virtual server is created. CLI Users: If the name includes

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
			one or more spaces, enclose the name in double or single quotation marks (for example, "my vserver" or 'my vserver'). . Minimum length = 1
gotopriorityexpression	<String>	Read-write	Expression specifying the priority of the next policy which will get evaluated if the current policy rule evaluates to TRUE.
invoke	<Boolean>	Read-write	Invoke policies bound to a virtual server or policy label.
labeltype	<String>	Read-write	The invocation type. Possible values = reqvserver, resvserver, policylabel
__count	<Double>	Read-write	count parameter

Operations

(click to see Properties)

ADD:| DELETE:| GET| GET (ALL)| COUNT

Some options that you can use for each operations:

- **Getting warnings in response:**NITRO allows you to get warnings in an operation by specifying the "warning" query parameter as "yes". For example, to get warnings while connecting to the Citrix ADC appliance, the URL is as follows:

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/login?warning=yes

If any, the warnings are displayed in the response payload with the HTTP code "209 X-NITRO-WARNING".

- **Authenticated access for individual NITRO operations:**NITRO allows you to logon to the Citrix ADC appliance to perform individual operations. You can use this option instead of creating a NITRO session (using the login object) and then using that session to perform all operations,

  To do this, you must specify the username and password in the request header of the NITRO request as follows:

  X-NITRO-USER:<username>

  X-NITRO-PASS:<password>

  Note: In such cases, make sure that the request header DOES not include the following:

  Cookie:NITRO_AUTH_TOKEN=<tokenvalue>

`*Note:`*

Mandatory parameters are marked in redand placeholder content is marked in <green>.

add

URL:http://<Citrix-ADC-IP-address(NSIP)/nitro/v1/config/lbvserver_cachepolicy_binding

HTTP Method:PUT

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Content-Type:application/json

Request Payload:

```
{

"lbvserver_cachepolicy_binding":{

<b>"name":<String_value>,

</b>"policyname":<String_value>,
```

```
"priority":<Double_value>,

"gotopriorityexpression":<String_value>,

"bindpoint":<String_value>,

"invoke":<Boolean_value>,

"labeltype":<String_value>,

"labelname":<String_value>

}}
```

Response:

HTTP Status Code on Success: 201 Created HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error

delete

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/lbvserver_cachepolicy_binding/name_value<String>

Query-parameters:

args

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/lbvserver_cachepolicy_binding/name_value<String>?
args=policyname:<String_value>,bindpoint:<String_value>,priority:<Double_value>

HTTP Method:DELETE

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue>

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error

get

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/lbvserver_cachepolicy_binding/name_value<String>

Query-parameters:

filter

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/lbvserver_cachepolicy_binding/name_value<String>?**filter=property-name1:property-value1,property-name2:property-value2**

Use this query-parameter to get the filtered set of lbvserver_cachepolicy_binding resources configured on Citrix ADC. Filtering can be done on any of the properties of the resource.

pagination

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/lbvserver_cachepolicy_binding/name_value<String>?**pagesize=#no;pageno=#no**

Use this query-parameter to get the lbvserver_cachepolicy_binding resources in chunks.

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "lbvserver_cachepolicy_binding": [ {
```

```
"priority":<Double_value>,

"bindpoint":<String_value>,

"policyname":<String_value>,

"labelname":<String_value>,

"name":<String_value>,

"gotopriorityexpression":<String_value>,

"invoke":<Boolean_value>,

"labeltype":<String_value>

}]}
```

get (all)

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/lbvserver_cachepolicy_binding

Query-parameters:

bulkbindings

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/lbvserver_cachepolicy_binding?**bulkbindings=yes**

NITRO allows you to fetch bindings in bulk.

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "lbvserver_cachepolicy_binding": [ {

"priority":<Double_value>,

"bindpoint":<String_value>,

"policyname":<String_value>,

"labelname":<String_value>,

"name":<String_value>,

"gotopriorityexpression":<String_value>,

"invoke":<Boolean_value>,

"labeltype":<String_value>

}]}
```

count

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/lbvserver_cachepolicy_binding/name_value<String>?**count=yes**

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{"lbvserver_cachepolicy_binding": [ { "__count": "#no"} ] }
```

lbvserver_capolicy_binding

Binding object showing the capolicy that can be bound to lbvserver.

Properties

(click to see Operations)

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
priority	<Double>	Read-write	Priority.
bindpoint	<String>	Read-write	The bindpoint to which the policy is bound. Possible values = REQUEST, RESPONSE
policyname	<String>	Read-write	Name of the policy bound to the LB vserver.
labelname	<String>	Read-write	Name of the virtual server or user-defined policy label to invoke if the policy evaluates to TRUE.
name	<String>	Read-write	Name for the virtual server. Must begin with an ASCII alphanumeric or underscore (_)

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
			character, and must contain only ASCII alphanumeric, underscore, hash (#), period (.), space, colon (:), at sign (@), equal sign (=), and hyphen (-) characters. Can be changed after the virtual server is created. CLI Users: If the name includes one or more spaces, enclose the name in double or single quotation marks (for example, "my vserver" or 'my vserver'). . Minimum length = 1
gotopriorityexpression	<String>	Read-write	Expression specifying the priority of the next policy which will get evaluated if the current policy rule evaluates to TRUE.
invoke	<Boolean>	Read-write	Invoke policies bound to a virtual server or policy label.
labeltype	<String>	Read-write	Type of policy label to invoke. Applicable only to rewrite, videooptimization and cache policies. Available settings function as follows: *

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
			reqvserver - Evaluate the request against the request-based policies bound to the specified virtual server. * resvserver - Evaluate the response against the response-based policies bound to the specified virtual server. * policylabel - invoke the request or response against the specified user-defined policy label. Possible values = reqvserver, resvserver, policylabel
__count	<Double>	Read-write	count parameter

Operations

(click to see Properties)

ADD:| DELETE:| GET| GET (ALL)| COUNT

Some options that you can use for each operations:

- **Getting warnings in response:**NITRO allows you to get warnings in an operation by specifying the "warning" query parameter as "yes". For example, to get warnings while connecting to the Citrix ADC appliance, the URL is as follows:

  http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/login?warning=yes

  If any, the warnings are displayed in the response payload with the HTTP code "209 X-NITRO-WARNING".

- **Authenticated access for individual NITRO operations:**NITRO allows you to logon to the Citrix ADC appliance to perform individual operations. You can use this option instead of creating a NITRO session (using the login object) and then using that session to perform all operations,

  To do this, you must specify the username and password in the request header of the NITRO request as follows:

  X-NITRO-USER:<username>

  X-NITRO-PASS:<password>

  Note: In such cases, make sure that the request header DOES not include the following:

  Cookie:NITRO_AUTH_TOKEN=<tokenvalue>

`*Note:`*

Mandatory parameters are marked in redand placeholder content is marked in <green>.

add

URL:http://<Citrix-ADC-IP-address(NSIP)/nitro/v1/config/lbvserver_capolicy_binding

HTTP Method:PUT

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Content-Type:application/json

Request Payload:

```
{

"lbvserver_capolicy_binding":{

<b>"name":<String_value>,

</b>"policyname":<String_value>,

"priority":<Double_value>,

"gotopriorityexpression":<String_value>,
```

```
"bindpoint":<String_value>,

"invoke":<Boolean_value>,

"labeltype":<String_value>,

"labelname":<String_value>

}}
```

Response:

HTTP Status Code on Success: 201 Created HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error

delete

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/lbvserver_capolicy_binding/name_value<String>

Query-parameters:

args

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/lbvserver_capolicy_binding/name_value<String>?**args=policyname:<String_value>,bindpoint:<String_value>,priority:<Double_value>**

HTTP Method:DELETE

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue>

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error

get

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/lbvserver_capolicy_binding/name_value<String>

Query-parameters:

filter

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/lbvserver_capolicy_binding/name_value<String>?**filter=property-name1:property-value1,property-name2:property-value2**

Use this query-parameter to get the filtered set of lbvserver_capolicy_binding resources configured on Citrix ADC. Filtering can be done on any of the properties of the resource.

pagination

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/lbvserver_capolicy_binding/name_value<String>?**pagesize=#no;pageno=#no**

Use this query-parameter to get the lbvserver_capolicy_binding resources in chunks.

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "lbvserver_capolicy_binding": [ {

"priority":<Double_value>,

"bindpoint":<String_value>,
```

```
"policyname":<String_value>,

"labelname":<String_value>,

"name":<String_value>,

"gotopriorityexpression":<String_value>,

"invoke":<Boolean_value>,

"labeltype":<String_value>

}]}
```

get (all)

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/lbvserver_capolicy_binding

Query-parameters:

bulkbindings

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/lbvserver_capolicy_binding?**bulkbindings=yes**

NITRO allows you to fetch bindings in bulk.

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "lbvserver_capolicy_binding": [ {

"priority":<Double_value>,

"bindpoint":<String_value>,

"policyname":<String_value>,

"labelname":<String_value>,

"name":<String_value>,

"gotopriorityexpression":<String_value>,

"invoke":<Boolean_value>,

"labeltype":<String_value>

}]}
```

count

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/lbvserver_capolicy_binding/name_value<String>?**count=yes**

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{"lbvserver_capolicy_binding": [ { "__count": "#no"} ] }
```

lbvserver_cmppolicy_binding

Binding object showing the cmppolicy that can be bound to lbvserver.

Properties

(click to see Operations)

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
priority	<Double>	Read-write	Priority.
bindpoint	<String>	Read-write	The bindpoint to which the policy is bound. Possible values = REQUEST, RESPONSE
policyname	<String>	Read-write	Name of the policy bound to the LB vserver.
labelname	<String>	Read-write	Name of the label invoked.
name	<String>	Read-write	Name for the virtual server. Must begin with an ASCII alphanumeric or underscore (_) character, and must contain only ASCII alphanumeric, underscore, hash (#), period (.), space, colon (:), at sign (@),

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
			equal sign (=), and hyphen (-) characters. Can be changed after the virtual server is created. CLI Users: If the name includes one or more spaces, enclose the name in double or single quotation marks (for example, "my vserver" or 'my vserver'). . Minimum length = 1
gotopriorityexpression	<String>	Read-write	Expression specifying the priority of the next policy which will get evaluated if the current policy rule evaluates to TRUE.
invoke	<Boolean>	Read-write	Invoke policies bound to a virtual server or policy label.
labeltype	<String>	Read-write	The invocation type. Possible values = reqvserver, resvserver, policylabel
sc	<String>	Read-only	Use SureConnect on the virtual server. Default value: OFF Possible values = ON, OFF

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
__count	<Double>	Read-only	count parameter

Operations

(click to see Properties)

ADD:| DELETE:| GET| GET (ALL)| COUNT

Some options that you can use for each operations:

- **Getting warnings in response:**NITRO allows you to get warnings in an operation by specifying the "warning" query parameter as "yes". For example, to get warnings while connecting to the Citrix ADC appliance, the URL is as follows:

  http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/login?warning=yes

  If any, the warnings are displayed in the response payload with the HTTP code "209 X-NITRO-WARNING".
- **Authenticated access for individual NITRO operations:**NITRO allows you to logon to the Citrix ADC appliance to perform individual operations. You can use this option instead of creating a NITRO session (using the login object) and then using that session to perform all operations,

  To do this, you must specify the username and password in the request header of the NITRO request as follows:

  X-NITRO-USER:<username>

  X-NITRO-PASS:<password>

  Note: In such cases, make sure that the request header DOES not include the following:

  Cookie:NITRO_AUTH_TOKEN=<tokenvalue>

`*Note:`*

Mandatory parameters are marked in redand placeholder content is marked in <green>.

add

URL:http://<Citrix-ADC-IP-address(NSIP)/nitro/v1/config/lbvserver_cmppolicy_binding

HTTP Method:PUT

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Content-Type:application/json

Request Payload:

```
{

"lbvserver_cmppolicy_binding":{

<b>"name":<String_value>,

</b>"policyname":<String_value>,

"priority":<Double_value>,

"gotopriorityexpression":<String_value>,

"bindpoint":<String_value>,

"invoke":<Boolean_value>,

"labeltype":<String_value>,

"labelname":<String_value>

}}
```

Response:

HTTP Status Code on Success: 201 Created HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error

delete

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/lbvserver_cmppolicy_binding/name_value<String>

Query-parameters:

args

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/lbvserver_cmppolicy_binding/name_value<String>?**args=policyname:<String_value>,bindpoint:<String_value>,priority:<Double_value>**

HTTP Method:DELETE

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue>

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error

get

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/lbvserver_cmppolicy_binding/name_value<String>

Query-parameters:

filter

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/lbvserver_cmppolicy_binding/name_value<String>?**filter=property-name1:property-value1,property-name2:property-value2**

Use this query-parameter to get the filtered set of lbvserver_cmppolicy_binding resources configured on Citrix ADC. Filtering can be done on any of the properties of the resource.

pagination

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/lbvserver_cmppolicy_binding/name_value<String>?**pagesize=#no;pageno=#no**

Use this query-parameter to get the lbvserver_cmppolicy_binding resources in chunks.

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "lbvserver_cmppolicy_binding": [ {

"priority":<Double_value>,

"bindpoint":<String_value>,

"policyname":<String_value>,

"labelname":<String_value>,

"name":<String_value>,

"gotopriorityexpression":<String_value>,

"invoke":<Boolean_value>,

"labeltype":<String_value>,

"sc":<String_value>

}]}
```

get (all)

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/lbvserver_cmppolicy_binding

Query-parameters:

bulkbindings

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/lbvserver_cmppolicy_binding?**bulkbindings=yes**

NITRO allows you to fetch bindings in bulk.

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "lbvserver_cmppolicy_binding": [ {

"priority":<Double_value>,

"bindpoint":<String_value>,

"policyname":<String_value>,

"labelname":<String_value>,

"name":<String_value>,

"gotopriorityexpression":<String_value>,

"invoke":<Boolean_value>,

"labeltype":<String_value>,

"sc":<String_value>

}]}
```

count

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/lbvserver_cmppolicy_binding/name_value<String>?**count=yes**

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{"lbvserver_cmppolicy_binding": [ { "__count": "#no"} ] }
```

lbvserver_csvserver_binding

Binding object showing the csvserver that can be bound to lbvserver.

Properties

(click to see Operations)

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
priority	<Double>	Read-write	Priority.
policyname	<String>	Read-write	Name of the policy bound to the LB vserver.

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
name	<String>	Read-write	Name for the virtual server. Must begin with an ASCII alphanumeric or underscore (_) character, and must contain only ASCII alphanumeric, underscore, hash (#), period (.), space, colon (:), at sign (@), equal sign (=), and hyphen (-) characters. Can be changed after the virtual server is created. CLI Users: If the name includes one or more spaces, enclose the name in double or single quotation marks (for example, "my vserver" or 'my vserver'). . Minimum length = 1
cachevserver	<String>	Read-write	Cache virtual server.
policysubtype	<Double>	Read-only	.
cachetype	<String>	Read-only	Cache type. Possible values = TRANSPARENT, REVERSE, FORWARD
hits	<Double>	Read-only	Number of hits.
pipolicyhits	<Double>	Read-only	Number of hits.

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
__count	<Double>	Read-only	count parameter

Operations

(click to see Properties)

GET| GET (ALL)| COUNT

Some options that you can use for each operations:

- **Getting warnings in response:**NITRO allows you to get warnings in an operation by specifying the "warning" query parameter as "yes". For example, to get warnings while connecting to the Citrix ADC appliance, the URL is as follows:

  http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/login?warning=yes

  If any, the warnings are displayed in the response payload with the HTTP code "209 X-NITRO-WARNING".
- **Authenticated access for individual NITRO operations:**NITRO allows you to logon to the Citrix ADC appliance to perform individual operations. You can use this option instead of creating a NITRO session (using the login object) and then using that session to perform all operations,

  To do this, you must specify the username and password in the request header of the NITRO request as follows:

  X-NITRO-USER:<username>

  X-NITRO-PASS:<password>

  Note: In such cases, make sure that the request header DOES not include the following:

  Cookie:NITRO_AUTH_TOKEN=<tokenvalue>

`*Note:`*

Mandatory parameters are marked in redand placeholder content is marked in <green>.

get

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/lbvserver_csvserver_binding/name_value<String>

Query-parameters:

filter

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/lbvserver_csvserver_binding/name_value<String>?**filter=property-name1:property-value1,property-name2:property-value2**

Use this query-parameter to get the filtered set of lbvserver_csvserver_binding resources configured on Citrix ADC. Filtering can be done on any of the properties of the resource.

pagination

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/lbvserver_csvserver_binding/name_value<String>?**pagesize=#no;pageno=#no**

Use this query-parameter to get the lbvserver_csvserver_binding resources in chunks.

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "lbvserver_csvserver_binding": [ {

"priority":<Double_value>,

"policyname":<String_value>,

"name":<String_value>,

"cachevserver":<String_value>,
```

```
"policysubtype":<Double_value>,

"cachetype":<String_value>,

"hits":<Double_value>,

"pipolicyhits":<Double_value>

}]}
```

get (all)

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/lbvserver_csvserver_binding

Query-parameters:

bulkbindings

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/lbvserver_csvserver_binding?**bulkbindings=yes**

NITRO allows you to fetch bindings in bulk.

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "lbvserver_csvserver_binding": [ {
```

```
"priority":<Double_value>,

"policyname":<String_value>,

"name":<String_value>,

"cachevserver":<String_value>,

"policysubtype":<Double_value>,

"cachetype":<String_value>,

"hits":<Double_value>,

"pipolicyhits":<Double_value>

}]}
```

count

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/lbvserver_csvserver_binding/name_value<String>?**count=yes**

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{"lbvserver_csvserver_binding": [ { "__count": "#no"} ] }
```

lbvserver_dnspolicy64_binding

Binding object showing the dnspolicy64 that can be bound to lbvserver.

Properties

(click to see Operations)

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
priority	<Double>	Read-write	Priority.
bindpoint	<String>	Read-write	Bind point to which to bind the policy. Applicable only to compression, rewrite, videooptimization and cache policies. Possible values = REQUEST, RESPONSE
policyname	<String>	Read-write	Name of the policy bound to the LB vserver.
labelname	<String>	Read-write	Name of the virtual server or user-defined policy label to invoke if the policy evaluates to TRUE.
name	<String>	Read-write	Name for the virtual server. Must begin with an ASCII alphanumeric or underscore (_) character, and must contain only ASCII alphanumeric,

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
			underscore, hash (#), period (.), space, colon (:), at sign (@), equal sign (=), and hyphen (-) characters. Can be changed after the virtual server is created. CLI Users: If the name includes one or more spaces, enclose the name in double or single quotation marks (for example, "my vserver" or 'my vserver'). . Minimum length = 1
gotopriorityexpression	<String>	Read-write	Expression specifying the priority of the next policy which will get evaluated if the current policy rule evaluates to TRUE.
invoke	<Boolean>	Read-write	Invoke policies bound to a virtual server or policy label.
labeltype	<String>	Read-write	Type of policy label to invoke. Applicable only to rewrite, videooptimization and cache policies. Available settings function as follows: * reqvserver - Evaluate the request against the request-

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
			based policies bound to the specified virtual server. * resvserver - Evaluate the response against the response-based policies bound to the specified virtual server. * policylabel - invoke the request or response against the specified user-defined policy label. Possible values = reqvserver, resvserver, policylabel
__count	<Double>	Read-write	count parameter

Operations

(click to see Properties)

ADD:| DELETE:| GET| GET (ALL)| COUNT

Some options that you can use for each operations:

- **Getting warnings in response:**NITRO allows you to get warnings in an operation by specifying the "warning" query parameter as "yes". For example, to get warnings while connecting to the Citrix ADC appliance, the URL is as follows:

  http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/login?warning=yes

  If any, the warnings are displayed in the response payload with the HTTP code "209 X-NITRO-WARNING".

- **Authenticated access for individual NITRO operations:**NITRO allows you to logon to the Citrix ADC appliance to perform individual operations. You can use this option instead of creating a NITRO session (using the login object) and then using that session to perform all operations,

To do this, you must specify the username and password in the request header of the NITRO request as follows:

X-NITRO-USER:<username>

X-NITRO-PASS:<password>

Note: In such cases, make sure that the request header DOES not include the following:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue>

`*Note:`*

Mandatory parameters are marked in redand placeholder content is marked in <green>.

add

URL:http://<Citrix-ADC-IP-address(NSIP)/nitro/v1/config/lbvserver_dnspolicy64_binding

HTTP Method:PUT

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Content-Type:application/json

Request Payload:

```
{

"lbvserver_dnspolicy64_binding":{

<b>"name":<String_value>,

</b>"policyname":<String_value>,

"priority":<Double_value>,

"gotopriorityexpression":<String_value>,

"bindpoint":<String_value>,

"invoke":<Boolean_value>,
```

```
"labeltype":<String_value>,

"labelname":<String_value>

}}
```

Response:

HTTP Status Code on Success: 201 Created HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error

delete

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/lbvserver_dnspolicy64_binding/name_value<String>

Query-parameters:

args

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/lbvserver_dnspolicy64_binding/name_value<String>?
args=policyname:<String_value>,bindpoint:<String_value>,priority:<Double_value>

HTTP Method:DELETE

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue>

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error

get

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/lbvserver_dnspolicy64_binding/name_value<String>

Query-parameters:

filter

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/lbvserver_dnspolicy64_binding/name_value<String>?**filter=property-name1:property-value1,property-name2:property-value2**

Use this query-parameter to get the filtered set of lbvserver_dnspolicy64_binding resources configured on Citrix ADC. Filtering can be done on any of the properties of the resource.

pagination

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/lbvserver_dnspolicy64_binding/name_value<String>?**pagesize=#no;pageno=#no**

Use this query-parameter to get the lbvserver_dnspolicy64_binding resources in chunks.

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "lbvserver_dnspolicy64_binding": [ {

"priority":<Double_value>,

"bindpoint":<String_value>,

"policyname":<String_value>,

"labelname":<String_value>,

"name":<String_value>,
```

```
"gotopriorityexpression":<String_value>,

"invoke":<Boolean_value>,

"labeltype":<String_value>

}]}
```

get (all)

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/lbvserver_dnspolicy64_binding

Query-parameters:

bulkbindings

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/lbvserver_dnspolicy64_binding?**bulkbindings=yes**

NITRO allows you to fetch bindings in bulk.

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "lbvserver_dnspolicy64_binding": [ {

"priority":<Double_value>,
```

```
"bindpoint":<String_value>,

"policyname":<String_value>,

"labelname":<String_value>,

"name":<String_value>,

"gotopriorityexpression":<String_value>,

"invoke":<Boolean_value>,

"labeltype":<String_value>

}]}
```

count

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/lbvserver_dnspolicy64_binding/name_value<String>?**count=yes**

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{"lbvserver_dnspolicy64_binding": [ { "__count": "#no"} ] }
```

lbvserver_dospolicy_binding

Binding object showing the dospolicy that can be bound to lbvserver.

Properties

(click to see Operations)

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
priority	<Double>	Read-write	Priority.
policyname	<String>	Read-write	Name of the policy bound to the LB vserver.
name	<String>	Read-write	Name for the virtual server. Must begin with an ASCII alphanumeric or underscore (_) character, and must contain only ASCII alphanumeric, underscore, hash (#), period (.), space, colon (:), at sign (@), equal sign (=), and hyphen (-) characters. Can be changed after the virtual server is created. CLI Users: If the name includes one or more spaces, enclose the name in double or single quotation marks (for example, "my vserver" or 'my vserver'). . Minimum length = 1
__count	<Double>	Read-write	count parameter

Operations

(click to see Properties)

GET| GET (ALL)| COUNT

Some options that you can use for each operations:

- **Getting warnings in response:**NITRO allows you to get warnings in an operation by specifying the "warning" query parameter as "yes". For example, to get warnings while connecting to the Citrix ADC appliance, the URL is as follows:

  http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/login?warning=yes

  If any, the warnings are displayed in the response payload with the HTTP code "209 X-NITRO-WARNING".

- **Authenticated access for individual NITRO operations:**NITRO allows you to logon to the Citrix ADC appliance to perform individual operations. You can use this option instead of creating a NITRO session (using the login object) and then using that session to perform all operations,

  To do this, you must specify the username and password in the request header of the NITRO request as follows:

  X-NITRO-USER:<username>

  X-NITRO-PASS:<password>

  Note: In such cases, make sure that the request header DOES not include the following:

  Cookie:NITRO_AUTH_TOKEN=<tokenvalue>

`*Note:`*

Mandatory parameters are marked in redand placeholder content is marked in <green>.

get

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/lbvserver_dospolicy_binding/name_value<String>

Query-parameters:

filter

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/lbvserver_dospolicy_binding/name_value<String>?**filter=property-name1:property-value1,property-name2:property-value2**

Use this query-parameter to get the filtered set of lbvserver_dospolicy_binding resources configured on Citrix ADC. Filtering can be done on any of the properties of the resource.

pagination

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/lbvserver_dospolicy_binding/name_value<String>?**pagesize=#no;pageno=#no**

Use this query-parameter to get the lbvserver_dospolicy_binding resources in chunks.

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "lbvserver_dospolicy_binding": [ {

"priority":<Double_value>,

"policyname":<String_value>,

"name":<String_value>

}]}
```

get (all)

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/lbvserver_dospolicy_binding

Query-parameters:

bulkbindings

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/lbvserver_dospolicy_binding?**bulkbindings=yes**

NITRO allows you to fetch bindings in bulk.

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "lbvserver_dospolicy_binding": [ {

"priority":<Double_value>,

"policyname":<String_value>,

"name":<String_value>

}]}
```

count

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/lbvserver_dospolicy_binding/name_value<String>?**count=yes**

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{"lbvserver_dospolicy_binding": [ { "__count": "#no"} ] }
```

lbvserver_feopolicy_binding

Binding object showing the feopolicy that can be bound to lbvserver.

Properties

(click to see Operations)

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
priority	<Double>	Read-write	Priority.
bindpoint	<String>	Read-write	The bindpoint to which the policy is bound. Possible values = REQUEST, RESPONSE

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
policyname	<String>	Read-write	Name of the policy bound to the LB vserver.
labelname	<String>	Read-write	Name of the virtual server or user-defined policy label to invoke if the policy evaluates to TRUE.
name	<String>	Read-write	Name for the virtual server. Must begin with an ASCII alphanumeric or underscore (_) character, and must contain only ASCII alphanumeric, underscore, hash (#), period (.), space, colon (:), at sign (@), equal sign (=), and hyphen (-) characters. Can be changed after the virtual server is created. CLI Users: If the name includes one or more spaces, enclose the name in double or single quotation marks (for example, "my vserver" or 'my vserver'). . Minimum length = 1
gotopriorityexpression	<String>	Read-write	Expression specifying the priority of the next policy which will get

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
			evaluated if the current policy rule evaluates to TRUE.
invoke	<Boolean>	Read-write	Invoke policies bound to a virtual server or policy label.
labeltype	<String>	Read-write	Type of policy label to invoke. Applicable only to rewrite, videooptimization and cache policies. Available settings function as follows: * reqvserver - Evaluate the request against the request-based policies bound to the specified virtual server. * resvserver - Evaluate the response against the response-based policies bound to the specified virtual server. * policylabel - invoke the request or response against the specified user-defined policy label. Possible values = reqvserver, resvserver, policylabel
__count	<Double>	Read-write	count parameter

Operations

(click to see Properties)

ADD:| DELETE:| GET| GET (ALL)| COUNT

Some options that you can use for each operations:

- **Getting warnings in response:**NITRO allows you to get warnings in an operation by specifying the "warning" query parameter as "yes". For example, to get warnings while connecting to the Citrix ADC appliance, the URL is as follows:

  http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/login?warning=yes

  If any, the warnings are displayed in the response payload with the HTTP code "209 X-NITRO-WARNING".

- **Authenticated access for individual NITRO operations:**NITRO allows you to logon to the Citrix ADC appliance to perform individual operations. You can use this option instead of creating a NITRO session (using the login object) and then using that session to perform all operations,

  To do this, you must specify the username and password in the request header of the NITRO request as follows:

  X-NITRO-USER:<username>

  X-NITRO-PASS:<password>

  Note: In such cases, make sure that the request header DOES not include the following:

  Cookie:NITRO_AUTH_TOKEN=<tokenvalue>

`*Note:`*

Mandatory parameters are marked in redand placeholder content is marked in <green>.

add

URL:http://<Citrix-ADC-IP-address(NSIP)/nitro/v1/config/lbvserver_feopolicy_binding

HTTP Method:PUT

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Content-Type:application/json

Request Payload:

```
{

"lbvserver_feopolicy_binding":{

<b>"name":<String_value>,

</b>"policyname":<String_value>,

"priority":<Double_value>,

"gotopriorityexpression":<String_value>,

"bindpoint":<String_value>,

"invoke":<Boolean_value>,

"labeltype":<String_value>,

"labelname":<String_value>

}}
```

Response:

HTTP Status Code on Success: 201 Created HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error

delete

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/lbvserver_feopolicy_binding/name_value<String>

Query-parameters:

args

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/lbvserver_feopolicy_binding/name_value<String>?
args=policyname:<String_value>,bindpoint:<String_value>,priority:<Double_value>

HTTP Method:DELETE

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue>

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error

get

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/lbvserver_feopolicy_binding/name_value<String>

Query-parameters:

filter

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/lbvserver_feopolicy_binding/name_value<String>?**filter=property-name1:property-value1,property-name2:property-value2**

Use this query-parameter to get the filtered set of lbvserver_feopolicy_binding resources configured on Citrix ADC. Filtering can be done on any of the properties of the resource.

pagination

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/lbvserver_feopolicy_binding/name_value<String>?**pagesize=#no;pageno=#no**

Use this query-parameter to get the lbvserver_feopolicy_binding resources in chunks.

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "lbvserver_feopolicy_binding": [ {

"priority":<Double_value>,

"bindpoint":<String_value>,

"policyname":<String_value>,

"labelname":<String_value>,

"name":<String_value>,

"gotopriorityexpression":<String_value>,

"invoke":<Boolean_value>,

"labeltype":<String_value>

}]}
```

get (all)

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/lbvserver_feopolicy_binding

Query-parameters:

bulkbindings

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/lbvserver_feopolicy_binding?**bulkbindings=yes**

NITRO allows you to fetch bindings in bulk.

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "lbvserver_feopolicy_binding": [ {

"priority":<Double_value>,

"bindpoint":<String_value>,

"policyname":<String_value>,

"labelname":<String_value>,

"name":<String_value>,

"gotopriorityexpression":<String_value>,

"invoke":<Boolean_value>,

"labeltype":<String_value>

}]}
```

count

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/lbvserver_feopolicy_binding/name_value<String>?**count=yes**

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{"lbvserver_feopolicy_binding": [ { "__count": "#no"} ] }
```

lbvserver_filterpolicy_binding

Binding object showing the filterpolicy that can be bound to lbvserver.

Properties

(click to see Operations)

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
priority	<Double>	Read-write	Priority.
bindpoint	<String>	Read-write	Bind point to which to bind the policy. Applicable only to compression, rewrite, videooptimization and cache policies. Possible values = REQUEST, RESPONSE
policyname	<String>	Read-write	Name of the policy bound to the LB vserver.

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
labelname	<String>	Read-write	Name of the virtual server or user-defined policy label to invoke if the policy evaluates to TRUE.
name	<String>	Read-write	Name for the virtual server. Must begin with an ASCII alphanumeric or underscore (_) character, and must contain only ASCII alphanumeric, underscore, hash (#), period (.), space, colon (:), at sign (@), equal sign (=), and hyphen (-) characters. Can be changed after the virtual server is created. CLI Users: If the name includes one or more spaces, enclose the name in double or single quotation marks (for example, "my vserver" or 'my vserver'). . Minimum length = 1
gotopriorityexpression	<String>	Read-write	Expression or other value specifying the next policy to be evaluated if the current policy evaluates to TRUE. Specify one of the following values: *

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
			NEXT - Evaluate the policy with the next higher priority number. * END - End policy evaluation. * USE_INVOCATION_RESULT - Applicable if this policy invokes another policy label. If the final goto in the invoked policy label has a value of END, the evaluation stops. If the final goto is anything other than END, the current policy label performs a NEXT. * An expression that evaluates to a number. If you specify an expression, the number to which it evaluates determines the next policy to evaluate, as follows: * If the expression evaluates to a higher numbered priority, the policy with that priority is evaluated next. * If the expression evaluates to the priority of the current policy, the policy with the next higher numbered priority is evaluated next. * If the

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
			expression evaluates to a priority number that is numerically higher than the highest numbered priority, policy evaluation ends. An UNDEF event is triggered if: * The expression is invalid. * The expression evaluates to a priority number that is numerically lower than the current policy's priority. * The expression evaluates to a priority number that is between the current policy's priority number (say, 30) and the highest priority number (say, 100), but does not match any configured priority number (for example, the expression evaluates to the number 85). This example assumes that the priority number increments by 10 for every successive policy, and therefore a priority number of 85 does not exist in the policy label.

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
invoke	<Boolean>	Read-write	Invoke policies bound to a virtual server or policy label.
labeltype	<String>	Read-write	Type of policy label to invoke. Applicable only to rewrite, videooptimization and cache policies. Available settings function as follows: * reqvserver - Evaluate the request against the request-based policies bound to the specified virtual server. * resvserver - Evaluate the response against the response-based policies bound to the specified virtual server. * policylabel - invoke the request or response against the specified user-defined policy label. Possible values = reqvserver, resvserver, policylabel
sc	<String>	Read-only	Use SureConnect on the virtual server. Default value: OFF Possible values = ON, OFF
__count	<Double>	Read-only	count parameter

Operations

(click to see Properties)

ADD:| DELETE:| GET| GET (ALL)| COUNT

Some options that you can use for each operations:

- **Getting warnings in response:**NITRO allows you to get warnings in an operation by specifying the "warning" query parameter as "yes". For example, to get warnings while connecting to the Citrix ADC appliance, the URL is as follows:

  http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/login?warning=yes

  If any, the warnings are displayed in the response payload with the HTTP code "209 X-NITRO-WARNING".

- **Authenticated access for individual NITRO operations:**NITRO allows you to logon to the Citrix ADC appliance to perform individual operations. You can use this option instead of creating a NITRO session (using the login object) and then using that session to perform all operations,

  To do this, you must specify the username and password in the request header of the NITRO request as follows:

  X-NITRO-USER:<username>

  X-NITRO-PASS:<password>

  Note: In such cases, make sure that the request header DOES not include the following:

  Cookie:NITRO_AUTH_TOKEN=<tokenvalue>

`*Note:`*

Mandatory parameters are marked in redand placeholder content is marked in <green>.

add

URL:http://<Citrix-ADC-IP-address(NSIP)/nitro/v1/config/lbvserver_filterpolicy_binding

HTTP Method:PUT

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Content-Type:application/json

Request Payload:

```
{
"lbvserver_filterpolicy_binding":{
<b>"name":<String_value>,
</b>"policyname":<String_value>,
"priority":<Double_value>,
"gotopriorityexpression":<String_value>,
"bindpoint":<String_value>,
"invoke":<Boolean_value>,
"labeltype":<String_value>,
"labelname":<String_value>
}}
```

Response:

HTTP Status Code on Success: 201 Created HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error

delete

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/lbvserver_filterpolicy_binding/name_value<String>

Query-parameters:

args

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/lbvserver_filterpolicy_binding/name_value<String>?**args=policyname:<String_value>,bindpoint:<String_value>,priority:<Double_value>**

HTTP Method:DELETE

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue>

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error

get

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/lbvserver_filterpolicy_binding/name_value<String>

Query-parameters:

filter

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/lbvserver_filterpolicy_binding/name_value<String>?**filter=property-name1:property-value1,property-name2:property-value2**

Use this query-parameter to get the filtered set of lbvserver_filterpolicy_binding resources configured on Citrix ADC. Filtering can be done on any of the properties of the resource.

pagination

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/lbvserver_filterpolicy_binding/name_value<String>?**pagesize=#no;pageno=#no**

Use this query-parameter to get the lbvserver_filterpolicy_binding resources in chunks.

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "lbvserver_filterpolicy_binding": [ {

"priority":<Double_value>,

"bindpoint":<String_value>,

"policyname":<String_value>,

"labelname":<String_value>,

"name":<String_value>,

"gotopriorityexpression":<String_value>,

"invoke":<Boolean_value>,

"labeltype":<String_value>,

"sc":<String_value>

}]}
```

get (all)

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/lbvserver_filterpolicy_binding

Query-parameters:

bulkbindings

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/lbvserver_filterpolicy_binding?**bulkbindings=yes**

NITRO allows you to fetch bindings in bulk.

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "lbvserver_filterpolicy_binding": [ {

"priority":<Double_value>,

"bindpoint":<String_value>,

"policyname":<String_value>,

"labelname":<String_value>,

"name":<String_value>,

"gotopriorityexpression":<String_value>,

"invoke":<Boolean_value>,

"labeltype":<String_value>,

"sc":<String_value>

}]}
```

count

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/lbvserver_filterpolicy_binding/name_value<String>?**count=yes**

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{"lbvserver_filterpolicy_binding": [ { "__count": "#no"} ] }
```

lbvserver_pqpolicy_binding

Binding object showing the pqpolicy that can be bound to lbvserver.

Properties

(click to see Operations)

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
priority	<Double>	Read-write	Integer specifying the policy's priority. The lower the priority number, the higher the policy's priority. Minimum value = 1

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
			Maximum value = 2147483647
bindpoint	<String>	Read-write	Bind point to which to bind the policy. Applicable only to compression, rewrite, videooptimization and cache policies. Possible values = REQUEST, RESPONSE
policyname	<String>	Read-write	Name of the policy bound to the LB vserver.
labelname	<String>	Read-write	Name of the virtual server or user-defined policy label to invoke if the policy evaluates to TRUE.
name	<String>	Read-write	Name for the virtual server. Must begin with an ASCII alphanumeric or underscore (_) character, and must contain only ASCII alphanumeric, underscore, hash (#), period (.), space, colon (:), at sign (@), equal sign (=), and hyphen (-) characters. Can be changed after the virtual server is created. CLI Users: If

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
			the name includes one or more spaces, enclose the name in double or single quotation marks (for example, "my vserver" or 'my vserver'). . Minimum length = 1
gotopriorityexpression	<String>	Read-write	Expression or other value specifying the next policy to be evaluated if the current policy evaluates to TRUE. Specify one of the following values: * NEXT - Evaluate the policy with the next higher priority number. * END - End policy evaluation. * USE_INVOCATION_RESULT - Applicable if this policy invokes another policy label. If the final goto in the invoked policy label has a value of END, the evaluation stops. If the final goto is anything other than END, the current policy label performs a NEXT. * An expression that evaluates to a number. If you specify an expression, the

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
			number to which it evaluates determines the next policy to evaluate, as follows: * If the expression evaluates to a higher numbered priority, the policy with that priority is evaluated next. * If the expression evaluates to the priority of the current policy, the policy with the next higher numbered priority is evaluated next. * If the expression evaluates to a priority number that is numerically higher than the highest numbered priority, policy evaluation ends. An UNDEF event is triggered if: * The expression is invalid. * The expression evaluates to a priority number that is numerically lower than the current policy's priority. * The expression evaluates to a priority number that is between the current policy's priority number (say, 30) and the highest priority number (say,

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
			100), but does not match any configured priority number (for example, the expression evaluates to the number 85). This example assumes that the priority number increments by 10 for every successive policy, and therefore a priority number of 85 does not exist in the policy label.
invoke	<Boolean>	Read-write	Invoke policies bound to a virtual server or policy label.
labeltype	<String>	Read-write	Type of policy label to invoke. Applicable only to rewrite, videooptimization and cache policies. Available settings function as follows: * reqvserver - Evaluate the request against the request-based policies bound to the specified virtual server. * resvserver - Evaluate the response against the response-based policies bound to the specified virtual server. * policylabel - invoke the request or

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
			response against the specified user-defined policy label. Possible values = reqvserver, resvserver, policylabel
__count	<Double>	Read-write	count parameter

Operations

(click to see Properties)

ADD:| DELETE:| GET| GET (ALL)| COUNT

Some options that you can use for each operations:

- **Getting warnings in response:**NITRO allows you to get warnings in an operation by specifying the "warning" query parameter as "yes". For example, to get warnings while connecting to the Citrix ADC appliance, the URL is as follows:

  http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/login?warning=yes

  If any, the warnings are displayed in the response payload with the HTTP code "209 X-NITRO-WARNING".

- **Authenticated access for individual NITRO operations:**NITRO allows you to logon to the Citrix ADC appliance to perform individual operations. You can use this option instead of creating a NITRO session (using the login object) and then using that session to perform all operations,

  To do this, you must specify the username and password in the request header of the NITRO request as follows:

  X-NITRO-USER:<username>

  X-NITRO-PASS:<password>

  Note: In such cases, make sure that the request header DOES not include the following:

  Cookie:NITRO_AUTH_TOKEN=<tokenvalue>

`*Note:` *

Mandatory parameters are marked in redand placeholder content is marked in <green>.

add

URL:http://<Citrix-ADC-IP-address(NSIP)/nitro/v1/config/lbvserver_pqpolicy_binding

HTTP Method:PUT

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Content-Type:application/json

Request Payload:

```
{

"lbvserver_pqpolicy_binding":{

<b>"name":<String_value>,

</b>"policyname":<String_value>,

"priority":<Double_value>,

"gotopriorityexpression":<String_value>,

"bindpoint":<String_value>,

"invoke":<Boolean_value>,

"labeltype":<String_value>,

"labelname":<String_value>

}}
```

Response:

HTTP Status Code on Success: 201 Created HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error

delete

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/lbvserver_pqpolicy_binding/name_value<String>

Query-parameters:

args

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/lbvserver_pqpolicy_binding/name_value<String>?**args=policyname:<String_value>,bindpoint:<String_value>,priority:<Double_value>**

HTTP Method:DELETE

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue>

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error

get

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/lbvserver_pqpolicy_binding/name_value<String>

Query-parameters:

filter

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/lbvserver_pqpolicy_binding/name_value<String>?**filter=property-name1:property-value1,property-name2:property-value2**

Use this query-parameter to get the filtered set of lbvserver_pqpolicy_binding resources configured on Citrix ADC. Filtering can be done on any of the properties of the resource.

pagination

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/lbvserver_pqpolicy_binding/name_value<String>?**pagesize=#no;pageno=#no**

Use this query-parameter to get the lbvserver_pqpolicy_binding resources in chunks.

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "lbvserver_pqpolicy_binding": [ {

"priority":<Double_value>,

"bindpoint":<String_value>,

"policyname":<String_value>,

"labelname":<String_value>,

"name":<String_value>,

"gotopriorityexpression":<String_value>,

"invoke":<Boolean_value>,

"labeltype":<String_value>

}]}
```

get (all)

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/lbvserver_pqpolicy_binding

Query-parameters:

bulkbindings

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/lbvserver_pqpolicy_binding?**bulkbindings=yes**

NITRO allows you to fetch bindings in bulk.

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "lbvserver_pqpolicy_binding": [ {

"priority":<Double_value>,

"bindpoint":<String_value>,

"policyname":<String_value>,

"labelname":<String_value>,

"name":<String_value>,

"gotopriorityexpression":<String_value>,
```

```
"invoke":<Boolean_value>,

"labeltype":<String_value>

}]}
```

count

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/lbvserver_pqpolicy_binding/name_value<String>?**count=yes**

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{"lbvserver_pqpolicy_binding": [ { "__count": "#no"} ] }
```

lbvserver_responderpolicy_binding

Binding object showing the responderpolicy that can be bound to lbvserver.

Properties

(click to see Operations)

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
priority	<Double>	Read-write	Priority.
bindpoint	<String>	Read-write	Bind point to which to bind the policy. Applicable only to compression, rewrite, videooptimization and cache policies. Possible values = REQUEST, RESPONSE
policyname	<String>	Read-write	Name of the policy bound to the LB vserver.
labelname	<String>	Read-write	Name of the label invoked.
name	<String>	Read-write	Name for the virtual server. Must begin with an ASCII alphanumeric or underscore (_) character, and must contain only ASCII alphanumeric, underscore, hash (#), period (.), space, colon (:), at sign (@), equal sign (=), and hyphen (-) characters. Can be changed after the virtual server is created. CLI Users: If the name includes one or more spaces, enclose the name in double or single

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
			quotation marks (for example, "my vserver" or 'my vserver'). . Minimum length = 1
gotopriorityexpression	<String>	Read-write	Expression specifying the priority of the next policy which will get evaluated if the current policy rule evaluates to TRUE.
invoke	<Boolean>	Read-write	Invoke policies bound to a virtual server or policy label.
labeltype	<String>	Read-write	The invocation type. Possible values = reqvserver, resvserver, policylabel
__count	<Double>	Read-write	count parameter

Operations

(click to see Properties)

ADD:| DELETE:| GET| GET (ALL)| COUNT

Some options that you can use for each operations:

- **Getting warnings in response:**NITRO allows you to get warnings in an operation by specifying the "warning" query parameter as "yes". For example, to get warnings while connecting to the Citrix ADC appliance, the URL is as follows:

  http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/login?warning=yes

If any, the warnings are displayed in the response payload with the HTTP code "209 X-NITRO-WARNING".

- **Authenticated access for individual NITRO operations:**NITRO allows you to logon to the Citrix ADC appliance to perform individual operations. You can use this option instead of creating a NITRO session (using the login object) and then using that session to perform all operations,

  To do this, you must specify the username and password in the request header of the NITRO request as follows:

  X-NITRO-USER:<username>

  X-NITRO-PASS:<password>

  Note: In such cases, make sure that the request header DOES not include the following:

  Cookie:NITRO_AUTH_TOKEN=<tokenvalue>

`*Note:`*

Mandatory parameters are marked in redand placeholder content is marked in <green>.

add

URL:http://<Citrix-ADC-IP-address(NSIP)/nitro/v1/config/lbvserver_responderpolicy_binding

HTTP Method:PUT

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Content-Type:application/json

Request Payload:

```
{

"lbvserver_responderpolicy_binding":{

<b>"name":<String_value>,

</b>"policyname":<String_value>,
```

```
"priority":<Double_value>,

"gotopriorityexpression":<String_value>,

"bindpoint":<String_value>,

"invoke":<Boolean_value>,

"labeltype":<String_value>,

"labelname":<String_value>

}}
```

Response:

HTTP Status Code on Success: 201 Created HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error

delete

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/lbvserver_responderpolicy_binding/name_value<String>

Query-parameters:

args

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/lbvserver_responderpolicy_binding/name_value<String>?
args=policyname:<String_value>,bindpoint:<String_value>,priority:<Double_value>

HTTP Method:DELETE

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue>

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error

get

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/lbvserver_responderpolicy_binding/name_value<String>

Query-parameters:

filter

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/lbvserver_responderpolicy_binding/name_value<String>?**filter=property-name1:property-value1,property-name2:property-value2**

Use this query-parameter to get the filtered set of lbvserver_responderpolicy_binding resources configured on Citrix ADC. Filtering can be done on any of the properties of the resource.

pagination

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/lbvserver_responderpolicy_binding/name_value<String>?**pagesize=#no;pageno=#no**

Use this query-parameter to get the lbvserver_responderpolicy_binding resources in chunks.

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "lbvserver_responderpolicy_binding": [ {
```

```
"priority":<Double_value>,

"bindpoint":<String_value>,

"policyname":<String_value>,

"labelname":<String_value>,

"name":<String_value>,

"gotopriorityexpression":<String_value>,

"invoke":<Boolean_value>,

"labeltype":<String_value>

}]}
```

get (all)

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/lbvserver_responderpolicy_binding

Query-parameters:

bulkbindings

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/lbvserver_responderpolicy_binding?**bulkbindings=yes**

NITRO allows you to fetch bindings in bulk.

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "lbvserver_responderpolicy_binding": [ {

"priority":<Double_value>,

"bindpoint":<String_value>,

"policyname":<String_value>,

"labelname":<String_value>,

"name":<String_value>,

"gotopriorityexpression":<String_value>,

"invoke":<Boolean_value>,

"labeltype":<String_value>

}]}
```

count

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/lbvserver_responderpolicy_binding/name_value<String>?**count=yes**

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{"lbvserver_responderpolicy_binding": [ { "__count": "#no"} ] }
```

lbvserver_rewritepolicy_binding

Binding object showing the rewritepolicy that can be bound to lbvserver.

Properties

(click to see Operations)

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
priority	<Double>	Read-write	Priority.
bindpoint	<String>	Read-write	The bindpoint to which the policy is bound. Possible values = REQUEST, RESPONSE
policyname	<String>	Read-write	Name of the policy bound to the LB vserver.
labelname	<String>	Read-write	Name of the label invoked.
name	<String>	Read-write	Name for the virtual server. Must begin with an ASCII alphanumeric or underscore (_) character, and must contain only ASCII alphanumeric,

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
			underscore, hash (#), period (.), space, colon (:), at sign (@), equal sign (=), and hyphen (-) characters. Can be changed after the virtual server is created. CLI Users: If the name includes one or more spaces, enclose the name in double or single quotation marks (for example, "my vserver" or 'my vserver'). . Minimum length = 1
gotopriorityexpression	<String>	Read-write	Expression specifying the priority of the next policy which will get evaluated if the current policy rule evaluates to TRUE.
invoke	<Boolean>	Read-write	Invoke policies bound to a virtual server or policy label.
labeltype	<String>	Read-write	The invocation type. Possible values = reqvserver, resvserver, policylabel
__count	<Double>	Read-write	count parameter

Operations

(click to see Properties)

ADD:| DELETE:| GET| GET (ALL)| COUNT

Some options that you can use for each operations:

- **Getting warnings in response:**NITRO allows you to get warnings in an operation by specifying the "warning" query parameter as "yes". For example, to get warnings while connecting to the Citrix ADC appliance, the URL is as follows:

  http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/login?warning=yes

  If any, the warnings are displayed in the response payload with the HTTP code "209 X-NITRO-WARNING".

- **Authenticated access for individual NITRO operations:**NITRO allows you to logon to the Citrix ADC appliance to perform individual operations. You can use this option instead of creating a NITRO session (using the login object) and then using that session to perform all operations,

  To do this, you must specify the username and password in the request header of the NITRO request as follows:

  X-NITRO-USER:<username>

  X-NITRO-PASS:<password>

  Note: In such cases, make sure that the request header DOES not include the following:

  Cookie:NITRO_AUTH_TOKEN=<tokenvalue>

`*Note:`*

Mandatory parameters are marked in redand placeholder content is marked in <green>.

add

URL:http://<Citrix-ADC-IP-address(NSIP)/nitro/v1/config/lbvserver_rewritepolicy_binding

HTTP Method:PUT

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Content-Type:application/json

Request Payload:

```
{
"lbvserver_rewritepolicy_binding":{
<b>"name":<String_value>,
</b>"policyname":<String_value>,
"priority":<Double_value>,
"gotopriorityexpression":<String_value>,
"bindpoint":<String_value>,
"invoke":<Boolean_value>,
"labeltype":<String_value>,
"labelname":<String_value>
}}
```

Response:

HTTP Status Code on Success: 201 Created HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error

delete

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/lbvserver_rewritepolicy_binding/name_value<String>

Query-parameters:

args

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/lbvserver_rewritepolicy_binding/name_value<String>?**args=policyname:<String_value>,bindpoint:<String_value>,priority:<Double_value>**

HTTP Method:DELETE

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue>

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error

get

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/lbvserver_rewritepolicy_binding/name_value<String>

Query-parameters:

filter

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/lbvserver_rewritepolicy_binding/name_value<String>?**filter=property-name1:property-value1,property-name2:property-value2**

Use this query-parameter to get the filtered set of lbvserver_rewritepolicy_binding resources configured on Citrix ADC. Filtering can be done on any of the properties of the resource.

pagination

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/lbvserver_rewritepolicy_binding/name_value<String>?**pagesize=#no;pageno=#no**

Use this query-parameter to get the lbvserver_rewritepolicy_binding resources in chunks.

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "lbvserver_rewritepolicy_binding": [ {

"priority":<Double_value>,

"bindpoint":<String_value>,

"policyname":<String_value>,

"labelname":<String_value>,

"name":<String_value>,

"gotopriorityexpression":<String_value>,

"invoke":<Boolean_value>,

"labeltype":<String_value>

}]}
```

get (all)

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/lbvserver_rewritepolicy_binding

Query-parameters:

bulkbindings

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/lbvserver_rewritepolicy_binding?**bulkbindings=yes**

NITRO allows you to fetch bindings in bulk.

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "lbvserver_rewritepolicy_binding": [ {

"priority":<Double_value>,

"bindpoint":<String_value>,

"policyname":<String_value>,

"labelname":<String_value>,

"name":<String_value>,

"gotopriorityexpression":<String_value>,

"invoke":<Boolean_value>,

"labeltype":<String_value>

}]}
```

count

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/lbvserver_rewritepolicy_binding/name_value<String>?**count=yes**

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{"lbvserver_rewritepolicy_binding": [ { "__count": "#no"} ] }
```

lbvserver_scpolicy_binding

Binding object showing the scpolicy that can be bound to lbvserver.

Properties

(click to see Operations)

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
priority	<Double>	Read-write	Priority.
bindpoint	<String>	Read-write	Bind point to which to bind the policy. Applicable only to compression, rewrite, videooptimization and cache policies. Possible values = REQUEST, RESPONSE

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
policyname	<String>	Read-write	Name of the policy bound to the LB vserver.
labelname	<String>	Read-write	Name of the virtual server or user-defined policy label to invoke if the policy evaluates to TRUE.
name	<String>	Read-write	Name for the virtual server. Must begin with an ASCII alphanumeric or underscore (_) character, and must contain only ASCII alphanumeric, underscore, hash (#), period (.), space, colon (:), at sign (@), equal sign (=), and hyphen (-) characters. Can be changed after the virtual server is created. CLI Users: If the name includes one or more spaces, enclose the name in double or single quotation marks (for example, "my vserver" or 'my vserver'). . Minimum length = 1
gotopriorityexpression	<String>	Read-write	Expression or other value specifying the next policy to be evaluated if the

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
			current policy evaluates to TRUE. Specify one of the following values: * NEXT - Evaluate the policy with the next higher priority number. * END - End policy evaluation. * USE_INVOCATION_RESULT - Applicable if this policy invokes another policy label. If the final goto in the invoked policy label has a value of END, the evaluation stops. If the final goto is anything other than END, the current policy label performs a NEXT. * An expression that evaluates to a number. If you specify an expression, the number to which it evaluates determines the next policy to evaluate, as follows: * If the expression evaluates to a higher numbered priority, the policy with that priority is evaluated next. * If the expression evaluates to the priority of the current policy, the

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
			policy with the next higher numbered priority is evaluated next. * If the expression evaluates to a priority number that is numerically higher than the highest numbered priority, policy evaluation ends. An UNDEF event is triggered if: * The expression is invalid. * The expression evaluates to a priority number that is numerically lower than the current policy's priority. * The expression evaluates to a priority number that is between the current policy's priority number (say, 30) and the highest priority number (say, 100), but does not match any configured priority number (for example, the expression evaluates to the number 85). This example assumes that the priority number increments by 10 for every successive policy, and therefore a priority number of

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
			85 does not exist in the policy label.
invoke	<Boolean>	Read-write	Invoke policies bound to a virtual server or policy label.
labeltype	<String>	Read-write	Type of policy label to invoke. Applicable only to rewrite, videooptimization and cache policies. Available settings function as follows: * reqvserver - Evaluate the request against the request-based policies bound to the specified virtual server. * resvserver - Evaluate the response against the response-based policies bound to the specified virtual server. * policylabel - invoke the request or response against the specified user-defined policy label. Possible values = reqvserver, resvserver, policylabel
__count	<Double>	Read-write	count parameter

Operations

(click to see Properties)

ADD:| DELETE:| GET| GET (ALL)| COUNT

Some options that you can use for each operations:

- **Getting warnings in response:**NITRO allows you to get warnings in an operation by specifying the "warning" query parameter as "yes". For example, to get warnings while connecting to the Citrix ADC appliance, the URL is as follows:

  http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/login?warning=yes

  If any, the warnings are displayed in the response payload with the HTTP code "209 X-NITRO-WARNING".

- **Authenticated access for individual NITRO operations:**NITRO allows you to logon to the Citrix ADC appliance to perform individual operations. You can use this option instead of creating a NITRO session (using the login object) and then using that session to perform all operations,

  To do this, you must specify the username and password in the request header of the NITRO request as follows:

  X-NITRO-USER:<username>

  X-NITRO-PASS:<password>

  Note: In such cases, make sure that the request header DOES not include the following:

  Cookie:NITRO_AUTH_TOKEN=<tokenvalue>

`*Note:`*

Mandatory parameters are marked in redand placeholder content is marked in <green>.

add

URL:http://<Citrix-ADC-IP-address(NSIP)/nitro/v1/config/lbvserver_scpolicy_binding

HTTP Method:PUT

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Content-Type:application/json

Request Payload:

```
{

"lbvserver_scpolicy_binding":{

<b>"name":<String_value>,

</b>"policyname":<String_value>,

"priority":<Double_value>,

"gotopriorityexpression":<String_value>,

"bindpoint":<String_value>,

"invoke":<Boolean_value>,

"labeltype":<String_value>,

"labelname":<String_value>

}}
```

Response:

HTTP Status Code on Success: 201 Created HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error

delete

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/lbvserver_scpolicy_binding/name_value<String>

Query-parameters:

args

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/lbvserver_scpolicy_binding/name_value<String>?
args=policyname:<String_value>,bindpoint:<String_value>,priority:<Double_value>

HTTP Method:DELETE

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue>

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error

get

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/lbvserver_scpolicy_binding/name_value<String>

Query-parameters:

filter

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/lbvserver_scpolicy_binding/name_value<String>?**filter=property-name1:property-value1,property-name2:property-value2**

Use this query-parameter to get the filtered set of lbvserver_scpolicy_binding resources configured on Citrix ADC. Filtering can be done on any of the properties of the resource.

pagination

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/lbvserver_scpolicy_binding/name_value<String>?**pagesize=#no;pageno=#no**

Use this query-parameter to get the lbvserver_scpolicy_binding resources in chunks.

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "lbvserver_scpolicy_binding": [ {

"priority":<Double_value>,

"bindpoint":<String_value>,

"policyname":<String_value>,

"labelname":<String_value>,

"name":<String_value>,

"gotopriorityexpression":<String_value>,

"invoke":<Boolean_value>,

"labeltype":<String_value>

}]}
```

get (all)

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/lbvserver_scpolicy_binding

Query-parameters:

bulkbindings

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/lbvserver_scpolicy_binding?**bulkbindings=yes**

NITRO allows you to fetch bindings in bulk.

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "lbvserver_scpolicy_binding": [ {

"priority":<Double_value>,

"bindpoint":<String_value>,

"policyname":<String_value>,

"labelname":<String_value>,

"name":<String_value>,

"gotopriorityexpression":<String_value>,

"invoke":<Boolean_value>,

"labeltype":<String_value>

}]}
```

count

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/lbvserver_scpolicy_binding/name_value<String>?**count=yes**

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{"lbvserver_scpolicy_binding": [ { "__count": "#no"} ] }
```

lbvserver_servicegroupmember_binding

Binding object showing the servicegroupmember that can be bound to lbvserver.

Properties

(click to see Operations)

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
servicegroupname	<String>	Read-write	The service group name bound to the selected load balancing virtual server.
name	<String>	Read-write	Name for the virtual server. Must begin with an ASCII alphanumeric or underscore (_) character, and must contain only ASCII alphanumeric, underscore, hash (#), period (.), space,

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
			colon (:), at sign (@), equal sign (=), and hyphen (-) characters. Can be changed after the virtual server is created. CLI Users: If the name includes one or more spaces, enclose the name in double or single quotation marks (for example, "my vserver" or 'my vserver'). . Minimum length = 1
weight	<Double>	Read-write	Weight to assign to the specified service. Minimum value = 1 Maximum value = 100
cookieipport	<String>	Read-only	Encryped Ip address and port of the service that is inserted into the set-cookie http header.
port	<Integer>	Read-only	Port number for the virtual server. Range 1 - 65535 * in CLI is represented as 65535 in NITRO API

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
curstate	<String>	Read-only	Current LB vserver state. Possible values = UP, DOWN, UNKNOWN, BUSY, OUT OF SERVICE, GOING OUT OF SERVICE, DOWN WHEN GOING OUT OF SERVICE, NS_EMPTY_STR, Unknown, DISABLED
preferredlocation	<String>	Read-only	Used for displaying the location of bound services.
vserverid	<String>	Read-only	Vserver Id.
ipv46	<String>	Read-only	IPv4 or IPv6 address to assign to the virtual server.
cookiename	<String>	Read-only	Use this parameter to specify the cookie name for COOKIE peristence type. It specifies the name of cookie with a maximum of 32 characters. If not specified, cookie name is internally generated.
dynamicweight	<Double>	Read-only	Dynamic weight.

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
servicetype	<String>	Read-only	Protocol used by the service (also called the service type). Possible values = HTTP, FTP, TCP, UDP, SSL, SSL_BRIDGE, SSL_TCP, DTLS, NNTP, DNS, DHCPRA, ANY, SIP_UDP, SIP_TCP, SIP_SSL, DNS_TCP, RTSP, PUSH, SSL_PUSH, RADIUS, RDP, MYSQL, MSSQL, DIAMETER, SSL_DIAMETER, TFTP, ORACLE, SMPP, SYSLOGTCP, SYSLOGUDP, FIX, SSL_FIX, PROXY, USER_TCP, USER_SSL_TCP, QUIC, IPFIX, LOGSTREAM
__count	<Double>	Read-only	count parameter

Operations

(click to see Properties)

GET| GET (ALL)| COUNT

Some options that you can use for each operations:

- **Getting warnings in response:**NITRO allows you to get warnings in an operation by specifying the "warning" query parameter as "yes". For example, to get warnings while connecting to the Citrix ADC appliance, the URL is as follows:

  http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/login?warning=yes

  If any, the warnings are displayed in the response payload with the HTTP code "209 X-NITRO-WARNING".

- **Authenticated access for individual NITRO operations:**NITRO allows you to logon to the Citrix ADC appliance to perform individual operations. You can use this option instead of creating a NITRO session (using the login object) and then using that session to perform all operations,

  To do this, you must specify the username and password in the request header of the NITRO request as follows:

  X-NITRO-USER:<username>

  X-NITRO-PASS:<password>

  Note: In such cases, make sure that the request header DOES not include the following:

  Cookie:NITRO_AUTH_TOKEN=<tokenvalue>

`*Note:`*

Mandatory parameters are marked in redand placeholder content is marked in <green>.

get

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/lbvserver_servicegroupmember_binding/name_value<String>

Query-parameters:

filter

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/lbvserver_servicegroupmember_binding/name_value<String>?**filter=property-name1:property-value1,property-name2:property-value2**

Use this query-parameter to get the filtered set of lbvserver_servicegroupmember_binding resources configured on Citrix ADC. Filtering can be done on any of the properties of the resource.

pagination

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/lbvserver_servicegroupmember_binding/name_value<String>?**pagesize=#no;pageno=#no**

Use this query-parameter to get the lbvserver_servicegroupmember_binding resources in chunks.

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "lbvserver_servicegroupmember_binding": [ {

"servicegroupname":<String_value>,

"name":<String_value>,

"weight":<Double_value>,

"cookieipport":<String_value>,

"port":<Integer_value>,

"curstate":<String_value>,

"preferredlocation":<String_value>,

"vserverid":<String_value>,

"ipv46":<String_value>,

"cookiename":<String_value>,
```

```
"dynamicweight":<Double_value>,

"servicetype":<String_value>

}]}
```

get (all)

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/lbvserver_servicegroupmember_binding

Query-parameters:

bulkbindings

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/lbvserver_servicegroupmember_binding?**bulkbindings=yes**

NITRO allows you to fetch bindings in bulk.

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "lbvserver_servicegroupmember_binding": [ {

"servicegroupname":<String_value>,

"name":<String_value>,
```

```
"weight":<Double_value>,

"cookieipport":<String_value>,

"port":<Integer_value>,

"curstate":<String_value>,

"preferredlocation":<String_value>,

"vserverid":<String_value>,

"ipv46":<String_value>,

"cookiename":<String_value>,

"dynamicweight":<Double_value>,

"servicetype":<String_value>

}]}
```

count

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/lbvserver_servicegroupmember_binding/name_value<String>?**count=yes**

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{"lbvserver_servicegroupmember_binding": [ { "__count": "#no"} ] }
```

lbvserver_servicegroup_binding

Binding object showing the servicegroup that can be bound to lbvserver.

Properties

(click to see Operations)

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
servicegroupname	<String>	Read-write	The service group name bound to the selected load balancing virtual server.
name	<String>	Read-write	Name for the virtual server. Must begin with an ASCII alphanumeric or underscore (_) character, and must contain only ASCII alphanumeric, underscore, hash (#), period (.), space, colon (:), at sign (@), equal sign (=), and hyphen (-) characters. Can be changed after the virtual server is created. CLI Users: If the name includes one or more spaces, enclose the name in double or single quotation marks (for

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
			example, "my vserver" or 'my vserver'). . Minimum length = 1
servicename	<String>	Read-write	Service to bind to the virtual server. Minimum length = 1
weight	<Double>	Read-write	Integer specifying the weight of the service. A larger number specifies a greater weight. Defines the capacity of the service relative to the other services in the load balancing configuration. Determines the priority given to the service in load balancing decisions. Default value: 1 Minimum value = 1 Maximum value = 100
__count	<Double>	Read-write	count parameter

Operations

(click to see Properties)

ADD:| DELETE:| GET| GET (ALL)| COUNT

Some options that you can use for each operations:

- **Getting warnings in response:**NITRO allows you to get warnings in an operation by specifying the "warning" query parameter as "yes". For example, to get warnings while connecting to the Citrix ADC appliance, the URL is as follows:

  http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/login?warning=yes

  If any, the warnings are displayed in the response payload with the HTTP code "209 X-NITRO-WARNING".

- **Authenticated access for individual NITRO operations:**NITRO allows you to logon to the Citrix ADC appliance to perform individual operations. You can use this option instead of creating a NITRO session (using the login object) and then using that session to perform all operations,

  To do this, you must specify the username and password in the request header of the NITRO request as follows:

  X-NITRO-USER:<username>

  X-NITRO-PASS:<password>

  Note: In such cases, make sure that the request header DOES not include the following:

  Cookie:NITRO_AUTH_TOKEN=<tokenvalue>

`*Note:`*

Mandatory parameters are marked in redand placeholder content is marked in <green>.

add

URL:http://<Citrix-ADC-IP-address(NSIP)/nitro/v1/config/lbvserver_servicegroup_binding

HTTP Method:PUT

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Content-Type:application/json

Request Payload:

```
{
```

```
"lbvserver_servicegroup_binding":{

<b>"name":<String_value>,

</b>"servicename":<String_value>,

"weight":<Double_value>,

"servicegroupname":<String_value>

}}
```

Response:

HTTP Status Code on Success: 201 Created HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error

delete

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/lbvserver_servicegroup_binding/name_value<String>

Query-parameters:

args

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/lbvserver_servicegroup_binding/name_value<String>?**args=servicename:<String_value>,servicegroupname:<String_value>**

HTTP Method:DELETE

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue>

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error

get

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/lbvserver_servicegroup_binding/name_value<String>

Query-parameters:

filter

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/lbvserver_servicegroup_binding/name_value<String>?**filter=property-name1:property-value1,property-name2:property-value2**

Use this query-parameter to get the filtered set of lbvserver_servicegroup_binding resources configured on Citrix ADC. Filtering can be done on any of the properties of the resource.

pagination

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/lbvserver_servicegroup_binding/name_value<String>?**pagesize=#no;pageno=#no**

Use this query-parameter to get the lbvserver_servicegroup_binding resources in chunks.

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "lbvserver_servicegroup_binding": [ {

"servicegroupname":<String_value>,

"name":<String_value>,
```

```
"servicename":<String_value>,

"weight":<Double_value>

}]}
```

get (all)

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/lbvserver_servicegroup_binding

Query-parameters:

bulkbindings

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/lbvserver_servicegroup_binding?**bulkbindings=yes**

NITRO allows you to fetch bindings in bulk.

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "lbvserver_servicegroup_binding": [ {

"servicegroupname":<String_value>,

"name":<String_value>,
```

```
"servicename":<String_value>,

"weight":<Double_value>

}]}
```

count

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/lbvserver_servicegroup_binding/name_value<String>?**count=yes**

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{"lbvserver_servicegroup_binding": [ { "__count": "#no"} ] }
```

lbvserver_service_binding

Binding object showing the service that can be bound to lbvserver.

Properties

(click to see Operations)

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
servicegroupname	<String>	Read-write	Name of the service group. Minimum length = 1
name	<String>	Read-write	Name for the virtual server. Must begin with an ASCII alphanumeric or underscore (_) character, and must contain only ASCII alphanumeric, underscore, hash (#), period (.), space, colon (:), at sign (@), equal sign (=), and hyphen (-) characters. Can be changed after the virtual server is created. CLI Users: If the name includes one or more spaces, enclose the name in double or single quotation marks (for example, "my vserver" or 'my vserver'). . Minimum length = 1
servicename	<String>	Read-write	Service to bind to the virtual server. Minimum length = 1
weight	<Double>	Read-write	Weight to assign to the specified service. Minimum value = 1

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
			Maximum value = 100
cookieipport	<String>	Read-only	Encryped Ip address and port of the service that is inserted into the set-cookie http header.
port	<Integer>	Read-only	Port number for the virtual server. Range 1 - 65535 * in CLI is represented as 65535 in NITRO API
vsvrbindsvcport	<Integer>	Read-only	used for showing ports of bound entities. Range 1 - 65535 * in CLI is represented as 65535 in NITRO API
curstate	<String>	Read-only	Current LB vserver state. Possible values = UP, DOWN, UNKNOWN, BUSY, OUT OF SERVICE, GOING OUT OF SERVICE, DOWN WHEN GOING OUT OF SERVICE, NS_EMPTY_STR,

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
			Unknown, DISABLED
preferredlocation	<String>	Read-only	Used for displaying the location of bound services.
vserverid	<String>	Read-only	Vserver Id.
ipv46	<String>	Read-only	IPv4 or IPv6 address to assign to the virtual server.
dynamicweight	<Double>	Read-only	Dynamic weight.
vsvrbindsvcip	<String>	Read-only	used for showing the ip of bound entities.
servicetype	<String>	Read-only	Protocol used by the service (also called the service type). Possible values = HTTP, FTP, TCP, UDP, SSL, SSL_BRIDGE, SSL_TCP, DTLS, NNTP, DNS, DHCPRA, ANY, SIP_UDP, SIP_TCP, SIP_SSL, DNS_TCP, RTSP, PUSH, SSL_PUSH, RADIUS, RDP, MYSQL, MSSQL, DIAMETER, SSL_DIAMETER, TFTP, ORACLE, SMPP, SYSLOGTCP,

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
			SYSLOGUDP, FIX, SSL_FIX, PROXY, USER_TCP, USER_SSL_TCP, QUIC, IPFIX, LOGSTREAM
__count	<Double>	Read-only	count parameter

Operations

(click to see Properties)

ADD:| DELETE:| GET| GET (ALL)| COUNT

Some options that you can use for each operations:

- **Getting warnings in response:**NITRO allows you to get warnings in an operation by specifying the "warning" query parameter as "yes". For example, to get warnings while connecting to the Citrix ADC appliance, the URL is as follows:

  http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/login?warning=yes

  If any, the warnings are displayed in the response payload with the HTTP code "209 X-NITRO-WARNING".

- **Authenticated access for individual NITRO operations:**NITRO allows you to logon to the Citrix ADC appliance to perform individual operations. You can use this option instead of creating a NITRO session (using the login object) and then using that session to perform all operations,

  To do this, you must specify the username and password in the request header of the NITRO request as follows:

  X-NITRO-USER:<username>

  X-NITRO-PASS:<password>

  Note: In such cases, make sure that the request header DOES not include the following:

  Cookie:NITRO_AUTH_TOKEN=<tokenvalue>

`*Note:`*

Mandatory parameters are marked in redand placeholder content is marked in <green>.

add

URL:http://<Citrix-ADC-IP-address(NSIP)/nitro/v1/config/lbvserver_service_binding

HTTP Method:PUT

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Content-Type:application/json

Request Payload:

```
{

"lbvserver_service_binding":{

<b>"name":<String_value>,

</b>"servicename":<String_value>,

"weight":<Double_value>,

"servicegroupname":<String_value>

}}
```

Response:

HTTP Status Code on Success: 201 Created HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error

delete

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/lbvserver_service_binding/name_value<String>

Query-parameters:

args

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/lbvserver_service_binding/name_value<String>?**args=servicename:<String_value>,servicegroupname:<String_value>**

HTTP Method:DELETE

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue>

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error

get

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/lbvserver_service_binding/name_value<String>

Query-parameters:

filter

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/lbvserver_service_binding/name_value<String>?**filter=property-name1:property-value1,property-name2:property-value2**

Use this query-parameter to get the filtered set of lbvserver_service_binding resources configured on Citrix ADC. Filtering can be done on any of the properties of the resource.

pagination

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/lbvserver_service_binding/name_value<String>?**pagesize=#no;pageno=#no**

Use this query-parameter to get the lbvserver_service_binding resources in chunks.

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "lbvserver_service_binding": [ {

"servicegroupname":<String_value>,

"name":<String_value>,

"servicename":<String_value>,

"weight":<Double_value>,

"cookieipport":<String_value>,

"port":<Integer_value>,

"vsvrbindsvcport":<Integer_value>,

"curstate":<String_value>,

"preferredlocation":<String_value>,

"vserverid":<String_value>,

"ipv46":<String_value>,

"dynamicweight":<Double_value>,

"vsvrbindsvcip":<String_value>,

"servicetype":<String_value>

}]}
```

get (all)

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/lbvserver_service_binding

Query-parameters:

bulkbindings

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/lbvserver_service_binding?**bulkbindings=yes**

NITRO allows you to fetch bindings in bulk.

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "lbvserver_service_binding": [ {

"servicegroupname":<String_value>,

"name":<String_value>,

"servicename":<String_value>,

"weight":<Double_value>,

"cookieipport":<String_value>,

"port":<Integer_value>,
```

```
"vsvrbindsvcport":<Integer_value>,

"curstate":<String_value>,

"preferredlocation":<String_value>,

"vserverid":<String_value>,

"ipv46":<String_value>,

"dynamicweight":<Double_value>,

"vsvrbindsvcip":<String_value>,

"servicetype":<String_value>

}]}
```

count

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/lbvserver_service_binding/name_value<String>?**count=yes**

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{"lbvserver_service_binding": [ { "__count": "#no"} ] }
```

lbvserver_spilloverpolicy_binding

Binding object showing the spilloverpolicy that can be bound to lbvserver.

Properties

(click to see Operations)

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
priority	<Double>	Read-write	Priority.
bindpoint	<String>	Read-write	The bindpoint to which the policy is bound. Possible values = REQUEST, RESPONSE
policyname	<String>	Read-write	Name of the policy bound to the LB vserver.
labelname	<String>	Read-write	Name of the virtual server or user-defined policy label to invoke if the policy evaluates to TRUE.
name	<String>	Read-write	Name for the virtual server. Must begin with an ASCII alphanumeric or underscore (_) character, and must contain only ASCII alphanumeric, underscore, hash (#), period (.), space, colon (:), at sign (@), equal sign (=), and

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
			hyphen (-) characters. Can be changed after the virtual server is created. CLI Users: If the name includes one or more spaces, enclose the name in double or single quotation marks (for example, "my vserver" or 'my vserver'). . Minimum length = 1
gotopriorityexpression	<String>	Read-write	Expression specifying the priority of the next policy which will get evaluated if the current policy rule evaluates to TRUE.
invoke	<Boolean>	Read-write	Invoke policies bound to a virtual server or policy label.
labeltype	<String>	Read-write	Type of policy label to invoke. Applicable only to rewrite, videooptimization and cache policies. Available settings function as follows: * reqvserver - Evaluate the request against the request-based policies bound to the specified virtual server. * resvserver - Evaluate

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
			the response against the response-based policies bound to the specified virtual server. * policylabel - invoke the request or response against the specified user-defined policy label. Possible values = reqvserver, resvserver, policylabel
__count	<Double>	Read-write	count parameter

Operations

(click to see Properties)

ADD:| DELETE:| GET| GET (ALL)| COUNT

Some options that you can use for each operations:

- **Getting warnings in response:**NITRO allows you to get warnings in an operation by specifying the "warning" query parameter as "yes". For example, to get warnings while connecting to the Citrix ADC appliance, the URL is as follows:

  http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/login?warning=yes

  If any, the warnings are displayed in the response payload with the HTTP code "209 X-NITRO-WARNING".

- **Authenticated access for individual NITRO operations:**NITRO allows you to logon to the Citrix ADC appliance to perform individual operations. You can use this option instead of creating a NITRO session (using the login object) and then using that session to perform all operations,

  To do this, you must specify the username and password in the request header of the NITRO request as follows:

  X-NITRO-USER:<username>

X-NITRO-PASS:<password>

Note: In such cases, make sure that the request header DOES not include the following:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue>

`*Note:`*

Mandatory parameters are marked in redand placeholder content is marked in <green>.

add

URL:http://<Citrix-ADC-IP-address(NSIP)/nitro/v1/config/lbvserver_spilloverpolicy_binding

HTTP Method:PUT

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Content-Type:application/json

Request Payload:

```
{
"lbvserver_spilloverpolicy_binding":{
<b>"name":<String_value>,
</b>"policyname":<String_value>,
"priority":<Double_value>,
"gotopriorityexpression":<String_value>,
"bindpoint":<String_value>,
"invoke":<Boolean_value>,
"labeltype":<String_value>,
"labelname":<String_value>
}}
```

Response:

HTTP Status Code on Success: 201 Created HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error

delete

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/lbvserver_spilloverpolicy_binding/name_value<String>

Query-parameters:

args

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/lbvserver_spilloverpolicy_binding/name_value<String>?**args=policyname:<String_value>,bindpoint:<String_value>,priority:<Double_value>**

HTTP Method:DELETE

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue>

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error

get

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/lbvserver_spilloverpolicy_binding/name_value<String>

Query-parameters:

filter

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/lbvserver_spilloverpolicy_binding/name_value<String>?**filter=property-name1:property-value1,property-name2:property-value2**

Use this query-parameter to get the filtered set of lbvserver_spilloverpolicy_binding resources configured on Citrix ADC. Filtering can be done on any of the properties of the resource.

pagination

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/lbvserver_spilloverpolicy_binding/name_value<String>?**pagesize=#no;pageno=#no**

Use this query-parameter to get the lbvserver_spilloverpolicy_binding resources in chunks.

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "lbvserver_spilloverpolicy_binding": [ {

"priority":<Double_value>,

"bindpoint":<String_value>,

"policyname":<String_value>,

"labelname":<String_value>,

"name":<String_value>,

"gotopriorityexpression":<String_value>,

"invoke":<Boolean_value>,
```

```
"labeltype":<String_value>

}]}
```

get (all)

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/lbvserver_spilloverpolicy_binding

Query-parameters:

bulkbindings

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/lbvserver_spilloverpolicy_binding?**bulkbindings=yes**

NITRO allows you to fetch bindings in bulk.

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "lbvserver_spilloverpolicy_binding": [ {

"priority":<Double_value>,

"bindpoint":<String_value>,

"policyname":<String_value>,
```

```
"labelname":<String_value>,

"name":<String_value>,

"gotopriorityexpression":<String_value>,

"invoke":<Boolean_value>,

"labeltype":<String_value>

}]}
```

count

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/lbvserver_spilloverpolicy_binding/name_value<String>?**count=yes**

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{"lbvserver_spilloverpolicy_binding": [ { "__count": "#no"} ] }
```

lbvserver_tmtrafficpolicy_binding

Binding object showing the tmtrafficpolicy that can be bound to lbvserver.

Properties

(click to see Operations)

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
priority	<Double>	Read-write	Priority.
bindpoint	<String>	Read-write	Bind point to which to bind the policy. Applicable only to compression, rewrite, videooptimization and cache policies. Possible values = REQUEST, RESPONSE
policyname	<String>	Read-write	Name of the policy bound to the LB vserver.
labelname	<String>	Read-write	Name of the virtual server or user-defined policy label to invoke if the policy evaluates to TRUE.
name	<String>	Read-write	Name for the virtual server. Must begin with an ASCII alphanumeric or underscore (_) character, and must contain only ASCII alphanumeric, underscore, hash (#), period (.), space, colon (:), at sign (@), equal sign (=), and hyphen (-) characters. Can be

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
			changed after the virtual server is created. CLI Users: If the name includes one or more spaces, enclose the name in double or single quotation marks (for example, "my vserver" or 'my vserver'). . Minimum length = 1
gotopriorityexpression	<String>	Read-write	Expression or other value specifying the next policy to be evaluated if the current policy evaluates to TRUE. Specify one of the following values: * NEXT - Evaluate the policy with the next higher priority number. * END - End policy evaluation. * USE_INVOCATION_RESULT - Applicable if this policy invokes another policy label. If the final goto in the invoked policy label has a value of END, the evaluation stops. If the final goto is anything other than END, the current policy label performs a NEXT. * An expression that evaluates to a

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
			number. If you specify an expression, the number to which it evaluates determines the next policy to evaluate, as follows: * If the expression evaluates to a higher numbered priority, the policy with that priority is evaluated next. * If the expression evaluates to the priority of the current policy, the policy with the next higher numbered priority is evaluated next. * If the expression evaluates to a priority number that is numerically higher than the highest numbered priority, policy evaluation ends. An UNDEF event is triggered if: * The expression is invalid. * The expression evaluates to a priority number that is numerically lower than the current policy's priority. * The expression evaluates to a priority number that is between the current policy's

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
			priority number (say, 30) and the highest priority number (say, 100), but does not match any configured priority number (for example, the expression evaluates to the number 85). This example assumes that the priority number increments by 10 for every successive policy, and therefore a priority number of 85 does not exist in the policy label.
invoke	<Boolean>	Read-write	Invoke policies bound to a virtual server or policy label.
labeltype	<String>	Read-write	Type of policy label to invoke. Applicable only to rewrite, videooptimization and cache policies. Available settings function as follows: * reqvserver - Evaluate the request against the request-based policies bound to the specified virtual server. * resvserver - Evaluate the response against the response-based policies bound to the

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
			specified virtual server. * policylabel - invoke the request or response against the specified user-defined policy label. Possible values = reqvserver, resvserver, policylabel
__count	<Double>	Read-write	count parameter

Operations

(click to see Properties)

ADD:| DELETE:| GET| GET (ALL)| COUNT

Some options that you can use for each operations:

- **Getting warnings in response:**NITRO allows you to get warnings in an operation by specifying the "warning" query parameter as "yes". For example, to get warnings while connecting to the Citrix ADC appliance, the URL is as follows:

  http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/login?warning=yes

  If any, the warnings are displayed in the response payload with the HTTP code "209 X-NITRO-WARNING".

- **Authenticated access for individual NITRO operations:**NITRO allows you to logon to the Citrix ADC appliance to perform individual operations. You can use this option instead of creating a NITRO session (using the login object) and then using that session to perform all operations,

  To do this, you must specify the username and password in the request header of the NITRO request as follows:

  X-NITRO-USER:<username>

  X-NITRO-PASS:<password>

Note: In such cases, make sure that the request header DOES not include the following:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue>

`*Note:` *

Mandatory parameters are marked in redand placeholder content is marked in <green>.

add

URL:http://<Citrix-ADC-IP-address(NSIP)/nitro/v1/config/lbvserver_tmtrafficpolicy_binding

HTTP Method:PUT

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Content-Type:application/json

Request Payload:

```
{
"lbvserver_tmtrafficpolicy_binding":{
<b>"name":<String_value>,
</b>"policyname":<String_value>,
"priority":<Double_value>,
"gotopriorityexpression":<String_value>,
"bindpoint":<String_value>,
"invoke":<Boolean_value>,
"labeltype":<String_value>,
"labelname":<String_value>
}}
```

Response:

HTTP Status Code on Success: 201 Created HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error

delete

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/lbvserver_tmtrafficpolicy_binding/name_value<String>

Query-parameters:

args

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/lbvserver_tmtrafficpolicy_binding/name_value<String>?**args=policyname:<String_value>,bindpoint:<String_value>,priority:<Double_value>**

HTTP Method:DELETE

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue>

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error

get

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/lbvserver_tmtrafficpolicy_binding/name_value<String>

Query-parameters:

filter

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/lbvserver_tmtrafficpolicy_binding/name_value<String>?**filter=property-name1:property-value1,property-name2:property-value2**

Use this query-parameter to get the filtered set of lbvserver_tmtrafficpolicy_binding resources configured on Citrix ADC. Filtering can be done on any of the properties of the resource.

pagination

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/lbvserver_tmtrafficpolicy_binding/name_value<String>?**pagesize=#no;pageno=#no**

Use this query-parameter to get the lbvserver_tmtrafficpolicy_binding resources in chunks.

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "lbvserver_tmtrafficpolicy_binding": [ {

"priority":<Double_value>,

"bindpoint":<String_value>,

"policyname":<String_value>,

"labelname":<String_value>,

"name":<String_value>,

"gotopriorityexpression":<String_value>,

"invoke":<Boolean_value>,

"labeltype":<String_value>

}]}
```

get (all)

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/lbvserver_tmtrafficpolicy_binding

Query-parameters:

bulkbindings

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/lbvserver_tmtrafficpolicy_binding?**bulkbindings=yes**

NITRO allows you to fetch bindings in bulk.

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "lbvserver_tmtrafficpolicy_binding": [ {

"priority":<Double_value>,

"bindpoint":<String_value>,

"policyname":<String_value>,

"labelname":<String_value>,

"name":<String_value>,
```

```
"gotopriorityexpression":<String_value>,

"invoke":<Boolean_value>,

"labeltype":<String_value>

}]}
```

count

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/lbvserver_tmtrafficpolicy_binding/name_value<String>?**count=yes**

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{"lbvserver_tmtrafficpolicy_binding": [ { "__count": "#no"} ] }
```

lbvserver_transformpolicy_binding

Binding object showing the transformpolicy that can be bound to lbvserver.

Properties

(click to see Operations)

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
priority	<Double>	Read-write	Priority.
bindpoint	<String>	Read-write	The bindpoint to which the policy is bound. Possible values = REQUEST, RESPONSE
policyname	<String>	Read-write	Name of the policy bound to the LB vserver.
labelname	<String>	Read-write	Name of the label invoked.
name	<String>	Read-write	Name for the virtual server. Must begin with an ASCII alphanumeric or underscore (_) character, and must contain only ASCII alphanumeric, underscore, hash (#), period (.), space, colon (:), at sign (@), equal sign (=), and hyphen (-) characters. Can be changed after the virtual server is created. CLI Users: If the name includes one or more spaces, enclose the name in double or single quotation marks (for example, "my

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
			vserver" or 'my vserver'). . Minimum length = 1
gotopriorityexpression	<String>	Read-write	Expression specifying the priority of the next policy which will get evaluated if the current policy rule evaluates to TRUE.
invoke	<Boolean>	Read-write	Invoke policies bound to a virtual server or policy label.
labeltype	<String>	Read-write	The invocation type. Possible values = reqvserver, resvserver, policylabel
__count	<Double>	Read-write	count parameter

Operations

(click to see Properties)

ADD:| DELETE:| GET| GET (ALL)| COUNT

Some options that you can use for each operations:

- **Getting warnings in response:**NITRO allows you to get warnings in an operation by specifying the "warning" query parameter as "yes". For example, to get warnings while connecting to the Citrix ADC appliance, the URL is as follows:

  http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/login?warning=yes

  If any, the warnings are displayed in the response payload with the HTTP code "209 X-NITRO-WARNING".

- **Authenticated access for individual NITRO operations:**NITRO allows you to logon to the Citrix ADC appliance to perform individual operations. You can use this option instead of creating a NITRO session (using the login object) and then using that session to perform all operations,

  To do this, you must specify the username and password in the request header of the NITRO request as follows:

  X-NITRO-USER:<username>

  X-NITRO-PASS:<password>

  Note: In such cases, make sure that the request header DOES not include the following:

  Cookie:NITRO_AUTH_TOKEN=<tokenvalue>

`*Note:`*

Mandatory parameters are marked in redand placeholder content is marked in <green>.

add

URL:http://<Citrix-ADC-IP-address(NSIP)/nitro/v1/config/lbvserver_transformpolicy_binding

HTTP Method:PUT

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Content-Type:application/json

Request Payload:

```
{

"lbvserver_transformpolicy_binding":{

<b>"name":<String_value>,

</b>"policyname":<String_value>,

"priority":<Double_value>,

"gotopriorityexpression":<String_value>,
```

```
"bindpoint":<String_value>,

"invoke":<Boolean_value>,

"labeltype":<String_value>,

"labelname":<String_value>

}}
```

Response:

HTTP Status Code on Success: 201 Created HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error

delete

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/lbvserver_transformpolicy_binding/name_value<String>

Query-parameters:

args

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/lbvserver_transformpolicy_binding/name_value<String>?
args=policyname:<String_value>,bindpoint:<String_value>,priority:<Double_value>

HTTP Method:DELETE

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue>

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error

get

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/lbvserver_transformpolicy_binding/name_value<String>

Query-parameters:

filter

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/lbvserver_transformpolicy_binding/name_value<String>?**filter=property-name1:property-value1,property-name2:property-value2**

Use this query-parameter to get the filtered set of lbvserver_transformpolicy_binding resources configured on Citrix ADC. Filtering can be done on any of the properties of the resource.

pagination

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/lbvserver_transformpolicy_binding/name_value<String>?**pagesize=#no;pageno=#no**

Use this query-parameter to get the lbvserver_transformpolicy_binding resources in chunks.

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "lbvserver_transformpolicy_binding": [ {

"priority":<Double_value>,

"bindpoint":<String_value>,
```

```
"policyname":<String_value>,

"labelname":<String_value>,

"name":<String_value>,

"gotopriorityexpression":<String_value>,

"invoke":<Boolean_value>,

"labeltype":<String_value>

}]}
```

get (all)

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/lbvserver_transformpolicy_binding

Query-parameters:

bulkbindings

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/lbvserver_transformpolicy_binding?**bulkbindings=yes**

NITRO allows you to fetch bindings in bulk.

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "lbvserver_transformpolicy_binding": [ {

"priority":<Double_value>,

"bindpoint":<String_value>,

"policyname":<String_value>,

"labelname":<String_value>,

"name":<String_value>,

"gotopriorityexpression":<String_value>,

"invoke":<Boolean_value>,

"labeltype":<String_value>

}]}
```

count

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/lbvserver_transformpolicy_binding/name_value<String>?**count=yes**

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{"lbvserver_transformpolicy_binding": [ { "__count": "#no"} ] }
```

lbwlm

Configuration for web log manager resource.

Properties

(click to see Operations)

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
wlmname	<String>	Read-write	The name of the Work Load Manager. Minimum length = 1
ipaddress	<String>	Read-write	The IP address of the WLM.
port	<Integer>	Read-write	The port of the WLM. Range 1 - 65535 * in CLI is represented as 65535 in NITRO API
lbuid	<String>	Read-write	The LBUID for the Load Balancer to communicate to the Work Load Manager.
katimeout	<Double>	Read-write	The idle time period after which Citrix ADC would probe the WLM. The value ranges from 1 to 1440 minutes.

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
			Default value: 2 Minimum value = 0 Maximum value = 1440
secure	<String>	Read-only	Use this parameter to enable secure mode of communication with WLM. Possible values = YES, NO
state	<String>	Read-only	State of the WLM. Possible values = ACTIVE, INACTIVE, UNKNOWN
__count	<Double>	Read-only	count parameter

Operations

(click to see Properties)

ADD| DELETE| UPDATE| UNSET| GET (ALL)| GET| COUNT

Some options that you can use for each operations:

- **Getting warnings in response:**NITRO allows you to get warnings in an operation by specifying the "warning" query parameter as "yes". For example, to get warnings while connecting to the Citrix ADC appliance, the URL is as follows:

  http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/login?warning=yes

  If any, the warnings are displayed in the response payload with the HTTP code "209 X-NITRO-WARNING".

- **Authenticated access for individual NITRO operations:**NITRO allows you to logon to the Citrix ADC appliance to perform individual operations. You can use this option instead of creating a NITRO session (using the login object) and then using that session to perform all operations,

  To do this, you must specify the username and password in the request header of the NITRO request as follows:

  X-NITRO-USER:<username>

  X-NITRO-PASS:<password>

  Note: In such cases, make sure that the request header DOES not include the following:

  Cookie:NITRO_AUTH_TOKEN=<tokenvalue>

`*Note:`*

Mandatory parameters are marked in redand placeholder content is marked in <green>.

add

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/lbwlm

HTTP Method:POST

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Content-Type:application/json

Request Payload:

```
{"lbwlm":{

<b>"wlmname":<String_value>,

</b>"ipaddress":<String_value>,

"port":<Integer_value>,

<b>"lbuid":<String_value>,

</b>"katimeout":<Double_value>
```

```
}}
```

Response:

HTTP Status Code on Success: 201 Created HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error

delete

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/lbwlm/wlmname_value<String>

HTTP Method:DELETE

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue>

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error

update

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/lbwlm

HTTP Method:PUT

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Content-Type:application/json

Request Payload:

```
{"lbwlm":{

<b>"wlmname":<String_value>,

</b>"katimeout":<Double_value>
```

```
}}
```

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error

unset

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/lbwlm?**action=unset**

HTTP Method:POST

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Content-Type:application/json

Request Payload:

```
{"lbwlm":{

<b>"wlmname":<String_value>,

</b>"katimeout":true

}}
```

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error

get (all)

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/lbwlm

Query-parameters:

attrs

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/lbwlm?**attrs=property-name1,property-name2**

Use this query parameter to specify the resource details that you want to retrieve.

filter

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/lbwlm?**filter=property-name1:property-val1,property-name2:property-val2**

Use this query-parameter to get the filtered set of lbwlm resources configured on Citrix ADC. Filtering can be done on any of the properties of the resource.

view

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/lbwlm?**view=summary**

Note: By default, the retrieved results are displayed in detail view (?view=detail).

pagination

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/lbwlm?**pagesize=#no;pageno=#no**

Use this query-parameter to get the lbwlm resources in chunks.

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "lbwlm": [ {

"wlmname":<String_value>,

"ipaddress":<String_value>,

"port":<Integer_value>,

"secure":<String_value>,

"katimeout":<Double_value>,

"lbuid":<String_value>,

"state":<String_value>

}]}
```

get

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/lbwlm/wlmname_value<String>

Query-parameters:

attrs

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/lbwlm/wlmname_value<String>?**attrs=property-name1,property-name2**

Use this query parameter to specify the resource details that you want to retrieve.

view

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/lbwlm/wlmname_value<String>?**view=summary**

Note: By default, the retrieved results are displayed in detail view (?view=detail).

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "lbwlm": [ {

"wlmname":<String_value>,

"ipaddress":<String_value>,

"port":<Integer_value>,

"secure":<String_value>,

"katimeout":<Double_value>,

"lbuid":<String_value>,

"state":<String_value>

}]}
```

count

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/lbwlm?**count=yes**

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "lbwlm": [ { "__count": "#no"} ] }
```

lbwlm_binding

Binding object which returns the resources bound to lbwlm.

Properties

(click to see Operations)

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
wlmname	<String>	Read-write	The name of the work load manager. Minimum length = 1
lbwlm_lbvserver_binding	<lbwlm_lbvserver_binding[]>	Read-only	lbvserver that can be bound to lbwlm.

Operations

(click to see Properties)

GET| GET (ALL)

Some options that you can use for each operations:

- **Getting warnings in response:**NITRO allows you to get warnings in an operation by specifying the "warning" query parameter as "yes". For example, to get warnings while connecting to the Citrix ADC appliance, the URL is as follows:

  http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/login?warning=yes

If any, the warnings are displayed in the response payload with the HTTP code "209 X-NITRO-WARNING".

- **Authenticated access for individual NITRO operations:**NITRO allows you to logon to the Citrix ADC appliance to perform individual operations. You can use this option instead of creating a NITRO session (using the login object) and then using that session to perform all operations,

  To do this, you must specify the username and password in the request header of the NITRO request as follows:

  X-NITRO-USER:<username>

  X-NITRO-PASS:<password>

  Note: In such cases, make sure that the request header DOES not include the following:

  Cookie:NITRO_AUTH_TOKEN=<tokenvalue>

`*Note:`*

Mandatory parameters are marked in redand placeholder content is marked in <green>.

get

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/lbwlm_binding/wlmname_value<String>

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "lbwlm_binding": [ {

"wlmname":<String_value>,

"lbwlm_lbvserver_binding":<lbwlm_lbvserver_binding[]_value>

}]}
```

get (all)

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/lbwlm_binding

Query-parameters:

bulkbindings

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/lbwlm_binding?**bulkbindings=yes**

NITRO allows you to fetch bindings in bulk.

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "lbwlm_binding": [ {

"wlmname":<String_value>,
```

```
"lbwlm_lbvserver_binding":<lbwlm_lbvserver_binding[]_value>

}]}
```

lbwlm_lbvserver_binding

Binding object showing the lbvserver that can be bound to lbwlm.

Properties

(click to see Operations)

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
wlmname	<String>	Read-write	The name of the Work Load Manager. Minimum length = 1
vservername	<String>	Read-write	Name of the virtual server which is to be bound to the WLM.
__count	<Double>	Read-write	count parameter

Operations

(click to see Properties)

ADD:| DELETE:| GET| GET (ALL)| COUNT

Some options that you can use for each operations:

- **Getting warnings in response:**NITRO allows you to get warnings in an operation by specifying the "warning" query parameter as "yes". For example, to get warnings while connecting to the Citrix ADC appliance, the URL is as follows:

  http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/login?warning=yes

  If any, the warnings are displayed in the response payload with the HTTP code "209 X-NITRO-WARNING".

- **Authenticated access for individual NITRO operations:**NITRO allows you to logon to the Citrix ADC appliance to perform individual operations. You can use this option instead of creating a NITRO session (using the login object) and then using that session to perform all operations,

  To do this, you must specify the username and password in the request header of the NITRO request as follows:

  X-NITRO-USER:<username>

  X-NITRO-PASS:<password>

  Note: In such cases, make sure that the request header DOES not include the following:

  Cookie:NITRO_AUTH_TOKEN=<tokenvalue>

`*Note:`*

Mandatory parameters are marked in redand placeholder content is marked in <green>.

add

URL:http://<Citrix-ADC-IP-address(NSIP)/nitro/v1/config/lbwlm_lbvserver_binding

HTTP Method:PUT

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Content-Type:application/json

Request Payload:

```
{

"lbwlm_lbvserver_binding":{

<b>"wlmname":<String_value>,

</b><b>"vservername":<String_value></b>

}}
```

Response:

HTTP Status Code on Success: 201 Created HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error

delete

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/lbwlm_lbvserver_binding/wlmname_value<String>

Query-parameters:

args

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/lbwlm_lbvserver_binding/wlmname_value<String>?**args=vservername:<String_value>**

HTTP Method:DELETE

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue>

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error

get

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/lbwlm_lbvserver_binding/wlmname_value<String>

Query-parameters:

filter

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/lbwlm_lbvserver_binding/wlmname_value<String>?**filter=property-name1:property-value1,property-name2:property-value2**

Use this query-parameter to get the filtered set of lbwlm_lbvserver_binding resources configured on Citrix ADC. Filtering can be done on any of the properties of the resource.

pagination

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/lbwlm_lbvserver_binding/wlmname_value<String>?**pagesize=#no;pageno=#no**

Use this query-parameter to get the lbwlm_lbvserver_binding resources in chunks.

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "lbwlm_lbvserver_binding": [ {

"wlmname":<String_value>,

"vservername":<String_value>

}]}
```

get (all)

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/lbwlm_lbvserver_binding

Query-parameters:

bulkbindings

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/lbwlm_lbvserver_binding?**bulkbindings=yes**

NITRO allows you to fetch bindings in bulk.

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "lbwlm_lbvserver_binding": [ {

"wlmname":<String_value>,

"vservername":<String_value>

}]}
```

count

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/lbwlm_lbvserver_binding/wlmname_value<String>?**count=yes**

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{"lbwlm_lbvserver_binding": [ { "__count": "#no"} ] }
```

lbmonbindings_gslbservicegroup_binding

Binding object showing the gslbservicegroup that can be bound to lbmonbindings.

Properties

(click to see Operations)

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
monitorname	<String>	Read-write	The name of the monitor. Minimum length = 1
servicegroupname	<String>	Read-write	The name of the service group.
boundservicegroups vrstate	<String>	Read-only	The state of the servicegroup. Possible values = ENABLED, DISABLED
monstate	<String>	Read-only	The configured state (enable/disable) of Monitor on this service. Possible values = ENABLED, DISABLED
servicetype	<String>	Read-only	The type of service.

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
			Possible values = HTTP, FTP, TCP, UDP, SSL, SSL_BRIDGE, SSL_TCP, DTLS, NNTP, RPCSVR, DNS, ADNS, SNMP, RTSP, DHCPRA, ANY, SIP_UDP, SIP_TCP, SIP_SSL, DNS_TCP, ADNS_TCP, MYSQL, MSSQL, ORACLE, RADIUS, RADIUSListener, RDP, DIAMETER, SSL_DIAMETER, TFTP, SMPP, PPTP, GRE, SYSLOGTCP, SYSLOGUDP, FIX, SSL_FIX, USER_TCP, USER_SSL_TCP, QUIC, IPFIX, LOGSTREAM
__count	<Double>	Read-only	count parameter

Operations

(click to see Properties)

GET| GET (ALL)| COUNT

Some options that you can use for each operations:

- **Getting warnings in response:**NITRO allows you to get warnings in an operation by specifying the "warning" query parameter as "yes". For example, to get warnings while connecting to the Citrix ADC appliance, the URL is as follows:

  http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/login?warning=yes

  If any, the warnings are displayed in the response payload with the HTTP code "209 X-NITRO-WARNING".

- **Authenticated access for individual NITRO operations:**NITRO allows you to logon to the Citrix ADC appliance to perform individual operations. You can use this option instead of creating a NITRO session (using the login object) and then using that session to perform all operations,

  To do this, you must specify the username and password in the request header of the NITRO request as follows:

  X-NITRO-USER:<username>

  X-NITRO-PASS:<password>

  Note: In such cases, make sure that the request header DOES not include the following:

  Cookie:NITRO_AUTH_TOKEN=<tokenvalue>

`*Note:`*

Mandatory parameters are marked in redand placeholder content is marked in <green>.

get

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/lbmonbindings_gslbservicegroup_binding/monitorname_value<String>

Query-parameters:

filter

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/lbmonbindings_gslbservicegroup_binding/monitorname_value<String>?**filter=property-name1:property-value1,property-name2:property-value2**

Use this query-parameter to get the filtered set of lbmonbindings_gslbservicegroup_binding resources configured on Citrix ADC. Filtering can be done on any of the properties of the resource.

pagination

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/lbmonbindings_gslbservicegroup_binding/monitorname_value<String>?**pagesize=#no;pageno=#no**

Use this query-parameter to get the lbmonbindings_gslbservicegroup_binding resources in chunks.

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "lbmonbindings_gslbservicegroup_binding": [ {

"monitorname":<String_value>,

"servicegroupname":<String_value>,

"boundservicegroupsvrstate":<String_value>,

"monstate":<String_value>,

"servicetype":<String_value>

}]}
```

get (all)

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/lbmonbindings_gslbservicegroup_binding

Query-parameters:

bulkbindings

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/lbmonbindings_gslbservicegroup_binding?**bulkbindings=yes**

NITRO allows you to fetch bindings in bulk.

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "lbmonbindings_gslbservicegroup_binding": [ {

"monitorname":<String_value>,

"servicegroupname":<String_value>,

"boundservicegroupsvrstate":<String_value>,

"monstate":<String_value>,

"servicetype":<String_value>

}]}
```

count

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/lbmonbindings_gslbservicegroup_binding/monitorname_value<String>?**count=yes**

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{"lbmonbindings_gslbservicegroup_binding": [ { "__count": "#no"} ] }
```

lbvserver_contentinspectionpolicy_binding

Binding object showing the contentinspectionpolicy that can be bound to lbvserver.

Properties

(click to see Operations)

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
priority	<Double>	Read-write	Priority.
bindpoint	<String>	Read-write	The bindpoint to which the policy is bound. Possible values = REQUEST, RESPONSE

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
policyname	<String>	Read-write	Name of the policy bound to the LB vserver.
labelname	<String>	Read-write	Name of the label invoked.
name	<String>	Read-write	Name for the virtual server. Must begin with an ASCII alphanumeric or underscore (_) character, and must contain only ASCII alphanumeric, underscore, hash (#), period (.), space, colon (:), at sign (@), equal sign (=), and hyphen (-) characters. Can be changed after the virtual server is created. CLI Users: If the name includes one or more spaces, enclose the name in double or single quotation marks (for example, "my vserver" or 'my vserver'). . Minimum length = 1
gotopriorityexpression	<String>	Read-write	Expression specifying the priority of the next policy which will get evaluated if the

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
			current policy rule evaluates to TRUE.
invoke	<Boolean>	Read-write	Invoke policies bound to a virtual server or policy label.
labeltype	<String>	Read-write	The invocation type. Possible values = reqvserver, resvserver, policylabel
__count	<Double>	Read-write	count parameter

Operations

(click to see Properties)

ADD:| DELETE:| GET| GET (ALL)| COUNT

Some options that you can use for each operations:

- **Getting warnings in response:**NITRO allows you to get warnings in an operation by specifying the "warning" query parameter as "yes". For example, to get warnings while connecting to the Citrix ADC appliance, the URL is as follows:

  http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/login?warning=yes

  If any, the warnings are displayed in the response payload with the HTTP code "209 X-NITRO-WARNING".

- **Authenticated access for individual NITRO operations:**NITRO allows you to logon to the Citrix ADC appliance to perform individual operations. You can use this option instead of creating a NITRO session (using the login object) and then using that session to perform all operations,

  To do this, you must specify the username and password in the request header of the NITRO request as follows:

  X-NITRO-USER:<username>

X-NITRO-PASS:<password>

Note: In such cases, make sure that the request header DOES not include the following:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue>

`*Note:`*

Mandatory parameters are marked in redand placeholder content is marked in <green>.

add

URL:http://<Citrix-ADC-IP-address(NSIP)/nitro/v1/config/lbvserver_contentinspectionpolicy_binding

HTTP Method:PUT

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Content-Type:application/json

Request Payload:

```
{
"lbvserver_contentinspectionpolicy_binding":{
<b>"name":<String_value>,
</b>"policyname":<String_value>,
"priority":<Double_value>,
"gotopriorityexpression":<String_value>,
"bindpoint":<String_value>,
"invoke":<Boolean_value>,
"labeltype":<String_value>,
"labelname":<String_value>
```

```
}}
```

Response:

HTTP Status Code on Success: 201 Created HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error

delete

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/lbvserver_contentinspectionpolicy_binding/name_value<String>

Query-parameters:

args

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/lbvserver_contentinspectionpolicy_binding/name_value<String>?**args=policyname:<String_value>,bindpoint:<String_value>,priority:<Double_value>**

HTTP Method:DELETE

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue>

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error

get

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/lbvserver_contentinspectionpolicy_binding/name_value<String>

Query-parameters:

filter

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/lbvserver_contentinspectionpolicy_binding/name_value<String>?**filter=property-name1:property-value1,property-name2:property-value2**

Use this query-parameter to get the filtered set of lbvserver_contentinspectionpolicy_binding resources configured on Citrix ADC. Filtering can be done on any of the properties of the resource.

pagination

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/lbvserver_contentinspectionpolicy_binding/name_value<String>?**pagesize=#no;pageno=#no**

Use this query-parameter to get the lbvserver_contentinspectionpolicy_binding resources in chunks.

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "lbvserver_contentinspectionpolicy_binding": [ {

"priority":<Double_value>,

"bindpoint":<String_value>,

"policyname":<String_value>,

"labelname":<String_value>,

"name":<String_value>,
```

```
"gotopriorityexpression":<String_value>,

"invoke":<Boolean_value>,

"labeltype":<String_value>

}]}
```

get (all)

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/lbvserver_contentinspectionpolicy_binding

Query-parameters:

bulkbindings

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/lbvserver_contentinspectionpolicy_binding?**bulkbindings=yes**

NITRO allows you to fetch bindings in bulk.

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "lbvserver_contentinspectionpolicy_binding": [ {

"priority":<Double_value>,
```

```
"bindpoint":<String_value>,

"policyname":<String_value>,

"labelname":<String_value>,

"name":<String_value>,

"gotopriorityexpression":<String_value>,

"invoke":<Boolean_value>,

"labeltype":<String_value>

}]}
```

count

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/lbvserver_contentinspectionpolicy_binding/name_value<String>?**count=yes**

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{"lbvserver_contentinspectionpolicy_binding": [ { "__count": "#no"} ] }
```

lbvserver_videooptimizationdetectionpolicy_binding

Binding object showing the videooptimizationdetectionpolicy that can be bound to lbvserver.

Properties

(click to see Operations)

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
priority	<Double>	Read-write	Priority.
bindpoint	<String>	Read-write	The bindpoint to which the policy is bound. Possible values = REQUEST, RESPONSE
policyname	<String>	Read-write	Name of the policy bound to the LB vserver.
labelname	<String>	Read-write	Name of the label invoked.
name	<String>	Read-write	Name for the virtual server. Must begin with an ASCII alphanumeric or underscore (_) character, and must contain only ASCII alphanumeric, underscore, hash (#), period (.), space, colon (:), at sign (@), equal sign (=), and hyphen (-) characters. Can be changed after the virtual server is created. CLI Users: If the name includes

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
			one or more spaces, enclose the name in double or single quotation marks (for example, "my vserver" or 'my vserver'). . Minimum length = 1
gotopriorityexpression	<String>	Read-write	Expression specifying the priority of the next policy which will get evaluated if the current policy rule evaluates to TRUE.
invoke	<Boolean>	Read-write	Invoke policies bound to a virtual server or policy label.
labeltype	<String>	Read-write	The invocation type. Possible values = reqvserver, resvserver, policylabel
__count	<Double>	Read-write	count parameter

Operations

(click to see Properties)

ADD:| DELETE:| GET| GET (ALL)| COUNT

Some options that you can use for each operations:

- **Getting warnings in response:**NITRO allows you to get warnings in an operation by specifying the "warning" query parameter as "yes". For example, to get warnings while connecting to the Citrix ADC appliance, the URL is as follows:

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/login?warning=yes

If any, the warnings are displayed in the response payload with the HTTP code "209 X-NITRO-WARNING".

- **Authenticated access for individual NITRO operations:**NITRO allows you to logon to the Citrix ADC appliance to perform individual operations. You can use this option instead of creating a NITRO session (using the login object) and then using that session to perform all operations,

  To do this, you must specify the username and password in the request header of the NITRO request as follows:

  X-NITRO-USER:<username>

  X-NITRO-PASS:<password>

  Note: In such cases, make sure that the request header DOES not include the following:

  Cookie:NITRO_AUTH_TOKEN=<tokenvalue>

`*Note:`*

Mandatory parameters are marked in redand placeholder content is marked in <green>.

add

URL:http://<Citrix-ADC-IP-address(NSIP)/nitro/v1/config/lbvserver_videooptimizationdetectionpolicy_binding

HTTP Method:PUT

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Content-Type:application/json

Request Payload:

```
{

"lbvserver_videooptimizationdetectionpolicy_binding":{

<b>"name":<String_value>,
```

```
</b>"policyname":<String_value>,

"priority":<Double_value>,

"gotopriorityexpression":<String_value>,

"bindpoint":<String_value>,

"invoke":<Boolean_value>,

"labeltype":<String_value>,

"labelname":<String_value>

}}
```

Response:

HTTP Status Code on Success: 201 Created HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error

delete

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/lbvserver_videooptimizationdetectionpolicy_binding/name_value<String>

Query-parameters:

args

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/lbvserver_videooptimizationdetectionpolicy_binding/name_value<String>?**args=policyname:<String_value>,bindpoint:<String_value>,priority:<Double_value>**

HTTP Method:DELETE

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue>

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error

get

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/lbvserver_videooptimizationdetectionpolicy_binding/name_value<String>

Query-parameters:

filter

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/lbvserver_videooptimizationdetectionpolicy_binding/name_value<String>?**filter=property-name1:property-value1,property-name2:property-value2**

Use this query-parameter to get the filtered set of lbvserver_videooptimizationdetectionpolicy_binding resources configured on Citrix ADC. Filtering can be done on any of the properties of the resource.

pagination

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/lbvserver_videooptimizationdetectionpolicy_binding/name_value<String>?**pagesize=#no;pageno=#no**

Use this query-parameter to get the lbvserver_videooptimizationdetectionpolicy_binding resources in chunks.

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "lbvserver_videooptimizationdetectionpolicy_binding": [ {

"priority":<Double_value>,

"bindpoint":<String_value>,

"policyname":<String_value>,

"labelname":<String_value>,

"name":<String_value>,

"gotopriorityexpression":<String_value>,

"invoke":<Boolean_value>,

"labeltype":<String_value>

}]}
```

get (all)

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/lbvserver_videooptimizationdetectionpolicy_binding

Query-parameters:

bulkbindings

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/lbvserver_videooptimizationdetectionpolicy_binding?**bulkbindings=yes**

NITRO allows you to fetch bindings in bulk.

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "lbvserver_videooptimizationdetectionpolicy_binding": [ {

"priority":<Double_value>,

"bindpoint":<String_value>,

"policyname":<String_value>,

"labelname":<String_value>,

"name":<String_value>,

"gotopriorityexpression":<String_value>,

"invoke":<Boolean_value>,

"labeltype":<String_value>

}]}
```

count

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/lbvserver_videooptimizationdetectionpolicy_binding/name_value<String>?**count=yes**

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{"lbvserver_videooptimizationdetectionpolicy_binding": [ { "__count":
"#no"} ] }
```

lbvserver_videooptimizationpacingpolicy_binding

Binding object showing the videooptimizationpacingpolicy that can be bound to lbvserver.

Properties

(click to see Operations)

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
priority	<Double>	Read-write	Priority.
bindpoint	<String>	Read-write	The bindpoint to which the policy is bound. Possible values = REQUEST, RESPONSE
policyname	<String>	Read-write	Name of the policy bound to the LB vserver.
labelname	<String>	Read-write	Name of the label invoked.
name	<String>	Read-write	Name for the virtual server. Must begin with an ASCII

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
			alphanumeric or underscore (_) character, and must contain only ASCII alphanumeric, underscore, hash (#), period (.), space, colon (:), at sign (@), equal sign (=), and hyphen (-) characters. Can be changed after the virtual server is created. CLI Users: If the name includes one or more spaces, enclose the name in double or single quotation marks (for example, "my vserver" or 'my vserver'). . Minimum length = 1
gotopriorityexpression	<String>	Read-write	Expression specifying the priority of the next policy which will get evaluated if the current policy rule evaluates to TRUE.
invoke	<Boolean>	Read-write	Invoke policies bound to a virtual server or policy label.
labeltype	<String>	Read-write	The invocation type. Possible values = reqvserver,

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
			resvserver, policylabel
__count	<Double>	Read-write	count parameter

Operations

(click to see Properties)

ADD:| DELETE:| GET| GET (ALL)| COUNT

Some options that you can use for each operations:

- **Getting warnings in response:**NITRO allows you to get warnings in an operation by specifying the "warning" query parameter as "yes". For example, to get warnings while connecting to the Citrix ADC appliance, the URL is as follows:

  http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/login?warning=yes

  If any, the warnings are displayed in the response payload with the HTTP code "209 X-NITRO-WARNING".

- **Authenticated access for individual NITRO operations:**NITRO allows you to logon to the Citrix ADC appliance to perform individual operations. You can use this option instead of creating a NITRO session (using the login object) and then using that session to perform all operations,

  To do this, you must specify the username and password in the request header of the NITRO request as follows:

  X-NITRO-USER:<username>

  X-NITRO-PASS:<password>

  Note: In such cases, make sure that the request header DOES not include the following:

  Cookie:NITRO_AUTH_TOKEN=<tokenvalue>

`*Note:`*

Mandatory parameters are marked in redand placeholder content is marked in <green>.

add

URL:http://<Citrix-ADC-IP-address(NSIP)/nitro/v1/config/lbvserver_videooptimizationpacingpolicy_binding

HTTP Method:PUT

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Content-Type:application/json

Request Payload:

```
{

"lbvserver_videooptimizationpacingpolicy_binding":{

<b>"name":<String_value>,

</b>"policyname":<String_value>,

"priority":<Double_value>,

"gotopriorityexpression":<String_value>,

"bindpoint":<String_value>,

"invoke":<Boolean_value>,

"labeltype":<String_value>,

"labelname":<String_value>

}}
```

Response:

HTTP Status Code on Success: 201 Created HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error

delete

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/lbvserver_videooptimizationpacingpolicy_binding/name_value<String>

Query-parameters:

args

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/lbvserver_videooptimizationpacingpolicy_binding/name_value<String>?**args=policyname:<String_value>,bindpoint:<String_value>,priority:<Double_value>**

HTTP Method:DELETE

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue>

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error

get

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/lbvserver_videooptimizationpacingpolicy_binding/name_value<String>

Query-parameters:

filter

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/lbvserver_videooptimizationpacingpolicy_binding/name_value<String>?**filter=property-name1:property-value1,property-name2:property-value2**

Use this query-parameter to get the filtered set of lbvserver_videooptimizationpacingpolicy_binding resources configured on Citrix ADC. Filtering can be done on any of the properties of the resource.

pagination

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/lbvserver_videooptimizationpacingpolicy_binding/name_value<String>?**pagesize=#no;pageno=#no**

Use this query-parameter to get the lbvserver_videooptimizationpacingpolicy_binding resources in chunks.

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "lbvserver_videooptimizationpacingpolicy_binding": [ {

"priority":<Double_value>,

"bindpoint":<String_value>,

"policyname":<String_value>,

"labelname":<String_value>,

"name":<String_value>,

"gotopriorityexpression":<String_value>,

"invoke":<Boolean_value>,

"labeltype":<String_value>

}]}
```

get (all)

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/lbvserver_videooptimizationpacingpolicy_binding

Query-parameters:

bulkbindings

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/lbvserver_videooptimizationpacingpolicy_binding?**bulkbindings=yes**

NITRO allows you to fetch bindings in bulk.

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "lbvserver_videooptimizationpacingpolicy_binding": [ {

"priority":<Double_value>,

"bindpoint":<String_value>,

"policyname":<String_value>,

"labelname":<String_value>,

"name":<String_value>,
```

```
"gotopriorityexpression":<String_value>,

"invoke":<Boolean_value>,

"labeltype":<String_value>

}]}
```

count

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/lbvserver_videooptimizationpacingpolicy_binding/name_value<String>?**count=yes**

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{"lbvserver_videooptimizationpacingpolicy_binding": [ { "__count":
"#no"} ] }
```

LSN

Lsn

Large Scale NAT commands

NAME	DESCRIPTION
lsnappsattributes	Configuration for LSN Application Attributes
lsnappsprofile	Configuration for LSN Application Profile
lsnappsprofile_binding	Binding object showing the resources that can be bound to lsnappsprofile
lsnappsprofile_lsnappsattributes_binding	Binding object showing the lsnappsattributes that can be bound to lsnappsprofile
lsnappsprofile_port_binding	Binding object showing the port that can be bound to lsnappsprofile
lsnclient	Configuration for lsn client
lsnclient_binding	Binding object showing the resources that can be bound to lsnclient
lsnclient_network6_binding	Binding object showing the network6 that can be bound to lsnclient
lsnclient_network_binding	Binding object showing the network that can be bound to lsnclient
lsnclient_nsacl6_binding	Binding object showing the nsacl6 that can be bound to lsnclient
lsnclient_nsacl_binding	Binding object showing the nsacl that can be bound to lsnclient
lsndeterministicnat	Configuration for deterministic NAT
lsngroup	Configuration for LSN group
lsngroup_binding	Binding object showing the resources that can be bound to lsngroup
lsngroup_ipsecalgprofile_binding	Binding object showing the ipsecalgprofile that can be bound to lsngroup
lsngroup_lsnappsprofile_binding	Binding object showing the lsnappsprofile that can be bound to lsngroup

NAME	DESCRIPTION
lsngroup_lsnhttphdrlogprofile_binding	Binding object showing the lsnhttphdrlogprofile that can be bound to lsngroup
lsngroup_lsnlogprofile_binding	Binding object showing the lsnlogprofile that can be bound to lsngroup
lsngroup_lsnpool_binding	Binding object showing the lsnpool that can be bound to lsngroup
lsngroup_lsnrtspalgprofile_binding	Binding object showing the lsnrtspalgprofile that can be bound to lsngroup
lsngroup_lsnsipalgprofile_binding	Binding object showing the lsnsipalgprofile that can be bound to lsngroup
lsngroup_lsntransportprofile_binding	Binding object showing the lsntransportprofile that can be bound to lsngroup
lsngroup_pcpserver_binding	Binding object showing the pcpserver that can be bound to lsngroup
lsnhttphdrlogprofile	Configuration for LSN HTTP header logging Profile
lsnip6profile	Configuration for LSN ip6 Profile
lsnlogprofile	Configuration for LSN logging Profile
lsnparameter	Configuration for LSN parameter
lsnpool	Configuration for LSN pool
lsnpool_binding	Binding object showing the resources that can be bound to lsnpool
lsnpool_lsnip_binding	Binding object showing the lsnip that can be bound to lsnpool
lsnrtspalgprofile	Configuration for LSN RTSPALG Profile
lsnrtspalgsession	Configuration for LSN RTSPALG session

NAME	DESCRIPTION
lsnrtspalgsession_binding	Binding object showing the resources that can be bound to lsnrtspalgsession
lsnrtspalgsession_datachannel_binding	Binding object showing the datachannel that can be bound to lsnrtspalgsession
lsnsession	Configuration for lsn session
lsnsipalgcall	Configuration for LSN SIPALG call
lsnsipalgcall_binding	Binding object showing the resources that can be bound to lsnsipalgcall
lsnsipalgcall_controlchannel_binding	Binding object showing the controlchannel that can be bound to lsnsipalgcall
lsnsipalgcall_datachannel_binding	Binding object showing the datachannel that can be bound to lsnsipalgcall
lsnsipalgprofile	Configuration for LSN SIPALG Profile
lsnstatic	Configuration for static mapping
lsntransportprofile	Configuration for LSN Transport Profile

lsnappsprofile

Configuration for LSN Application Profile resource.

Properties

(click to see Operations)

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
appsprofilename	<String>	Read-write	Name for the LSN application profile. Must begin with an ASCII alphanumeric or underscore (_) character, and must contain only ASCII

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
			alphanumeric, underscore, hash (#), period (.), space, colon (:), at (@), equals (=), and hyphen (-) characters. Cannot be changed after the LSN application profile is created. The following requirement applies only to the Citrix ADC CLI: If the name includes one or more spaces, enclose the name in double or single quotation marks (for example, "lsn application profile1" or 'lsn application profile1'). Minimum length = 1 Maximum length = 127
transportprotocol	<String>	Read-write	Name of the protocol for which the parameters of this LSN application profile applies. Possible values = TCP, UDP, ICMP
ippooling	<String>	Read-write	NAT IP address allocation options for sessions associated

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
			with the same subscriber. Available options function as follows: * Paired - The Citrix ADC allocates the same NAT IP address for all sessions associated with the same subscriber. When all the ports of a NAT IP address are used in LSN sessions (for same or multiple subscribers), the Citrix ADC ADC drops any new connection from the subscriber. * Random - The Citrix ADC allocates random NAT IP addresses, from the pool, for different sessions associated with the same subscriber. This parameter is applicable to dynamic NAT allocation only.

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
			Default value: RANDOM Possible values = PAIRED, RANDOM
mapping	<String>	Read-write	Type of LSN mapping to apply to subsequent packets originating from the same subscriber IP address and port. Consider an example of an LSN mapping that includes the mapping of the subscriber IP:port (X:x), NAT IP:port (N:n), and external host IP:port (Y:y). Available options function as follows: * ENDPOINT-INDEPENDENT - Reuse the LSN mapping for subsequent packets sent from the same subscriber IP address and port (X:x) to any external IP address and port.

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
			* ADDRESS-DEPENDENT - Reuse the LSN mapping for subsequent packets sent from the same subscriber IP address and port (X:x) to the same external IP address (Y), regardless of the external port. * ADDRESS-PORT-DEPENDENT - Reuse the LSN mapping for subsequent packets sent from the same internal IP address and port (X:x) to the same external IP address and port (Y:y) while the mapping is still active. Default value: ADDRESS-PORT-DEPENDENT Possible values = ENDPOINT-INDEPENDENT, ADDRESS-DEPENDENT,

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
			ADDRESS-PORT-DEPENDENT
filtering	<String>	Read-write	Type of filter to apply to packets originating from external hosts. Consider an example of an LSN mapping that includes the mapping of subscriber IP:port (X:x), NAT IP:port (N:n), and external host IP:port (Y:y). Available options function as follows: * ENDPOINT INDEPENDENT - Filters out only packets not destined to the subscriber IP address and port X:x, regardless of the external host IP address and port source (Z:z). The Citrix ADC forwards any packets destined to X:x. In other words, sending packets from the subscriber to any external IP

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
			address is sufficient to allow packets from any external hosts to the subscriber. * ADDRESS DEPENDENT - Filters out packets not destined to subscriber IP address and port X:x. In addition, the ADC filters out packets from Y:y destined for the subscriber (X:x) if the client has not previously sent packets to Y:anyport (external port independent). In other words, receiving packets from a specific external host requires that the subscriber first send packets to that specific external host's IP address. * ADDRESS PORT DEPENDENT (the default) - Filters out packets not destined to

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
			subscriber IP address and port (X:x). In addition, the Citrix ADC filters out packets from Y:y destined for the subscriber (X:x) if the subscriber has not previously sent packets to Y:y. In other words, receiving packets from a specific external host requires that the subscriber first send packets first to that external IP address and port. Default value: ADDRESS-PORT-DEPENDENT Possible values = ENDPOINT-INDEPENDENT, ADDRESS-DEPENDENT, ADDRESS-PORT-DEPENDENT
tcpproxy	<String>	Read-write	Enable TCP proxy, which enables the Citrix ADC to optimize the TCP

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
			traffic by using Layer 4 features. Default value: DISABLED Possible values = ENABLED, DISABLED
td	<Double>	Read-write	ID of the traffic domain through which the Citrix ADC sends the outbound traffic after performing LSN. If you do not specify an ID, the ADC sends the outbound traffic through the default traffic domain, which has an ID of 0. Default value: 4095
l2info	<String>	Read-write	Enable l2info by creating natpcbs for LSN, which enables the Citrix ADC to use L2CONN/MBF with LSN. Default value: DISABLED Possible values = ENABLED, DISABLED
__count	<Double>	Read-only	count parameter

Operations

(click to see Properties)

ADD| DELETE| UPDATE| UNSET| GET (ALL)| GET| COUNT

Some options that you can use for each operations:

- **Getting warnings in response:**NITRO allows you to get warnings in an operation by specifying the "warning" query parameter as "yes". For example, to get warnings while connecting to the Citrix ADC appliance, the URL is as follows:

  http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/login?warning=yes

  If any, the warnings are displayed in the response payload with the HTTP code "209 X-NITRO-WARNING".

- **Authenticated access for individual NITRO operations:**NITRO allows you to logon to the Citrix ADC appliance to perform individual operations. You can use this option instead of creating a NITRO session (using the login object) and then using that session to perform all operations,

  To do this, you must specify the username and password in the request header of the NITRO request as follows:

  X-NITRO-USER:<username>

  X-NITRO-PASS:<password>

  Note: In such cases, make sure that the request header DOES not include the following:

  Cookie:NITRO_AUTH_TOKEN=<tokenvalue>

`*Note:`*

Mandatory parameters are marked in redand placeholder content is marked in <green>.

add

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/lsnappsprofile

HTTP Method:POST

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Content-Type:application/json

Request Payload:

```
{"lsnappsprofile":{

<b>"appsprofilename":<String_value>,

</b><b>"transportprotocol":<String_value>,

</b>"ippooling":<String_value>,

"mapping":<String_value>,

"filtering":<String_value>,

"tcpproxy":<String_value>,

"td":<Double_value>,

"l2info":<String_value>

}}
```

Response:

HTTP Status Code on Success: 201 Created HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error

delete

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/lsnappsprofile/appsprofilename_value<String>

HTTP Method:DELETE

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue>

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error

update

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/lsnappsprofile

HTTP Method:PUT

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Content-Type:application/json

Request Payload:

```
{"lsnappsprofile":{

<b>"appsprofilename":<String_value>,

</b>"ippooling":<String_value>,

"mapping":<String_value>,

"filtering":<String_value>,

"tcpproxy":<String_value>,

"td":<Double_value>,

"l2info":<String_value>

}}
```

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error

unset

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/lsnappsprofile?**action=unset**

HTTP Method:POST

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Content-Type:application/json

Request Payload:

```
{"lsnappsprofile":{

<b>"appsprofilename":<String_value>,

</b>"ippooling":true,

"mapping":true,

"filtering":true,

"tcpproxy":true,

"td":true,

"l2info":true

}}
```

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error

get (all)

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/lsnappsprofile

Query-parameters:

attrs

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/lsnappsprofile?**attrs=property-name1,property-name2**

Use this query parameter to specify the resource details that you want to retrieve.

filter

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/lsnappsprofile?**filter=property-name1:property-val1,property-name2:property-val2**

Use this query-parameter to get the filtered set of lsnappsprofile resources configured on Citrix ADC. Filtering can be done on any of the properties of the resource.

view

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/lsnappsprofile?**view=summary**

Note: By default, the retrieved results are displayed in detail view (?view=detail).

pagination

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/lsnappsprofile?**pagesize=#no;pageno=#no**

Use this query-parameter to get the lsnappsprofile resources in chunks.

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "lsnappsprofile": [ {

"appsprofilename":<String_value>,

"transportprotocol":<String_value>,

"ippooling":<String_value>,

"mapping":<String_value>,

"filtering":<String_value>,

"tcpproxy":<String_value>,

"td":<Double_value>,

"l2info":<String_value>

}]}
```

get

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/lsnappsprofile/appsprofilename_value<String>

Query-parameters:

attrs

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/lsnappsprofile/appsprofilename_value<String>?**attrs=property-name1,property-name2**

Use this query parameter to specify the resource details that you want to retrieve.

view

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/lsnappsprofile/appsprofilename_value<String>?**view=summary**

Note: By default, the retrieved results are displayed in detail view (?view=detail).

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "lsnappsprofile": [ {

"appsprofilename":<String_value>,

"transportprotocol":<String_value>,

"ippooling":<String_value>,

"mapping":<String_value>,

"filtering":<String_value>,

"tcpproxy":<String_value>,

"td":<Double_value>,

"l2info":<String_value>

}]}
```

count

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/lsnappsprofile?**count=yes**

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "lsnappsprofile": [ { "__count": "#no"} ] }
```

lsnappsprofile_binding

Binding object which returns the resources bound to lsnappsprofile.

Properties

(click to see Operations)

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
appsprofilename	<String>	Read-write	Name for the LSN application profile. Must begin with an ASCII alphanumeric or underscore (_) character, and must contain only ASCII alphanumeric, underscore, hash (#), period (.), space, colon (:), at (@), equals (=), and hyphen (-) characters. Cannot

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
			be changed after the LSN application profile is created. The following requirement applies only to the Citrix ADC CLI: If the name includes one or more spaces, enclose the name in double or single quotation marks (for example, "lsn application profile1" or 'lsn application profile1'). Minimum length = 1 Maximum length = 127
lsnappsprofile_lsnappsattributes_binding	<lsnappsprofile_lsnappsattributes_binding[]>	Read-only	lsnappsattributes that can be bound to lsnappsprofile.
lsnappsprofile_port_binding	<lsnappsprofile_port_binding[]>	Read-only	port that can be bound to lsnappsprofile.

Operations

(click to see Properties)

GET| GET (ALL)

Some options that you can use for each operations:

- **Getting warnings in response:**NITRO allows you to get warnings in an operation by specifying the "warning" query parameter as "yes". For example, to get warnings while connecting to the Citrix ADC appliance, the URL is as follows:

  http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/login?warning=yes

If any, the warnings are displayed in the response payload with the HTTP code "209 X-NITRO-WARNING".

- **Authenticated access for individual NITRO operations:**NITRO allows you to logon to the Citrix ADC appliance to perform individual operations. You can use this option instead of creating a NITRO session (using the login object) and then using that session to perform all operations,

  To do this, you must specify the username and password in the request header of the NITRO request as follows:

  X-NITRO-USER:<username>

  X-NITRO-PASS:<password>

  Note: In such cases, make sure that the request header DOES not include the following:

  Cookie:NITRO_AUTH_TOKEN=<tokenvalue>

`*Note:`*

Mandatory parameters are marked in redand placeholder content is marked in <green>.

get

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/lsnappsprofile_binding/appsprofilename_value<String>

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "lsnappsprofile_binding": [ {

"appsprofilename":<String_value>,

"lsnappsprofile_port_binding":<lsnappsprofile_port_binding[]_value>,

"lsnappsprofile_lsnappsattributes_binding":<lsnappsprofile_lsnappsattributes_binding[]_value>

}]}
```

get (all)

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/lsnappsprofile_binding

Query-parameters:

bulkbindings

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/lsnappsprofile_binding?**bulkbindings=yes**

NITRO allows you to fetch bindings in bulk.

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "lsnappsprofile_binding": [ {

"appsprofilename":<String_value>,

"lsnappsprofile_port_binding":<lsnappsprofile_port_binding[]_value>,

"lsnappsprofile_lsnappsattributes_binding":<lsnappsprofile_lsnappsattributes_binding[]_value>

}]}
```

lsnappsprofile_port_binding

Binding object showing the port that can be bound to lsnappsprofile.

Properties

(click to see Operations)

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
appsprofilename	<String>	Read-write	Name for the LSN application profile. Must begin with an ASCII alphanumeric or underscore (_) character, and must contain only ASCII alphanumeric, underscore, hash (#), period (.), space, colon (:), at (@), equals (=), and hyphen (-) characters. Cannot be changed after the LSN application profile is created. The following requirement applies

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
			only to the Citrix ADC CLI: If the name includes one or more spaces, enclose the name in double or single quotation marks (for example, "lsn application profile1" or 'lsn application profile1'). Minimum length = 1 Maximum length = 127
lsnport	<String>	Read-write	Port numbers or range of port numbers to match against the destination port of the incoming packet from a subscriber. When the destination port is matched, the LSN application profile is applied for the LSN session. Separate a range of ports with a hyphen. For example, 40-90. Minimum length = 1
__count	<Double>	Read-write	count parameter

Operations

(click to see Properties)

ADD:| DELETE:| GET| GET (ALL)| COUNT

Some options that you can use for each operations:

- **Getting warnings in response:**NITRO allows you to get warnings in an operation by specifying the "warning" query parameter as "yes". For example, to get warnings while connecting to the Citrix ADC appliance, the URL is as follows:

  http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/login?warning=yes

  If any, the warnings are displayed in the response payload with the HTTP code "209 X-NITRO-WARNING".

- **Authenticated access for individual NITRO operations:**NITRO allows you to logon to the Citrix ADC appliance to perform individual operations. You can use this option instead of creating a NITRO session (using the login object) and then using that session to perform all operations,

  To do this, you must specify the username and password in the request header of the NITRO request as follows:

  X-NITRO-USER:<username>

  X-NITRO-PASS:<password>

  Note: In such cases, make sure that the request header DOES not include the following:

  Cookie:NITRO_AUTH_TOKEN=<tokenvalue>

`*Note:`*

Mandatory parameters are marked in redand placeholder content is marked in <green>.

add

URL:http://<Citrix-ADC-IP-address(NSIP)/nitro/v1/config/lsnappsprofile_port_binding

HTTP Method:PUT

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Content-Type:application/json

Request Payload:

```
{

"lsnappsprofile_port_binding":{

<b>"appsprofilename":<String_value>,

</b>"lsnport":<String_value>

}}
```

Response:

HTTP Status Code on Success: 201 Created HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error

delete

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/lsnappsprofile_port_binding/appsprofilename_value<String>

Query-parameters:

args

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/lsnappsprofile_port_binding/appsprofilename_value<String>?**args=lsnport:<String_value>**

HTTP Method:DELETE

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue>

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error

get

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/lsnappsprofile_port_binding/appsprofilename_value<String>

Query-parameters:

filter

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/lsnappsprofile_port_binding/appsprofilename_value<String>?**filter=property-name1:property-value1,property-name2:property-value2**

Use this query-parameter to get the filtered set of lsnappsprofile_port_binding resources configured on Citrix ADC. Filtering can be done on any of the properties of the resource.

pagination

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/lsnappsprofile_port_binding/appsprofilename_value<String>?**pagesize=#no;pageno=#no**

Use this query-parameter to get the lsnappsprofile_port_binding resources in chunks.

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "lsnappsprofile_port_binding": [ {

"appsprofilename":<String_value>,
```

```
"lsnport":<String_value>

}]}
```

get (all)

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/lsnappsprofile_port_binding

Query-parameters:

bulkbindings

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/lsnappsprofile_port_binding?**bulkbindings=yes**

NITRO allows you to fetch bindings in bulk.

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "lsnappsprofile_port_binding": [ {

"appsprofilename":<String_value>,

"lsnport":<String_value>

}]}
```

count

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/lsnappsprofile_port_binding/appsprofilename_value<String>?**count=yes**

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{"lsnappsprofile_port_binding": [ { "__count": "#no"} ] }
```

lsnclient

Configuration for lsn client resource.

Properties

(click to see Operations)

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
clientname	<String>	Read-write	Name for the LSN client entity. Must begin with an ASCII alphanumeric or underscore (_) character, and must

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
			contain only ASCII alphanumeric, underscore, hash (#), period (.), space, colon (:), at (@), equals (=), and hyphen (-) characters. Cannot be changed after the LSN client is created. The following requirement applies only to the Citrix ADC CLI: If the name includes one or more spaces, enclose the name in double or single quotation marks (for example, "lsn client1" or 'lsn client1'). . Minimum length = 1 Maximum length = 127
__count	<Double>	Read-only	count parameter

Operations

(click to see Properties)

ADD| DELETE| GET (ALL)| GET| COUNT

Some options that you can use for each operations:

- **Getting warnings in response:**NITRO allows you to get warnings in an operation by specifying the "warning" query parameter as "yes". For example, to get warnings while connecting to the Citrix ADC appliance, the URL is as follows:

  http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/login?warning=yes

If any, the warnings are displayed in the response payload with the HTTP code "209 X-NITRO-WARNING".

- **Authenticated access for individual NITRO operations:**NITRO allows you to logon to the Citrix ADC appliance to perform individual operations. You can use this option instead of creating a NITRO session (using the login object) and then using that session to perform all operations,

  To do this, you must specify the username and password in the request header of the NITRO request as follows:

  X-NITRO-USER:<username>

  X-NITRO-PASS:<password>

  Note: In such cases, make sure that the request header DOES not include the following:

  Cookie:NITRO_AUTH_TOKEN=<tokenvalue>

`*Note:`*

Mandatory parameters are marked in redand placeholder content is marked in <green>.

add

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/lsnclient

HTTP Method:POST

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Content-Type:application/json

Request Payload:

```
{"lsnclient":{

<b>"clientname":<String_value>

</b>}}
```

Response:

HTTP Status Code on Success: 201 Created HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error

delete

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/lsnclient/clientname_value<String>

HTTP Method:DELETE

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue>

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error

get (all)

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/lsnclient

Query-parameters:

attrs

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/lsnclient?**attrs=property-name1,property-name2**

Use this query parameter to specify the resource details that you want to retrieve.

filter

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/lsnclient?**filter=property-name1:property-val1,property-name2:property-val2**

Use this query-parameter to get the filtered set of lsnclient resources configured on Citrix ADC. Filtering can be done on any of the properties of the resource.

view

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/lsnclient?**view=summary**

Note: By default, the retrieved results are displayed in detail view (?view=detail).

pagination

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/lsnclient?**pagesize=#no;pageno=#no**

Use this query-parameter to get the lsnclient resources in chunks.

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "lsnclient": [ {

"clientname":<String_value>,



}]}
```

get

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/lsnclient/clientname_value<String>

Query-parameters:

attrs

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/lsnclient/clientname_value<String>?**attrs=property-name1,property-name2**

Use this query parameter to specify the resource details that you want to retrieve.

view

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/lsnclient/clientname_value<String>?**view=summary**

Note: By default, the retrieved results are displayed in detail view (?view=detail).

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "lsnclient": [ {

"clientname":<String_value>,



}]}
```

count

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/lsnclient?**count=yes**

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "lsnclient": [ { "__count": "#no"} ] }
```

lsnclient_binding

Binding object which returns the resources bound to lsnclient.

Properties

(click to see Operations)

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
clientname	<String>	Read-write	Name for the LSN client entity. Must begin with an ASCII alphanumeric or underscore (_) character, and must contain only ASCII alphanumeric, underscore, hash (#), period (.), space, colon (:), at (@), equals (=), and hyphen (-) characters. Cannot be changed after the LSN client is created. The following requirement applies only to the Citrix ADC CLI: If the

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
			name includes one or more spaces, enclose the name in double or single quotation marks (for example, "lsn client1" or 'lsn client1'). . Minimum length = 1 Maximum length = 127
lsnclient_network_binding	<lsnclient_network_binding[]>	Read-only	network that can be bound to lsnclient.
lsnclient_network6_binding	<lsnclient_network6_binding[]>	Read-only	network6 that can be bound to lsnclient.
lsnclient_nsacl6_binding	<lsnclient_nsacl6_binding[]>	Read-only	nsacl6 that can be bound to lsnclient.
lsnclient_nsacl_binding	<lsnclient_nsacl_binding[]>	Read-only	nsacl that can be bound to lsnclient.

Operations

(click to see Properties)

GET| GET (ALL)

Some options that you can use for each operations:

- **Getting warnings in response:**NITRO allows you to get warnings in an operation by specifying the "warning" query parameter as "yes". For example, to get warnings while connecting to the Citrix ADC appliance, the URL is as follows:

  http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/login?warning=yes

  If any, the warnings are displayed in the response payload with the HTTP code "209 X-NITRO-WARNING".

- **Authenticated access for individual NITRO operations:**NITRO allows you to logon to the Citrix ADC appliance to perform individual operations. You can use this option instead of creating a NITRO session (using the login object) and then using that session to perform all operations,

  To do this, you must specify the username and password in the request header of the NITRO request as follows:

  X-NITRO-USER:<username>

  X-NITRO-PASS:<password>

  Note: In such cases, make sure that the request header DOES not include the following:

  Cookie:NITRO_AUTH_TOKEN=<tokenvalue>

`*Note:`*

Mandatory parameters are marked in redand placeholder content is marked in <green>.

get

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/lsnclient_binding/clientname_value<String>

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "lsnclient_binding": [ {
```

```
"clientname":<String_value>,

"lsnclient_nsacl6_binding":<lsnclient_nsacl6_binding[]_value>,

"lsnclient_network_binding":<lsnclient_network_binding[]_value>,

"lsnclient_network6_binding":<lsnclient_network6_binding[]_value>,

"lsnclient_nsacl_binding":<lsnclient_nsacl_binding[]_value>

}]}
```

get (all)

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/lsnclient_binding

Query-parameters:

bulkbindings

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/lsnclient_binding?**bulkbindings=yes**

NITRO allows you to fetch bindings in bulk.

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "lsnclient_binding": [ {
```

```
"clientname":<String_value>,

"lsnclient_nsacl6_binding":<lsnclient_nsacl6_binding[]_value>,

"lsnclient_network_binding":<lsnclient_network_binding[]_value>,

"lsnclient_network6_binding":<lsnclient_network6_binding[]_value>,

"lsnclient_nsacl_binding":<lsnclient_nsacl_binding[]_value>

}]}
```

lsnclient_network6_binding

Binding object showing the network6 that can be bound to lsnclient.

Properties

(click to see Operations)

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
network	<String>	Read-write	IPv4 address(es) of the LSN subscriber(s) or subscriber network(s) on whose traffic you want the Citrix ADC to perform Large Scale NAT. Minimum length = 1
clientname	<String>	Read-write	Name for the LSN client entity. Must begin with an ASCII alphanumeric or underscore (_) character, and must contain only ASCII alphanumeric, underscore, hash

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
			(#), period (.), space, colon (:), at (@), equals (=), and hyphen (-) characters. Cannot be changed after the LSN client is created. The following requirement applies only to the Citrix ADC CLI: If the name includes one or more spaces, enclose the name in double or single quotation marks (for example, "lsn client1" or 'lsn client1'). . Minimum length = 1 Maximum length = 127
network6	<String>	Read-write	IPv6 address(es) of the LSN subscriber(s) or subscriber network(s) on whose traffic you want the Citrix ADC to perform Large Scale NAT. Minimum length = 1
td	<Double>	Read-write	ID of the traffic domain on which this subscriber or the subscriber network (as specified by the network parameter) belongs. If you do

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
			not specify an ID, the subscriber or the subscriber network becomes part of the default traffic domain. Default value: 0 Minimum value = 0 Maximum value = 4094
netmask	<String>	Read-write	Subnet mask for the IPv4 address specified in the Network parameter.
__count	<Double>	Read-write	count parameter

Operations

(click to see Properties)

ADD:| DELETE:| GET| GET (ALL)| COUNT

Some options that you can use for each operations:

- **Getting warnings in response:**NITRO allows you to get warnings in an operation by specifying the "warning" query parameter as "yes". For example, to get warnings while connecting to the Citrix ADC appliance, the URL is as follows:

  http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/login?warning=yes

  If any, the warnings are displayed in the response payload with the HTTP code "209 X-NITRO-WARNING".

- **Authenticated access for individual NITRO operations:**NITRO allows you to logon to the Citrix ADC appliance to perform individual operations. You can use this option instead of creating a NITRO session (using the login object) and then using that session to perform all operations,

To do this, you must specify the username and password in the request header of the NITRO request as follows:

X-NITRO-USER:<username>

X-NITRO-PASS:<password>

Note: In such cases, make sure that the request header DOES not include the following:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue>

`*Note:`*

Mandatory parameters are marked in redand placeholder content is marked in <green>.

add

URL:http://<Citrix-ADC-IP-address(NSIP)/nitro/v1/config/lsnclient_network6_binding

HTTP Method:PUT

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Content-Type:application/json

Request Payload:

```
{

"lsnclient_network6_binding":{

<b>"clientname":<String_value>,

</b>"network":<String_value>,

"netmask":<String_value>,

"network6":<String_value>,

"td":<Double_value>

}}
```

Response:

HTTP Status Code on Success: 201 Created HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error

delete

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/lsnclient_network6_binding/clientname_value<String>

Query-parameters:

args

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/lsnclient_network6_binding/clientname_value<String>?**args=network:<String_value>,netmask:<String_value>,network6:<String_value>,td:<Double_value>**

HTTP Method:DELETE

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue>

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error

get

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/lsnclient_network6_binding/clientname_value<String>

Query-parameters:

filter

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/lsnclient_network6_binding/clientname_value<String>?**filter=property-name1:property-value1,property-name2:property-value2**

Use this query-parameter to get the filtered set of lsnclient_network6_binding resources configured on Citrix ADC. Filtering can be done on any of the properties of the resource.

pagination

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/lsnclient_network6_binding/clientname_value<String>?**pagesize=#no;pageno=#no**

Use this query-parameter to get the lsnclient_network6_binding resources in chunks.

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "lsnclient_network6_binding": [ {

"network":<String_value>,

"clientname":<String_value>,

"network6":<String_value>,

"td":<Double_value>,

"netmask":<String_value>

}]}
```

get (all)

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/lsnclient_network6_binding

Query-parameters:

bulkbindings

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/lsnclient_network6_binding?**bulkbindings=yes**

NITRO allows you to fetch bindings in bulk.

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "lsnclient_network6_binding": [ {

"network":<String_value>,

"clientname":<String_value>,

"network6":<String_value>,

"td":<Double_value>,

"netmask":<String_value>

}]}
```

count

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/lsnclient_network6_binding/clientname_value<String>?**count=yes**

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{"lsnclient_network6_binding": [ { "__count": "#no"} ] }
```

lsnclient_network_binding

Binding object showing the network that can be bound to lsnclient.

Properties

(click to see Operations)

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
network	<String>	Read-write	IPv4 address(es) of the LSN subscriber(s) or subscriber network(s) on whose traffic you want the

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
			Citrix ADC to perform Large Scale NAT. Minimum length = 1
clientname	<String>	Read-write	Name for the LSN client entity. Must begin with an ASCII alphanumeric or underscore (_) character, and must contain only ASCII alphanumeric, underscore, hash (#), period (.), space, colon (:), at (@), equals (=), and hyphen (-) characters. Cannot be changed after the LSN client is created. The following requirement applies only to the Citrix ADC CLI: If the name includes one or more spaces, enclose the name in double or single quotation marks (for example, "lsn client1" or 'lsn client1'). . Minimum length = 1 Maximum length = 127
td	<Double>	Read-write	ID of the traffic domain on which this subscriber or the subscriber network

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
			(as specified by the network parameter) belongs. If you do not specify an ID, the subscriber or the subscriber network becomes part of the default traffic domain. Default value: 0 Minimum value = 0 Maximum value = 4094
netmask	<String>	Read-write	Subnet mask for the IPv4 address specified in the Network parameter.
__count	<Double>	Read-write	count parameter

Operations

(click to see Properties)

ADD:| DELETE:| GET| GET (ALL)| COUNT

Some options that you can use for each operations:

- **Getting warnings in response:**NITRO allows you to get warnings in an operation by specifying the "warning" query parameter as "yes". For example, to get warnings while connecting to the Citrix ADC appliance, the URL is as follows:

  http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/login?warning=yes

  If any, the warnings are displayed in the response payload with the HTTP code "209 X-NITRO-WARNING".

- **Authenticated access for individual NITRO operations:**NITRO allows you to logon to the Citrix ADC appliance to perform individual operations. You can use this option instead of

creating a NITRO session (using the login object) and then using that session to perform all operations,

To do this, you must specify the username and password in the request header of the NITRO request as follows:

X-NITRO-USER:<username>

X-NITRO-PASS:<password>

Note: In such cases, make sure that the request header DOES not include the following:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue>

`*Note:`*

Mandatory parameters are marked in redand placeholder content is marked in <green>.

add

URL:http://<Citrix-ADC-IP-address(NSIP)/nitro/v1/config/lsnclient_network_binding

HTTP Method:PUT

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Content-Type:application/json

Request Payload:

```
{

"lsnclient_network_binding":{

<b>"clientname":<String_value>,

</b>"network":<String_value>,

"netmask":<String_value>,

"td":<Double_value>

}}
```

Response:

HTTP Status Code on Success: 201 Created HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error

delete

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/lsnclient_network_binding/clientname_value<String>

Query-parameters:

args

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/lsnclient_network_binding/clientname_value<String>?**args=network:<String_value>,netmask:<String_value>,td:<Double_value>**

HTTP Method:DELETE

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue>

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error

get

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/lsnclient_network_binding/clientname_value<String>

Query-parameters:

filter

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/lsnclient_network_binding/clientname_value<String>?**filter=property-name1:property-value1,property-name2:property-value2**

Use this query-parameter to get the filtered set of lsnclient_network_binding resources configured on Citrix ADC. Filtering can be done on any of the properties of the resource.

pagination

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/lsnclient_network_binding/clientname_value<String>?**pagesize=#no;pageno=#no**

Use this query-parameter to get the lsnclient_network_binding resources in chunks.

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "lsnclient_network_binding": [ {

"network":<String_value>,

"clientname":<String_value>,

"td":<Double_value>,

"netmask":<String_value>

}]}
```

get (all)

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/lsnclient_network_binding

Query-parameters:

bulkbindings

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/lsnclient_network_binding?**bulkbindings=yes**

NITRO allows you to fetch bindings in bulk.

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "lsnclient_network_binding": [ {

"network":<String_value>,

"clientname":<String_value>,

"td":<Double_value>,

"netmask":<String_value>

}]}
```

count

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/lsnclient_network_binding/clientname_value<String>?**count=yes**

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{"lsnclient_network_binding": [ { "__count": "#no"} ] }
```

lsnclient_nsacl6_binding

Binding object showing the nsacl6 that can be bound to lsnclient.

Properties

(click to see Operations)

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
clientname	<String>	Read-write	Name for the LSN client entity. Must begin with an ASCII alphanumeric or underscore (_) character, and must contain only ASCII alphanumeric, underscore, hash (#), period (.), space, colon (:), at (@), equals (=), and hyphen (-) characters. Cannot

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
			be changed after the LSN client is created. The following requirement applies only to the Citrix ADC CLI: If the name includes one or more spaces, enclose the name in double or single quotation marks (for example, "lsn client1" or 'lsn client1'). . Minimum length = 1 Maximum length = 127
acl6name	<String>	Read-write	Name of any configured extended ACL6 whose action is ALLOW. The condition specified in the extended ACL6 rule is used as the condition for the LSN client. Minimum length = 1
td	<Double>	Read-write	ID of the traffic domain on which this subscriber or the subscriber network (as specified by the network parameter) belongs. If you do not specify an ID, the subscriber or the subscriber network becomes part of the

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
			default traffic domain. Default value: 0 Minimum value = 0 Maximum value = 4094
__count	<Double>	Read-write	count parameter

Operations

(click to see Properties)

ADD:| DELETE:| GET| GET (ALL)| COUNT

Some options that you can use for each operations:

- **Getting warnings in response:**NITRO allows you to get warnings in an operation by specifying the "warning" query parameter as "yes". For example, to get warnings while connecting to the Citrix ADC appliance, the URL is as follows:

  http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/login?warning=yes

  If any, the warnings are displayed in the response payload with the HTTP code "209 X-NITRO-WARNING".

- **Authenticated access for individual NITRO operations:**NITRO allows you to logon to the Citrix ADC appliance to perform individual operations. You can use this option instead of creating a NITRO session (using the login object) and then using that session to perform all operations,

  To do this, you must specify the username and password in the request header of the NITRO request as follows:

  X-NITRO-USER:<username>

  X-NITRO-PASS:<password>

  Note: In such cases, make sure that the request header DOES not include the following:

  Cookie:NITRO_AUTH_TOKEN=<tokenvalue>

`*Note:` *

Mandatory parameters are marked in redand placeholder content is marked in <green>.

add

URL:http://<Citrix-ADC-IP-address(NSIP)/nitro/v1/config/lsnclient_nsacl6_binding

HTTP Method:PUT

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Content-Type:application/json

Request Payload:

```
{

"lsnclient_nsacl6_binding":{

<b>"clientname":<String_value>,

</b>"acl6name":<String_value>,

"td":<Double_value>

}}
```

Response:

HTTP Status Code on Success: 201 Created HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error

delete

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/lsnclient_nsacl6_binding/clientname_value<String>

Query-parameters:

args

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/lsnclient_nsacl6_binding/clientname_value<String>?**args=acl6name:<String_value>,td:<Double_value>**

HTTP Method:DELETE

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue>

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error

get

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/lsnclient_nsacl6_binding/clientname_value<String>

Query-parameters:

filter

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/lsnclient_nsacl6_binding/clientname_value<String>?**filter=property-name1:property-value1,property-name2:property-value2**

Use this query-parameter to get the filtered set of lsnclient_nsacl6_binding resources configured on Citrix ADC. Filtering can be done on any of the properties of the resource.

pagination

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/lsnclient_nsacl6_binding/clientname_value<String>?**pagesize=#no;pageno=#no**

Use this query-parameter to get the lsnclient_nsacl6_binding resources in chunks.

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "lsnclient_nsacl6_binding": [ {

"clientname":<String_value>,

"acl6name":<String_value>,

"td":<Double_value>

}]}
```

get (all)

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/lsnclient_nsacl6_binding

Query-parameters:

bulkbindings

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/lsnclient_nsacl6_binding?**bulkbindings=yes**

NITRO allows you to fetch bindings in bulk.

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "lsnclient_nsacl6_binding": [ {

"clientname":<String_value>,

"acl6name":<String_value>,

"td":<Double_value>

}]}
```

count

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/lsnclient_nsacl6_binding/clientname_value<String>?**count=yes**

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{"lsnclient_nsacl6_binding": [ { "__count": "#no"} ] }
```

lsnclient_nsacl_binding

Binding object showing the nsacl that can be bound to lsnclient.

Properties

(click to see Operations)

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
clientname	<String>	Read-write	Name for the LSN client entity. Must begin with an ASCII alphanumeric or underscore (_) character, and must contain only ASCII alphanumeric, underscore, hash (#), period (.), space, colon (:), at (@), equals (=), and hyphen (-) characters. Cannot be changed after the LSN client is created. The following requirement applies only to the Citrix ADC CLI: If the name includes one or more spaces, enclose the name in double or single quotation marks (for example, "lsn client1" or 'lsn client1'). . Minimum length = 1 Maximum length = 127

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
aclname	<String>	Read-write	Name(s) of any configured extended ACL(s) whose action is ALLOW. The condition specified in the extended ACL rule identifies the traffic from an LSN subscriber for which the Citrix ADC is to perform large scale NAT. . Minimum length = 1
td	<Double>	Read-write	ID of the traffic domain on which this subscriber or the subscriber network (as specified by the network parameter) belongs. If you do not specify an ID, the subscriber or the subscriber network becomes part of the default traffic domain. Default value: 0 Minimum value = 0 Maximum value = 4094
__count	<Double>	Read-write	count parameter

Operations

(click to see Properties)

ADD:| DELETE:| GET| GET (ALL)| COUNT

Some options that you can use for each operations:

- **Getting warnings in response:**NITRO allows you to get warnings in an operation by specifying the "warning" query parameter as "yes". For example, to get warnings while connecting to the Citrix ADC appliance, the URL is as follows:

  http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/login?warning=yes

  If any, the warnings are displayed in the response payload with the HTTP code "209 X-NITRO-WARNING".

- **Authenticated access for individual NITRO operations:**NITRO allows you to logon to the Citrix ADC appliance to perform individual operations. You can use this option instead of creating a NITRO session (using the login object) and then using that session to perform all operations,

  To do this, you must specify the username and password in the request header of the NITRO request as follows:

  X-NITRO-USER:<username>

  X-NITRO-PASS:<password>

  Note: In such cases, make sure that the request header DOES not include the following:

  Cookie:NITRO_AUTH_TOKEN=<tokenvalue>

`*Note:`*

Mandatory parameters are marked in redand placeholder content is marked in <green>.

add

URL:http://<Citrix-ADC-IP-address(NSIP)/nitro/v1/config/lsnclient_nsacl_binding

HTTP Method:PUT

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Content-Type:application/json

Request Payload:

```
{

"lsnclient_nsacl_binding":{

<b>"clientname":<String_value>,

</b>"aclname":<String_value>,

"td":<Double_value>

}}
```

Response:

HTTP Status Code on Success: 201 Created HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error

delete

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/lsnclient_nsacl_binding/clientname_value<String>

Query-parameters:

args

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/lsnclient_nsacl_binding/clientname_value<String>?**args=aclname:<String_value>,td:<Double_value>**

HTTP Method:DELETE

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue>

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error

get

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/lsnclient_nsacl_binding/clientname_value<String>

Query-parameters:

filter

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/lsnclient_nsacl_binding/clientname_value<String>?**filter=property-name1:property-value1,property-name2:property-value2**

Use this query-parameter to get the filtered set of lsnclient_nsacl_binding resources configured on Citrix ADC. Filtering can be done on any of the properties of the resource.

pagination

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/lsnclient_nsacl_binding/clientname_value<String>?**pagesize=#no;pageno=#no**

Use this query-parameter to get the lsnclient_nsacl_binding resources in chunks.

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "lsnclient_nsacl_binding": [ {

"clientname":<String_value>,
```

```
"aclname":<String_value>,

"td":<Double_value>

}]}
```

get (all)

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/lsnclient_nsacl_binding

Query-parameters:

bulkbindings

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/lsnclient_nsacl_binding?**bulkbindings=yes**

NITRO allows you to fetch bindings in bulk.

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "lsnclient_nsacl_binding": [ {

"clientname":<String_value>,

"aclname":<String_value>,

"td":<Double_value>
```

```
}]}
```

count

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/lsnclient_nsacl_binding/clientname_value<String>?**count=yes**

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{"lsnclient_nsacl_binding": [ { "__count": "#no"} ] }
```

lsndeterministicnat

Configuration for deterministic NAT resource.

Properties

(click to see Operations)

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
clientname	<String>	Read-write	The name of the LSN Client.

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
network6	<String>	Read-write	IPv6 address of the LSN subscriber or B4 device.
subscrip	<String>	Read-write	The Client IP address.
td	<Double>	Read-write	The LSN client TD. Default value: 0 Minimum value = 0 Maximum value = 4094
natip	<String>	Read-write	The NAT IP address. Minimum length = 1
natprefix	<String>	Read-only	IPv6 address of the LSN subscriber network(B4-Device) on whose traffic the Citrix ADC perform Large Scale NAT.
subscrip2	<String>	Read-only	The Subscriber IP address.
natip2	<String>	Read-only	The NAT IP address.
firstport	<Integer>	Read-only	The Application Protocol Port of the Profile. Minimum value = 1 Maximum value = 65535
lastport	<Integer>	Read-only	The Application Protocol Ending Port of the Profile.

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
			Minimum value = 1 Maximum value = 65535
srctd	<Double>	Read-only	The LSN client TD. Minimum value = 0 Maximum value = 4094
nattype	<String>	Read-only	Type of subscriber info(ex: nat44 or ds-lite). Default value: NAT44 Possible values = NAT44, DS-Lite, NAT64
__count	<Double>	Read-only	count parameter

Operations

(click to see Properties)

GET (ALL)| COUNT

Some options that you can use for each operations:

- **Getting warnings in response:**NITRO allows you to get warnings in an operation by specifying the "warning" query parameter as "yes". For example, to get warnings while connecting to the Citrix ADC appliance, the URL is as follows:

  http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/login?warning=yes

  If any, the warnings are displayed in the response payload with the HTTP code "209 X-NITRO-WARNING".

- **Authenticated access for individual NITRO operations:**NITRO allows you to logon to the Citrix ADC appliance to perform individual operations. You can use this option instead of creating a NITRO session (using the login object) and then using that session to perform all operations,

  To do this, you must specify the username and password in the request header of the NITRO request as follows:

  X-NITRO-USER:<username>

  X-NITRO-PASS:<password>

  Note: In such cases, make sure that the request header DOES not include the following:

  Cookie:NITRO_AUTH_TOKEN=<tokenvalue>

`*Note:`*

Mandatory parameters are marked in redand placeholder content is marked in <green>.

get (all)

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/lsndeterministicnat

Query-parameters:

args

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/lsndeterministicnat?**args=clientname:<String_value>,network6:<String_value>,subscrip:<String_value>,td:<Double_value>,**

Use this query-parameter to get lsndeterministicnat resources based on additional properties.

attrs

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/lsndeterministicnat?**attrs=property-name1,property-name2**

Use this query parameter to specify the resource details that you want to retrieve.

filter

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/lsndeterministicnat?**filter=property-name1:property-val1,property-name2:property-val2**

Use this query-parameter to get the filtered set of lsndeterministicnat resources configured on Citrix ADC. Filtering can be done on any of the properties of the resource.

view

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/lsndeterministicnat?**view=summary**

Note: By default, the retrieved results are displayed in detail view (?view=detail).

pagination

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/lsndeterministicnat?**pagesize=#no;pageno=#no**

Use this query-parameter to get the lsndeterministicnat resources in chunks.

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "lsndeterministicnat": [ {

clientname:<String_value>,network6:<String_value>,subscrip:<String_value>,
td:<Double_value>,natip:<String_value>"natprefix":<String_value>,

"subscrip2":<String_value>,

"natip2":<String_value>,

"firstport":<Integer_value>,
```

```
"lastport":<Integer_value>,

"srctd":<Double_value>,

"nattype":<String_value>

}]}
```

count

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/lsndeterministicnat?**count=yes**

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "lsndeterministicnat": [ { "__count": "#no"} ] }
```

lsngroup

Configuration for LSN group resource.

Properties

(click to see Operations)

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
groupname	<String>	Read-write	Name for the LSN group. Must begin with an ASCII alphanumeric or underscore (_) character, and must contain only ASCII alphanumeric, underscore, hash (#), period (.), space, colon (:), at (@), equals (=), and hyphen (-) characters. Cannot be changed after the LSN group is created. The following requirement applies only to the Citrix ADC CLI: If the name includes one or more spaces, enclose the name in double or single quotation marks (for example, "lsn group1" or 'lsn group1'). Minimum length = 1 Maximum length = 127
clientname	<String>	Read-write	Name of the LSN client entity to be associated with the LSN group. You can associate only one LSN client entity with an LSN group.You

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
			cannot remove this association or replace with another LSN client entity once the LSN group is created.
nattype	<String>	Read-write	Type of NAT IP address and port allocation (from the bound LSN pools) for subscribers: Available options function as follows: * Deterministic - Allocate a NAT IP address and a block of ports to each subscriber (of the LSN client bound to the LSN group). The Citrix ADC sequentially allocates NAT resources to these subscribers. The Citrix ADC ADC assigns the first block of ports (block size determined by the port block size parameter of the LSN group) on the beginning NAT IP address to the beginning subscriber IP

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
			address. The next range of ports is assigned to the next subscriber, and so on, until the NAT address does not have enough ports for the next subscriber. In this case, the first port block on the next NAT address is used for the subscriber, and so on. Because each subscriber now receives a deterministic NAT IP address and a block of ports, a subscriber can be identified without any need for logging. For a connection, a subscriber can be identified based only on the NAT IP address and port, and the destination IP address and port. The maximum number of LSN subscribers

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
			allowed, globally, is 1 million. * Dynamic - Allocate a random NAT IP address and a port from the LSN NAT pool for a subscriber's connection. If port block allocation is enabled (in LSN pool) and a port block size is specified (in the LSN group), the Citrix ADC allocates a random NAT IP address and a block of ports for a subscriber when it initiates a connection for the first time. The ADC allocates this NAT IP address and a port (from the allocated block of ports) for different connections from this subscriber. If all the ports are allocated (for different subscriber's connections) from

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
			the subscriber's allocated port block, the ADC allocates a new random port block for the subscriber. Default value: DYNAMIC Possible values = DYNAMIC, DETERMINISTIC
allocpolicy	<String>	Read-write	NAT IP and PORT block allocation policy for Deterministic NAT. Supported Policies are, 1: PORTS: Port blocks from single NATIP will be allocated to LSN subscribers sequentially. After all blocks are exhausted, port blocks from next NATIP will be allocated and so on. 2: IPADDRS(Default): One port block from each NATIP will be allocated and once all the NATIPs are

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
			over second port block from each NATIP will be allocated and so on. To understand better if we assume port blocks of all NAT IPs as two dimensional array, PORTS policy follows "row major order" and IPADDRS policy follows "column major order" while allocating port blocks. Example: Client IPs: 2.2.2.1, 2.2.2.2 and 2.2.2.3 NAT IPs and PORT Blocks: 4.4.4.1:PB1, PB2, PB3,., PBn 4.4.4.2: PB1, PB2, PB3,., PBn PORTS Policy: 2.2.2.1 => 4.4.4.1:PB1

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
			2.2.2.2 => 4.4.4.1:PB2 2.2.2.3 => 4.4.4.1:PB3 IPADDRS Policy: 2.2.2.1 => 4.4.4.1:PB1 2.2.2.2 => 4.4.4.2:PB1 2.2.2.3 => 4.4.4.1:PB2. Default value: IPADDRS Possible values = PORTS, IPADDRS
portblocksize	<Double>	Read-write	Size of the NAT port block to be allocated for each subscriber. To set this parameter for Dynamic NAT, you must enable the port block allocation parameter in the bound LSN pool. For Deterministic NAT, the port block allocation parameter is always

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
			enabled, and you cannot disable it. In Dynamic NAT, the Citrix ADC allocates a random NAT port block, from the available NAT port pool of an NAT IP address, for each subscriber. For a subscriber, if all the ports are allocated from the subscriber's allocated port block, the ADC allocates a new random port block for the subscriber. The default port block size is 256 for Deterministic NAT, and 0 for Dynamic NAT. Default value: 0 Minimum value = 256 Maximum value = 65536
logging	<String>	Read-write	Log mapping entries and sessions created or deleted for this LSN group. The

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
			Citrix ADC logs LSN sessions for this LSN group only when both logging and session logging parameters are enabled. The ADC uses its existing syslog and audit log framework to log LSN information. You must enable global level LSN logging by enabling the LSN parameter in the related NSLOG action and SYLOG action entities. When the Logging parameter is enabled, the Citrix ADC generates log messages related to LSN mappings and LSN sessions of this LSN group. The ADC then sends these log messages to servers associated with the NSLOG action and SYSLOG actions entities.

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
			A log message for an LSN mapping entry consists of the following information: * NSIP address of the Citrix ADC * Time stamp * Entry type (MAPPING or SESSION) * Whether the LSN mapping entry is created or deleted * Subscriber's IP address, port, and traffic domain ID * NAT IP address and port * Protocol name * Destination IP address, port, and traffic domain ID might be present, depending on the following conditions: ` Destination IP address and port are not logged for Endpoint-

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
			Independent mapping ` Only Destination IP address (and not port) is logged for Address-Dependent mapping  ` Destination IP address and port are logged for Address-Port-Dependent mapping. Default value: DISABLED Possible values = ENABLED, DISABLED
sessionlogging	<String>	Read-write	Log sessions created or deleted for the LSN group. The Citrix ADC logs LSN sessions for this LSN group only when both logging and session logging parameters are enabled. A log message for an LSN session consists of the following information:

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
			* NSIP address of the Citrix ADC * Time stamp * Entry type (MAPPING or SESSION) * Whether the LSN session is created or removed * Subscriber's IP address, port, and traffic domain ID * NAT IP address and port * Protocol name * Destination IP address, port, and traffic domain ID. Default value: DISABLED Possible values = ENABLED, DISABLED
sessionsync	<String>	Read-write	In a high availability (HA) deployment, synchronize information of all LSN sessions related to this LSN group with the secondary node. After a failover,

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
			established TCP connections and UDP packet flows are kept active and resumed on the secondary node (new primary). For this setting to work, you must enable the global session synchronization parameter. Default value: ENABLED Possible values = ENABLED, DISABLED
snmptraplimit	<Double>	Read-write	Maximum number of SNMP Trap messages that can be generated for the LSN group in one minute. Default value: 100 Minimum value = 0 Maximum value = 10000
ftp	<String>	Read-write	Enable Application Layer Gateway (ALG) for the FTP protocol. For some application-layer protocols, the IP

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
			addresses and protocol port numbers are usually communicated in the packet's payload. When acting as an ALG, the Citrix ADC changes the packet's payload to ensure that the protocol continues to work over LSN. Note: The Citrix ADC also includes ALG for ICMP and TFTP protocols. ALG for the ICMP protocol is enabled by default, and there is no provision to disable it. ALG for the TFTP protocol is disabled by default. ALG is enabled automatically for an LSN group when you bind a UDP LSN application profile, with endpoint-independent-mapping, endpoint-independent filtering, and destination port as 69 (well-known port

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
			for TFTP), to the LSN group. Default value: ENABLED Possible values = ENABLED, DISABLED
pptp	<String>	Read-write	Enable the PPTP Application Layer Gateway. Default value: DISABLED Possible values = ENABLED, DISABLED
sipalg	<String>	Read-write	Enable the SIP ALG. Default value: DISABLED Possible values = ENABLED, DISABLED
rtspalg	<String>	Read-write	Enable the RTSP ALG. Default value: DISABLED Possible values = ENABLED, DISABLED
ip6profile	<String>	Read-write	Name of the LSN ip6 profile to associate with the specified

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
			LSN group. An ip6 profile can be associated with a group only during group creation. By default, no LSN ip6 profile is associated with an LSN group during its creation. Only one ip6profile can be associated with a group. Minimum length = 1 Maximum length = 127
ftpcm	<String>	Read-write	Enable the FTP connection mirroring for specified LSN group. Connection mirroring (CM or connection failover) refers to keeping active an established TCP or UDP connection when a failover occurs. Default value: DISABLED Possible values = ENABLED, DISABLED
groupid	<Double>	Read-only	. Minimum value = 1

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
__count	<Double>	Read-only	count parameter

Operations

(click to see Properties)

ADD| DELETE| UPDATE| UNSET| GET (ALL)| GET| COUNT

Some options that you can use for each operations:

- **Getting warnings in response:**NITRO allows you to get warnings in an operation by specifying the "warning" query parameter as "yes". For example, to get warnings while connecting to the Citrix ADC appliance, the URL is as follows:

  http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/login?warning=yes

  If any, the warnings are displayed in the response payload with the HTTP code "209 X-NITRO-WARNING".

- **Authenticated access for individual NITRO operations:**NITRO allows you to logon to the Citrix ADC appliance to perform individual operations. You can use this option instead of creating a NITRO session (using the login object) and then using that session to perform all operations,

  To do this, you must specify the username and password in the request header of the NITRO request as follows:

  X-NITRO-USER:<username>

  X-NITRO-PASS:<password>

  Note: In such cases, make sure that the request header DOES not include the following:

  Cookie:NITRO_AUTH_TOKEN=<tokenvalue>

`*Note:`*

Mandatory parameters are marked in redand placeholder content is marked in <green>.

add

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/lsngroup

HTTP Method:POST

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Content-Type:application/json

Request Payload:

```
{"lsngroup":{

<b>"groupname":<String_value>,

</b><b>"clientname":<String_value>,

</b>"nattype":<String_value>,

"allocpolicy":<String_value>,

"portblocksize":<Double_value>,

"logging":<String_value>,

"sessionlogging":<String_value>,

"sessionsync":<String_value>,

"snmptraplimit":<Double_value>,

"ftp":<String_value>,

"pptp":<String_value>,

"sipalg":<String_value>,

"rtspalg":<String_value>,

"ip6profile":<String_value>,

"ftpcm":<String_value>

}}
```

Response:

HTTP Status Code on Success: 201 Created HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error

delete

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/lsngroup/groupname_value<String>

HTTP Method:DELETE

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue>

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error

update

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/lsngroup

HTTP Method:PUT

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Content-Type:application/json

Request Payload:

```
{"lsngroup":{

<b>"groupname":<String_value>,

</b>"portblocksize":<Double_value>,

"logging":<String_value>,

"sessionlogging":<String_value>,
```

```
"sessionsync":<String_value>,

"snmptraplimit":<Double_value>,

"ftp":<String_value>,

"pptp":<String_value>,

"sipalg":<String_value>,

"rtspalg":<String_value>,

"ftpcm":<String_value>

}}
```

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error

unset

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/lsngroup?**action=unset**

HTTP Method:POST

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Content-Type:application/json

Request Payload:

```
{"lsngroup":{

<b>"groupname":<String_value>,

</b>"portblocksize":true,
```

```
"logging":true,

"sessionlogging":true,

"sessionsync":true,

"snmptraplimit":true,

"ftp":true,

"pptp":true,

"sipalg":true,

"rtspalg":true,

"ftpcm":true

}}
```

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error

get (all)

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/lsngroup

Query-parameters:

attrs

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/lsngroup?**attrs=property-name1,property-name2**

Use this query parameter to specify the resource details that you want to retrieve.

filter

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/lsngroup?**filter=property-name1:property-val1,property-name2:property-val2**

Use this query-parameter to get the filtered set of lsngroup resources configured on Citrix ADC. Filtering can be done on any of the properties of the resource.

view

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/lsngroup?**view=summary**

Note: By default, the retrieved results are displayed in detail view (?view=detail).

pagination

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/lsngroup?**pagesize=#no;pageno=#no**

Use this query-parameter to get the lsngroup resources in chunks.

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "lsngroup": [ {

"groupname":<String_value>,

"clientname":<String_value>,

"nattype":<String_value>,

"allocpolicy":<String_value>,
```

```
"portblocksize":<Double_value>,

"logging":<String_value>,

"sessionlogging":<String_value>,

"sessionsync":<String_value>,

"snmptraplimit":<Double_value>,

"ftp":<String_value>,

"pptp":<String_value>,

"groupid":<Double_value>,

"sipalg":<String_value>,

"rtspalg":<String_value>,

"ip6profile":<String_value>,

"ftpcm":<String_value>

}]}
```

get

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/lsngroup/groupname_value<String>

Query-parameters:

attrs

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/lsngroup/groupname_value<String>?**attrs=property-name1,property-name2**

Use this query parameter to specify the resource details that you want to retrieve.

view

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/lsngroup/groupname_value<String>?**view=summary**

Note: By default, the retrieved results are displayed in detail view (?view=detail).

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "lsngroup": [ {

"groupname":<String_value>,

"clientname":<String_value>,

"nattype":<String_value>,

"allocpolicy":<String_value>,

"portblocksize":<Double_value>,

"logging":<String_value>,

"sessionlogging":<String_value>,

"sessionsync":<String_value>,

"snmptraplimit":<Double_value>,

"ftp":<String_value>,
```

```
"pptp":<String_value>,

"groupid":<Double_value>,

"sipalg":<String_value>,

"rtspalg":<String_value>,

"ip6profile":<String_value>,

"ftpcm":<String_value>

}]}
```

count

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/lsngroup?**count=yes**

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "lsngroup": [ { "__count": "#no"} ] }
```

lsngroup_binding

Binding object which returns the resources bound to lsngroup.

Properties

(click to see Operations)

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
groupname	<String>	Read-write	Name for the LSN group. Must begin with an ASCII alphanumeric or underscore (_) character, and must contain only ASCII alphanumeric, underscore, hash (#), period (.), space, colon (:), at (@), equals (=), and hyphen (-) characters. Cannot be changed after the LSN group is created. The following requirement applies only to the Citrix ADC CLI: If the name includes one or more spaces, enclose the name in double or single quotation marks (for example, "lsn group1" or 'lsn group1'). Minimum length = 1 Maximum length = 127
lsngroup_lsnsipalgprofile_binding	<lsngroup_lsnsipalgprofile_binding[]>	Read-only	lsnsipalgprofile that can be bound to lsngroup.

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
lsngroup_ipsecalgprofile_binding	<lsngroup_ipsecalgprofile_binding[]>	Read-only	ipsecalgprofile that can be bound to lsngroup.
lsngroup_pcpserver_binding	<lsngroup_pcpserver_binding[]>	Read-only	pcpserver that can be bound to lsngroup.
lsngroup_lsnhttphdrlogprofile_binding	<lsngroup_lsnhttphdrlogprofile_binding[]>	Read-only	lsnhttphdrlogprofile that can be bound to lsngroup.
lsngroup_lsnpool_binding	<lsngroup_lsnpool_binding[]>	Read-only	lsnpool that can be bound to lsngroup.
lsngroup_lsntransportprofile_binding	<lsngroup_lsntransportprofile_binding[]>	Read-only	lsntransportprofile that can be bound to lsngroup.
lsngroup_lsnlogprofile_binding	<lsngroup_lsnlogprofile_binding[]>	Read-only	lsnlogprofile that can be bound to lsngroup.
lsngroup_lsnrtspalgprofile_binding	<lsngroup_lsnrtspalgprofile_binding[]>	Read-only	lsnrtspalgprofile that can be bound to lsngroup.
lsngroup_lsnappsprofile_binding	<lsngroup_lsnappsprofile_binding[]>	Read-only	lsnappsprofile that can be bound to lsngroup.

Operations

(click to see Properties)

GET| GET (ALL)

Some options that you can use for each operations:

- **Getting warnings in response:**NITRO allows you to get warnings in an operation by specifying the "warning" query parameter as "yes". For example, to get warnings while connecting to the Citrix ADC appliance, the URL is as follows:

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/login?warning=yes

If any, the warnings are displayed in the response payload with the HTTP code "209 X-NITRO-WARNING".

- **Authenticated access for individual NITRO operations:**NITRO allows you to logon to the Citrix ADC appliance to perform individual operations. You can use this option instead of creating a NITRO session (using the login object) and then using that session to perform all operations,

  To do this, you must specify the username and password in the request header of the NITRO request as follows:

  X-NITRO-USER:<username>

  X-NITRO-PASS:<password>

  Note: In such cases, make sure that the request header DOES not include the following:

  Cookie:NITRO_AUTH_TOKEN=<tokenvalue>

`*Note:`*

Mandatory parameters are marked in redand placeholder content is marked in <green>.

get

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/lsngroup_binding/groupname_value<String>

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "lsngroup_binding": [ {

"groupname":<String_value>,

"lsngroup_lsnhttphdrlogprofile_binding":<lsngroup_lsnhttphdrlogprofile_binding[]_value>,

"lsngroup_lsnappsprofile_binding":<lsngroup_lsnappsprofile_binding[]_value>,

"lsngroup_lsnrtspalgprofile_binding":<lsngroup_lsnrtspalgprofile_binding[]_value>,

"lsngroup_lsnsipalgprofile_binding":<lsngroup_lsnsipalgprofile_binding[]_value>,

"lsngroup_lsnpool_binding":<lsngroup_lsnpool_binding[]_value>,

"lsngroup_lsntransportprofile_binding":<lsngroup_lsntransportprofile_binding[]_value>,

"lsngroup_pcpserver_binding":<lsngroup_pcpserver_binding[]_value>,

"lsngroup_ipsecalgprofile_binding":<lsngroup_ipsecalgprofile_binding[]_value>,

"lsngroup_lsnlogprofile_binding":<lsngroup_lsnlogprofile_binding[]_value>

}]}
```

get (all)

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/lsngroup_binding

Query-parameters:

bulkbindings

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/lsngroup_binding?**bulkbindings=yes**

NITRO allows you to fetch bindings in bulk.

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "lsngroup_binding": [ {

"groupname":<String_value>,

"lsngroup_lsnhttphdrlogprofile_binding":<lsngroup_lsnhttphdrlogprofile_bin
ding[]_value>,

"lsngroup_lsnappsprofile_binding":<lsngroup_lsnappsprofile_binding[]_value
>,

"lsngroup_lsnrtspalgprofile_binding":<lsngroup_lsnrtspalgprofile_binding[]
_value>,

"lsngroup_lsnsipalgprofile_binding":<lsngroup_lsnsipalgprofile_binding[]_v
alue>,

"lsngroup_lsnpool_binding":<lsngroup_lsnpool_binding[]_value>,

"lsngroup_lsntransportprofile_binding":<lsngroup_lsntransportprofile_bindi
ng[]_value>,

"lsngroup_pcpserver_binding":<lsngroup_pcpserver_binding[]_value>,

"lsngroup_ipsecalgprofile_binding":<lsngroup_ipsecalgprofile_binding[]_val
ue>,
```

```
"lsngroup_lsnlogprofile_binding":<lsngroup_lsnlogprofile_binding[]_value>

}]}
```

lsngroup_ipsecalgprofile_binding

Binding object showing the ipsecalgprofile that can be bound to lsngroup.

Properties

(click to see Operations)

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
ipsecalgprofile	<String>	Read-write	Name of the IPSec ALG profile to bind to the specified LSN group.
groupname	<String>	Read-write	Name for the LSN group. Must begin with an ASCII alphanumeric or underscore (_) character, and must contain only ASCII alphanumeric, underscore, hash (#), period (.), space, colon (:), at (@), equals (=), and hyphen (-) characters. Cannot be changed after the LSN group is created. The following requirement applies only to the Citrix ADC CLI: If the name includes one

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
			or more spaces, enclose the name in double or single quotation marks (for example, "lsn group1" or 'lsn group1'). Minimum length = 1 Maximum length = 127
__count	<Double>	Read-write	count parameter

Operations

(click to see Properties)

ADD:| DELETE:| GET| GET (ALL)| COUNT

Some options that you can use for each operations:

- **Getting warnings in response:**NITRO allows you to get warnings in an operation by specifying the "warning" query parameter as "yes". For example, to get warnings while connecting to the Citrix ADC appliance, the URL is as follows:

  http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/login?warning=yes

  If any, the warnings are displayed in the response payload with the HTTP code "209 X-NITRO-WARNING".

- **Authenticated access for individual NITRO operations:**NITRO allows you to logon to the Citrix ADC appliance to perform individual operations. You can use this option instead of creating a NITRO session (using the login object) and then using that session to perform all operations,

  To do this, you must specify the username and password in the request header of the NITRO request as follows:

  X-NITRO-USER:<username>

  X-NITRO-PASS:<password>

Note: In such cases, make sure that the request header DOES not include the following:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue>

`*Note:` *

Mandatory parameters are marked in redand placeholder content is marked in <green>.

add

URL:http://<Citrix-ADC-IP-address(NSIP)/nitro/v1/config/lsngroup_ipsecalgprofile_binding

HTTP Method:PUT

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Content-Type:application/json

Request Payload:

```
{

"lsngroup_ipsecalgprofile_binding":{

<b>"groupname":<String_value>,

</b>"ipsecalgprofile":<String_value>

}}
```

Response:

HTTP Status Code on Success: 201 Created HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error

delete

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/lsngroup_ipsecalgprofile_binding/groupname_value<String>

Query-parameters:

args

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/lsngroup_ipsecalgprofile_binding/groupname_value<String>?**args=ipsecalgprofile:<String_value>**

HTTP Method:DELETE

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue>

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error

get

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/lsngroup_ipsecalgprofile_binding/groupname_value<String>

Query-parameters:

filter

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/lsngroup_ipsecalgprofile_binding/groupname_value<String>?**filter=property-name1:property-value1,property-name2:property-value2**

Use this query-parameter to get the filtered set of lsngroup_ipsecalgprofile_binding resources configured on Citrix ADC. Filtering can be done on any of the properties of the resource.

pagination

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/lsngroup_ipsecalgprofile_binding/groupname_value<String>?**pagesize=#no;pageno=#no**

Use this query-parameter to get the lsngroup_ipsecalgprofile_binding resources in chunks.

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "lsngroup_ipsecalgprofile_binding": [ {

"ipsecalgprofile":<String_value>,

"groupname":<String_value>

}]}
```

get (all)

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/lsngroup_ipsecalgprofile_binding

Query-parameters:

bulkbindings

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/lsngroup_ipsecalgprofile_binding?**bulkbindings=yes**

NITRO allows you to fetch bindings in bulk.

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "lsngroup_ipsecalgprofile_binding": [ {

"ipsecalgprofile":<String_value>,

"groupname":<String_value>

}]}
```

count

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/lsngroup_ipsecalgprofile_binding/groupname_value<String>?**count=yes**

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{"lsngroup_ipsecalgprofile_binding": [ { "__count": "#no"} ] }
```

lsngroup_lsnappsprofile_binding

Binding object showing the lsnappsprofile that can be bound to lsngroup.

Properties

(click to see Operations)

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
appsprofilename	<String>	Read-write	Name of the LSN application profile to bind to the specified LSN group. For each set of destination ports, bind a profile for each protocol for which you want to specify settings. By default, one LSN application profile with default settings for TCP, UDP, and ICMP protocols for all destination ports is bound to an LSN group during its creation. This profile is called a default application profile. When you bind an LSN application profile, with a specified set of destination ports, to an LSN group, the bound profile overrides the default LSN application profile for that

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
			protocol at that set of destination ports.
groupname	<String>	Read-write	Name for the LSN group. Must begin with an ASCII alphanumeric or underscore (_) character, and must contain only ASCII alphanumeric, underscore, hash (#), period (.), space, colon (:), at (@), equals (=), and hyphen (-) characters. Cannot be changed after the LSN group is created. The following requirement applies only to the Citrix ADC CLI: If the name includes one or more spaces, enclose the name in double or single quotation marks (for example, "lsn group1" or 'lsn group1'). Minimum length = 1 Maximum length = 127
__count	<Double>	Read-write	count parameter

Operations

(click to see Properties)

ADD:| DELETE:| GET| GET (ALL)| COUNT

Some options that you can use for each operations:

- **Getting warnings in response:**NITRO allows you to get warnings in an operation by specifying the "warning" query parameter as "yes". For example, to get warnings while connecting to the Citrix ADC appliance, the URL is as follows:

  http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/login?warning=yes

  If any, the warnings are displayed in the response payload with the HTTP code "209 X-NITRO-WARNING".

- **Authenticated access for individual NITRO operations:**NITRO allows you to logon to the Citrix ADC appliance to perform individual operations. You can use this option instead of creating a NITRO session (using the login object) and then using that session to perform all operations,

  To do this, you must specify the username and password in the request header of the NITRO request as follows:

  X-NITRO-USER:<username>

  X-NITRO-PASS:<password>

  Note: In such cases, make sure that the request header DOES not include the following:

  Cookie:NITRO_AUTH_TOKEN=<tokenvalue>

`*Note:`*

Mandatory parameters are marked in redand placeholder content is marked in <green>.

add

URL:http://<Citrix-ADC-IP-address(NSIP)/nitro/v1/config/lsngroup_lsnappsprofile_binding

HTTP Method:PUT

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Content-Type:application/json

Request Payload:

```
{

"lsngroup_lsnappsprofile_binding":{

<b>"groupname":<String_value>,

</b>"appsprofilename":<String_value>

}}
```

Response:

HTTP Status Code on Success: 201 Created HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error

delete

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/lsngroup_lsnappsprofile_binding/groupname_value<String>

Query-parameters:

args

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/lsngroup_lsnappsprofile_binding/groupname_value<String>?**args=appsprofilename:<String_value>**

HTTP Method:DELETE

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue>

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error

get

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/lsngroup_lsnappsprofile_binding/groupname_value<String>

Query-parameters:

filter

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/lsngroup_lsnappsprofile_binding/groupname_value<String>?**filter=property-name1:property-value1,property-name2:property-value2**

Use this query-parameter to get the filtered set of lsngroup_lsnappsprofile_binding resources configured on Citrix ADC. Filtering can be done on any of the properties of the resource.

pagination

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/lsngroup_lsnappsprofile_binding/groupname_value<String>?**pagesize=#no;pageno=#no**

Use this query-parameter to get the lsngroup_lsnappsprofile_binding resources in chunks.

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "lsngroup_lsnappsprofile_binding": [ {

"appsprofilename":<String_value>,
```

```
"groupname":<String_value>

}]}
```

get (all)

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/lsngroup_lsnappsprofile_binding

Query-parameters:

bulkbindings

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/lsngroup_lsnappsprofile_binding?**bulkbindings=yes**

NITRO allows you to fetch bindings in bulk.

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "lsngroup_lsnappsprofile_binding": [ {

"appsprofilename":<String_value>,

"groupname":<String_value>

}]}
```

count

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/lsngroup_lsnappsprofile_binding/groupname_value<String>?**count=yes**

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{"lsngroup_lsnappsprofile_binding": [ { "__count": "#no"} ] }
```

lsngroup_lsnhttphdrlogprofile_binding

Binding object showing the lsnhttphdrlogprofile that can be bound to lsngroup.

Properties

(click to see Operations)

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
httphdrlogprofilename	<String>	Read-write	The name of the LSN HTTP header logging Profile.
groupname	<String>	Read-write	Name for the LSN group. Must begin

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
			with an ASCII alphanumeric or underscore (_) character, and must contain only ASCII alphanumeric, underscore, hash (#), period (.), space, colon (:), at (@), equals (=), and hyphen (-) characters. Cannot be changed after the LSN group is created. The following requirement applies only to the Citrix ADC CLI: If the name includes one or more spaces, enclose the name in double or single quotation marks (for example, "lsn group1" or 'lsn group1'). Minimum length = 1 Maximum length = 127
__count	<Double>	Read-write	count parameter

Operations

(click to see Properties)

ADD:| DELETE:| GET| GET (ALL)| COUNT

Some options that you can use for each operations:

- **Getting warnings in response:**NITRO allows you to get warnings in an operation by specifying the "warning" query parameter as "yes". For example, to get warnings while connecting to the Citrix ADC appliance, the URL is as follows:

  http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/login?warning=yes

  If any, the warnings are displayed in the response payload with the HTTP code "209 X-NITRO-WARNING".

- **Authenticated access for individual NITRO operations:**NITRO allows you to logon to the Citrix ADC appliance to perform individual operations. You can use this option instead of creating a NITRO session (using the login object) and then using that session to perform all operations,

  To do this, you must specify the username and password in the request header of the NITRO request as follows:

  X-NITRO-USER:<username>

  X-NITRO-PASS:<password>

  Note: In such cases, make sure that the request header DOES not include the following:

  Cookie:NITRO_AUTH_TOKEN=<tokenvalue>

`*Note:`*

Mandatory parameters are marked in redand placeholder content is marked in <green>.

add

URL:http://<Citrix-ADC-IP-address(NSIP)/nitro/v1/config/lsngroup_lsnhttphdrlogprofile_binding

HTTP Method:PUT

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Content-Type:application/json

Request Payload:

```
{
```

```
"lsngroup_lsnhttphdrlogprofile_binding":{

<b>"groupname":<String_value>,

</b>"httphdrlogprofilename":<String_value>

}}
```

Response:

HTTP Status Code on Success: 201 Created HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error

delete

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/lsngroup_lsnhttphdrlogprofile_binding/groupname_value<String>

Query-parameters:

args

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/lsngroup_lsnhttphdrlogprofile_binding/groupname_value<String>?**args=httphdrlogprofilename:<String_value>**

HTTP Method:DELETE

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue>

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error

get

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/lsngroup_lsnhttphdrlogprofile_binding/groupname_value<String>

Query-parameters:

filter

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/lsngroup_lsnhttphdrlogprofile_binding/groupname_value<String>?**filter=property-name1:property-value1,property-name2:property-value2**

Use this query-parameter to get the filtered set of lsngroup_lsnhttphdrlogprofile_binding resources configured on Citrix ADC. Filtering can be done on any of the properties of the resource.

pagination

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/lsngroup_lsnhttphdrlogprofile_binding/groupname_value<String>?**pagesize=#no;pageno=#no**

Use this query-parameter to get the lsngroup_lsnhttphdrlogprofile_binding resources in chunks.

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "lsngroup_lsnhttphdrlogprofile_binding": [ {

"httphdrlogprofilename":<String_value>,

"groupname":<String_value>

}]}
```

get (all)

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/lsngroup_lsnhttphdrlogprofile_binding

Query-parameters:

bulkbindings

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/lsngroup_lsnhttphdrlogprofile_binding?**bulkbindings=yes**

NITRO allows you to fetch bindings in bulk.

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "lsngroup_lsnhttphdrlogprofile_binding": [ {

"httphdrlogprofilename":<String_value>,

"groupname":<String_value>

}]}
```

count

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/lsngroup_lsnhttphdrlogprofile_binding/groupname_value<String>?**count=yes**

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{"lsngroup_lsnhttphdrlogprofile_binding": [ { "__count": "#no"} ] }
```

lsngroup_lsnlogprofile_binding

Binding object showing the lsnlogprofile that can be bound to lsngroup.

Properties

(click to see Operations)

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
logprofilename	<String>	Read-write	The name of the LSN logging Profile.
groupname	<String>	Read-write	Name for the LSN group. Must begin with an ASCII alphanumeric or underscore (_)

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
			character, and must contain only ASCII alphanumeric, underscore, hash (#), period (.), space, colon (:), at (@), equals (=), and hyphen (-) characters. Cannot be changed after the LSN group is created. The following requirement applies only to the Citrix ADC CLI: If the name includes one or more spaces, enclose the name in double or single quotation marks (for example, "lsn group1" or 'lsn group1'). Minimum length = 1 Maximum length = 127
__count	<Double>	Read-write	count parameter

Operations

(click to see Properties)

ADD:| DELETE:| GET| GET (ALL)| COUNT

Some options that you can use for each operations:

- **Getting warnings in response:**NITRO allows you to get warnings in an operation by specifying the "warning" query parameter as "yes". For example, to get warnings while connecting to the Citrix ADC appliance, the URL is as follows:

  http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/login?warning=yes

  If any, the warnings are displayed in the response payload with the HTTP code "209 X-NITRO-WARNING".

- **Authenticated access for individual NITRO operations:**NITRO allows you to logon to the Citrix ADC appliance to perform individual operations. You can use this option instead of creating a NITRO session (using the login object) and then using that session to perform all operations,

  To do this, you must specify the username and password in the request header of the NITRO request as follows:

  X-NITRO-USER:<username>

  X-NITRO-PASS:<password>

  Note: In such cases, make sure that the request header DOES not include the following:

  Cookie:NITRO_AUTH_TOKEN=<tokenvalue>

`*Note:`*

Mandatory parameters are marked in redand placeholder content is marked in <green>.

add

URL:http://<Citrix-ADC-IP-address(NSIP)/nitro/v1/config/lsngroup_lsnlogprofile_binding

HTTP Method:PUT

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Content-Type:application/json

Request Payload:

```
{
```

```
"lsngroup_lsnlogprofile_binding":{

<b>"groupname":<String_value>,

</b>"logprofilename":<String_value>

}}
```

Response:

HTTP Status Code on Success: 201 Created HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error

delete

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/lsngroup_lsnlogprofile_binding/groupname_value<String>

Query-parameters:

args

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/lsngroup_lsnlogprofile_binding/groupname_value<String>?**args=logprofilename:<String_value>**

HTTP Method:DELETE

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue>

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error

get

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/lsngroup_lsnlogprofile_binding/groupname_value<String>

Query-parameters:

filter

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/lsngroup_lsnlogprofile_binding/groupname_value<String>?**filter=property-name1:property-value1,property-name2:property-value2**

Use this query-parameter to get the filtered set of lsngroup_lsnlogprofile_binding resources configured on Citrix ADC. Filtering can be done on any of the properties of the resource.

pagination

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/lsngroup_lsnlogprofile_binding/groupname_value<String>?**pagesize=#no;pageno=#no**

Use this query-parameter to get the lsngroup_lsnlogprofile_binding resources in chunks.

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "lsngroup_lsnlogprofile_binding": [ {

"logprofilename":<String_value>,

"groupname":<String_value>

}]}
```

get (all)

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/lsngroup_lsnlogprofile_binding

Query-parameters:

bulkbindings

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/lsngroup_lsnlogprofile_binding?**bulkbindings=yes**

NITRO allows you to fetch bindings in bulk.

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "lsngroup_lsnlogprofile_binding": [ {

"logprofilename":<String_value>,

"groupname":<String_value>

}]}
```

count

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/lsngroup_lsnlogprofile_binding/groupname_value<String>?**count=yes**

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{"lsngroup_lsnlogprofile_binding": [ { "__count": "#no"} ] }
```

lsngroup_lsnpool_binding

Binding object showing the lsnpool that can be bound to lsngroup.

Properties

(click to see Operations)

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
poolname	<String>	Read-write	Name of the LSN pool to bind to the specified LSN group. Only LSN Pools and LSN groups with the same NAT type settings can be bound together.

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
			Multiples LSN pools can be bound to an LSN group. For Deterministic NAT, pools bound to an LSN group cannot be bound to other LSN groups. For Dynamic NAT, pools bound to an LSN group can be bound to multiple LSN groups.
groupname	<String>	Read-write	Name for the LSN group. Must begin with an ASCII alphanumeric or underscore (_) character, and must contain only ASCII alphanumeric, underscore, hash (#), period (.), space, colon (:), at (@), equals (=), and hyphen (-) characters. Cannot be changed after the LSN group is created. The following requirement applies only to the Citrix ADC CLI: If the name includes one or more spaces, enclose the name in double or single quotation marks (for example, "lsn

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
			group1" or 'lsn group1'). Minimum length = 1 Maximum length = 127
__count	<Double>	Read-write	count parameter

Operations

(click to see Properties)

ADD:| DELETE:| GET| GET (ALL)| COUNT

Some options that you can use for each operations:

- **Getting warnings in response:**NITRO allows you to get warnings in an operation by specifying the "warning" query parameter as "yes". For example, to get warnings while connecting to the Citrix ADC appliance, the URL is as follows:

  http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/login?warning=yes

  If any, the warnings are displayed in the response payload with the HTTP code "209 X-NITRO-WARNING".

- **Authenticated access for individual NITRO operations:**NITRO allows you to logon to the Citrix ADC appliance to perform individual operations. You can use this option instead of creating a NITRO session (using the login object) and then using that session to perform all operations,

  To do this, you must specify the username and password in the request header of the NITRO request as follows:

  X-NITRO-USER:<username>

  X-NITRO-PASS:<password>

  Note: In such cases, make sure that the request header DOES not include the following:

  Cookie:NITRO_AUTH_TOKEN=<tokenvalue>

`*Note:`*

Mandatory parameters are marked in redand placeholder content is marked in <green>.

add

URL:http://<Citrix-ADC-IP-address(NSIP)/nitro/v1/config/lsngroup_lsnpool_binding

HTTP Method:PUT

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Content-Type:application/json

Request Payload:

```
{

"lsngroup_lsnpool_binding":{

<b>"groupname":<String_value>,

</b>"poolname":<String_value>

}}
```

Response:

HTTP Status Code on Success: 201 Created HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error

delete

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/lsngroup_lsnpool_binding/groupname_value<String>

Query-parameters:

args

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/lsngroup_lsnpool_binding/groupname_value<String>?**args=poolname:<String_value>**

HTTP Method:DELETE

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue>

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error

get

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/lsngroup_lsnpool_binding/groupname_value<String>

Query-parameters:

filter

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/lsngroup_lsnpool_binding/groupname_value<String>?**filter=property-name1:property-value1,property-name2:property-value2**

Use this query-parameter to get the filtered set of lsngroup_lsnpool_binding resources configured on Citrix ADC. Filtering can be done on any of the properties of the resource.

pagination

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/lsngroup_lsnpool_binding/groupname_value<String>?**pagesize=#no;pageno=#no**

Use this query-parameter to get the lsngroup_lsnpool_binding resources in chunks.

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "lsngroup_lsnpool_binding": [ {

"poolname":<String_value>,

"groupname":<String_value>

}]}
```

get (all)

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/lsngroup_lsnpool_binding

Query-parameters:

bulkbindings

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/lsngroup_lsnpool_binding?**bulkbindings=yes**

NITRO allows you to fetch bindings in bulk.

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "lsngroup_lsnpool_binding": [ {

"poolname":<String_value>,

"groupname":<String_value>

}]}
```

count

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/lsngroup_lsnpool_binding/groupname_value<String>?**count=yes**

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{"lsngroup_lsnpool_binding": [ { "__count": "#no"} ] }
```

lsngroup_lsnrtspalgprofile_binding

Binding object showing the lsnrtspalgprofile that can be bound to lsngroup.

Properties

(click to see Operations)

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
rtspalgprofilename	<String>	Read-write	The name of the LSN RTSP ALG Profile.
groupname	<String>	Read-write	Name for the LSN group. Must begin with an ASCII alphanumeric or underscore (_) character, and must contain only ASCII alphanumeric, underscore, hash (#), period (.), space, colon (:), at (@), equals (=), and hyphen (-) characters. Cannot be changed after the LSN group is created. The following requirement applies only to the Citrix ADC CLI: If the name includes one or more spaces, enclose the name in double or single quotation marks (for example, "lsn group1" or 'lsn group1'). Minimum length = 1 Maximum length = 127

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
__count	<Double>	Read-write	count parameter

Operations

(click to see Properties)

ADD:| DELETE:| GET| GET (ALL)| COUNT

Some options that you can use for each operations:

- **Getting warnings in response:**NITRO allows you to get warnings in an operation by specifying the "warning" query parameter as "yes". For example, to get warnings while connecting to the Citrix ADC appliance, the URL is as follows:

  http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/login?warning=yes

  If any, the warnings are displayed in the response payload with the HTTP code "209 X-NITRO-WARNING".
- **Authenticated access for individual NITRO operations:**NITRO allows you to logon to the Citrix ADC appliance to perform individual operations. You can use this option instead of creating a NITRO session (using the login object) and then using that session to perform all operations,

  To do this, you must specify the username and password in the request header of the NITRO request as follows:

  X-NITRO-USER:<username>

  X-NITRO-PASS:<password>

  Note: In such cases, make sure that the request header DOES not include the following:

  Cookie:NITRO_AUTH_TOKEN=<tokenvalue>

`*Note:`*

Mandatory parameters are marked in redand placeholder content is marked in <green>.

add

URL:http://<Citrix-ADC-IP-address(NSIP)/nitro/v1/config/lsngroup_lsnrtspalgprofile_binding

HTTP Method:PUT

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Content-Type:application/json

Request Payload:

```
{

"lsngroup_lsnrtspalgprofile_binding":{

<b>"groupname":<String_value>,

</b>"rtspalgprofilename":<String_value>

}}
```

Response:

HTTP Status Code on Success: 201 Created HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error

delete

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/lsngroup_lsnrtspalgprofile_binding/groupname_value<String>

Query-parameters:

args

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/lsngroup_lsnrtspalgprofile_binding/groupname_value<String>?**args=rtspalgprofilename:<String_value>**

HTTP Method:DELETE

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue>

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error

get

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/lsngroup_lsnrtspalgprofile_binding/groupname_value<String>

Query-parameters:

filter

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/lsngroup_lsnrtspalgprofile_binding/groupname_value<String>?**filter=property-name1:property-value1,property-name2:property-value2**

Use this query-parameter to get the filtered set of lsngroup_lsnrtspalgprofile_binding resources configured on Citrix ADC. Filtering can be done on any of the properties of the resource.

pagination

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/lsngroup_lsnrtspalgprofile_binding/groupname_value<String>?**pagesize=#no;pageno=#no**

Use this query-parameter to get the lsngroup_lsnrtspalgprofile_binding resources in chunks.

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "lsngroup_lsnrtspalgprofile_binding": [ {

"rtspalgprofilename":<String_value>,

"groupname":<String_value>

}]}
```

get (all)

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/lsngroup_lsnrtspalgprofile_binding

Query-parameters:

bulkbindings

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/lsngroup_lsnrtspalgprofile_binding?**bulkbindings=yes**

NITRO allows you to fetch bindings in bulk.

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "lsngroup_lsnrtspalgprofile_binding": [ {
```

```
"rtspalgprofilename":<String_value>,

"groupname":<String_value>

}]}
```

count

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/lsngroup_lsnrtspalgprofile_binding/groupname_value<String>?**count=yes**

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{"lsngroup_lsnrtspalgprofile_binding": [ { "__count": "#no"} ] }
```

lsngroup_lsnsipalgprofile_binding

Binding object showing the lsnsipalgprofile that can be bound to lsngroup.

Properties

(click to see Operations)

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
sipalgprofilename	<String>	Read-write	The name of the LSN SIP ALG Profile.
groupname	<String>	Read-write	Name for the LSN group. Must begin with an ASCII alphanumeric or underscore (_) character, and must contain only ASCII alphanumeric, underscore, hash (#), period (.), space, colon (:), at (@), equals (=), and hyphen (-) characters. Cannot be changed after the LSN group is created. The following requirement applies only to the Citrix ADC CLI: If the name includes one or more spaces, enclose the name in double or single quotation marks (for example, "lsn group1" or 'lsn group1'). Minimum length = 1 Maximum length = 127
__count	<Double>	Read-write	count parameter

Operations

(click to see Properties)

ADD:| DELETE:| GET| GET (ALL)| COUNT

Some options that you can use for each operations:

- **Getting warnings in response:**NITRO allows you to get warnings in an operation by specifying the "warning" query parameter as "yes". For example, to get warnings while connecting to the Citrix ADC appliance, the URL is as follows:

  http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/login?warning=yes

  If any, the warnings are displayed in the response payload with the HTTP code "209 X-NITRO-WARNING".

- **Authenticated access for individual NITRO operations:**NITRO allows you to logon to the Citrix ADC appliance to perform individual operations. You can use this option instead of creating a NITRO session (using the login object) and then using that session to perform all operations,

  To do this, you must specify the username and password in the request header of the NITRO request as follows:

  X-NITRO-USER:<username>

  X-NITRO-PASS:<password>

  Note: In such cases, make sure that the request header DOES not include the following:

  Cookie:NITRO_AUTH_TOKEN=<tokenvalue>

`*Note:`*

Mandatory parameters are marked in redand placeholder content is marked in <green>.

add

URL:http://<Citrix-ADC-IP-address(NSIP)/nitro/v1/config/lsngroup_lsnsipalgprofile_binding

HTTP Method:PUT

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Content-Type:application/json

Request Payload:

```
{

"lsngroup_lsnsipalgprofile_binding":{

<b>"groupname":<String_value>,

</b>"sipalgprofilename":<String_value>

}}
```

Response:

HTTP Status Code on Success: 201 Created HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error

delete

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/lsngroup_lsnsipalgprofile_binding/groupname_value<String>

Query-parameters:

args

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/lsngroup_lsnsipalgprofile_binding/groupname_value<String>?**args=sipalgprofilename:<String_value>**

HTTP Method:DELETE

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue>

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error

get

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/lsngroup_lsnsipalgprofile_binding/groupname_value<String>

Query-parameters:

filter

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/lsngroup_lsnsipalgprofile_binding/groupname_value<String>?**filter=property-name1:property-value1,property-name2:property-value2**

Use this query-parameter to get the filtered set of lsngroup_lsnsipalgprofile_binding resources configured on Citrix ADC. Filtering can be done on any of the properties of the resource.

pagination

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/lsngroup_lsnsipalgprofile_binding/groupname_value<String>?**pagesize=#no;pageno=#no**

Use this query-parameter to get the lsngroup_lsnsipalgprofile_binding resources in chunks.

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "lsngroup_lsnsipalgprofile_binding": [ {

"sipalgprofilename":<String_value>,
```

```
"groupname":<String_value>

}]}
```

get (all)

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/lsngroup_lsnsipalgprofile_binding

Query-parameters:

bulkbindings

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/lsngroup_lsnsipalgprofile_binding?**bulkbindings=yes**

NITRO allows you to fetch bindings in bulk.

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "lsngroup_lsnsipalgprofile_binding": [ {

"sipalgprofilename":<String_value>,

"groupname":<String_value>

}]}
```

count

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/lsngroup_lsnsipalgprofile_binding/groupname_value<String>?**count=yes**

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{"lsngroup_lsnsipalgprofile_binding": [ { "__count": "#no"} ] }
```

lsngroup_lsntransportprofile_binding

Binding object showing the lsntransportprofile that can be bound to lsngroup.

Properties

(click to see Operations)

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
transportprofilename	<String>	Read-write	Name of the LSN transport profile to bind to the specified LSN group. Bind a profile for each protocol for which

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
			you want to specify settings. By default, one LSN transport profile with default settings for TCP, UDP, and ICMP protocols is bound to an LSN group during its creation. This profile is called a default transport. An LSN transport profile that you bind to an LSN group overrides the default LSN transport profile for that protocol.
groupname	<String>	Read-write	Name for the LSN group. Must begin with an ASCII alphanumeric or underscore (_) character, and must contain only ASCII alphanumeric, underscore, hash (#), period (.), space, colon (:), at (@), equals (=), and hyphen (-) characters. Cannot be changed after the LSN group is created. The following requirement applies only to the Citrix ADC CLI: If the name includes one or more spaces,

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
			enclose the name in double or single quotation marks (for example, "lsn group1" or 'lsn group1'). Minimum length = 1 Maximum length = 127
__count	<Double>	Read-write	count parameter

Operations

(click to see Properties)

ADD:| DELETE:| GET| GET (ALL)| COUNT

Some options that you can use for each operations:

- **Getting warnings in response:**NITRO allows you to get warnings in an operation by specifying the "warning" query parameter as "yes". For example, to get warnings while connecting to the Citrix ADC appliance, the URL is as follows:

  http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/login?warning=yes

  If any, the warnings are displayed in the response payload with the HTTP code "209 X-NITRO-WARNING".

- **Authenticated access for individual NITRO operations:**NITRO allows you to logon to the Citrix ADC appliance to perform individual operations. You can use this option instead of creating a NITRO session (using the login object) and then using that session to perform all operations,

  To do this, you must specify the username and password in the request header of the NITRO request as follows:

  X-NITRO-USER:<username>

  X-NITRO-PASS:<password>

Note: In such cases, make sure that the request header DOES not include the following:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue>

`*Note:` *

Mandatory parameters are marked in redand placeholder content is marked in <green>.

add

URL:http://<Citrix-ADC-IP-address(NSIP)/nitro/v1/config/lsngroup_lsntransportprofile_binding

HTTP Method:PUT

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Content-Type:application/json

Request Payload:

```
{

"lsngroup_lsntransportprofile_binding":{

<b>"groupname":<String_value>,

</b>"transportprofilename":<String_value>

}}
```

Response:

HTTP Status Code on Success: 201 Created HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error

delete

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/lsngroup_lsntransportprofile_binding/groupname_value<String>

Query-parameters:

args

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/lsngroup_lsntransportprofile_binding/groupname_value<String>?**args=transportprofilename:<String_value>**

HTTP Method:DELETE

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue>

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error

get

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/lsngroup_lsntransportprofile_binding/groupname_value<String>

Query-parameters:

filter

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/lsngroup_lsntransportprofile_binding/groupname_value<String>?**filter=property-name1:property-value1,property-name2:property-value2**

Use this query-parameter to get the filtered set of lsngroup_lsntransportprofile_binding resources configured on Citrix ADC. Filtering can be done on any of the properties of the resource.

pagination

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/lsngroup_lsntransportprofile_binding/groupname_value<String>?**pagesize=#no;pageno=#no**

Use this query-parameter to get the lsngroup_lsntransportprofile_binding resources in chunks.

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "lsngroup_lsntransportprofile_binding": [ {

"transportprofilename":<String_value>,

"groupname":<String_value>

}]}
```

get (all)

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/lsngroup_lsntransportprofile_binding

Query-parameters:

bulkbindings

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/lsngroup_lsntransportprofile_binding?**bulkbindings=yes**

NITRO allows you to fetch bindings in bulk.

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "lsngroup_lsntransportprofile_binding": [ {

"transportprofilename":<String_value>,

"groupname":<String_value>

}]}
```

count

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/lsngroup_lsntransportprofile_binding/groupname_value<String>?**count=yes**

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{"lsngroup_lsntransportprofile_binding": [ { "__count": "#no"} ] }
```

lsngroup_pcpserver_binding

Binding object showing the pcpserver that can be bound to lsngroup.

Properties

(click to see Operations)

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
pcpserver	<String>	Read-write	Name of the PCP server to be associated with lsn group.
groupname	<String>	Read-write	Name for the LSN group. Must begin with an ASCII alphanumeric or underscore (_) character, and must contain only ASCII alphanumeric, underscore, hash (#), period (.), space, colon (:), at (@), equals (=), and hyphen (-) characters. Cannot be changed after the LSN group is created. The following requirement applies only to the Citrix ADC CLI: If the name includes one or more spaces, enclose the name in double or single quotation marks (for

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
			example, "lsn group1" or 'lsn group1'). Minimum length = 1 Maximum length = 127
__count	<Double>	Read-write	count parameter

Operations

(click to see Properties)

ADD:| DELETE:| GET| GET (ALL)| COUNT

Some options that you can use for each operations:

- **Getting warnings in response:**NITRO allows you to get warnings in an operation by specifying the "warning" query parameter as "yes". For example, to get warnings while connecting to the Citrix ADC appliance, the URL is as follows:

  http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/login?warning=yes

  If any, the warnings are displayed in the response payload with the HTTP code "209 X-NITRO-WARNING".

- **Authenticated access for individual NITRO operations:**NITRO allows you to logon to the Citrix ADC appliance to perform individual operations. You can use this option instead of creating a NITRO session (using the login object) and then using that session to perform all operations,

  To do this, you must specify the username and password in the request header of the NITRO request as follows:

  X-NITRO-USER:<username>

  X-NITRO-PASS:<password>

  Note: In such cases, make sure that the request header DOES not include the following:

  Cookie:NITRO_AUTH_TOKEN=<tokenvalue>

`*Note:` *

Mandatory parameters are marked in redand placeholder content is marked in <green>.

add

URL:http://<Citrix-ADC-IP-address(NSIP)/nitro/v1/config/lsngroup_pcpserver_binding

HTTP Method:PUT

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Content-Type:application/json

Request Payload:

```
{

"lsngroup_pcpserver_binding":{

<b>"groupname":<String_value>,

</b>"pcpserver":<String_value>

}}
```

Response:

HTTP Status Code on Success: 201 Created HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error

delete

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/lsngroup_pcpserver_binding/groupname_value<String>

Query-parameters:

args

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/lsngroup_pcpserver_binding/groupname_value<String>?**args=pcpserver:<String_value>**

HTTP Method:DELETE

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue>

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error

get

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/lsngroup_pcpserver_binding/groupname_value<String>

Query-parameters:

filter

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/lsngroup_pcpserver_binding/groupname_value<String>?**filter=property-name1:property-value1,property-name2:property-value2**

Use this query-parameter to get the filtered set of lsngroup_pcpserver_binding resources configured on Citrix ADC. Filtering can be done on any of the properties of the resource.

pagination

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/lsngroup_pcpserver_binding/groupname_value<String>?**pagesize=#no;pageno=#no**

Use this query-parameter to get the lsngroup_pcpserver_binding resources in chunks.

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "lsngroup_pcpserver_binding": [ {

"pcpserver":<String_value>,

"groupname":<String_value>

}]}
```

get (all)

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/lsngroup_pcpserver_binding

Query-parameters:

bulkbindings

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/lsngroup_pcpserver_binding?**bulkbindings=yes**

NITRO allows you to fetch bindings in bulk.

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "lsngroup_pcpserver_binding": [ {

"pcpserver":<String_value>,

"groupname":<String_value>

}]}
```

count

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/lsngroup_pcpserver_binding/groupname_value<String>?**count=yes**

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{"lsngroup_pcpserver_binding": [ { "__count": "#no"} ] }
```

lsnhttphdrlogprofile

Configuration for LSN HTTP header logging Profile resource.

Properties

(click to see Operations)

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
httphdrlogprofilename	<String>	Read-write	The name of the HTTP header logging Profile. Minimum length = 1 Maximum length = 127
logurl	<String>	Read-write	URL information is logged if option is enabled. Default value: ENABLED Possible values = ENABLED, DISABLED
logmethod	<String>	Read-write	HTTP method information is logged if option is enabled. Default value: ENABLED Possible values = ENABLED, DISABLED
logversion	<String>	Read-write	Version information is logged if option is enabled.

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
			Default value: ENABLED Possible values = ENABLED, DISABLED
loghost	<String>	Read-write	Host information is logged if option is enabled. Default value: ENABLED Possible values = ENABLED, DISABLED
__count	<Double>	Read-only	count parameter

Operations

(click to see Properties)

ADD| DELETE| UPDATE| UNSET| GET (ALL)| GET| COUNT

Some options that you can use for each operations:

- **Getting warnings in response:**NITRO allows you to get warnings in an operation by specifying the "warning" query parameter as "yes". For example, to get warnings while connecting to the Citrix ADC appliance, the URL is as follows:

  http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/login?warning=yes

  If any, the warnings are displayed in the response payload with the HTTP code "209 X-NITRO-WARNING".

- **Authenticated access for individual NITRO operations:**NITRO allows you to logon to the Citrix ADC appliance to perform individual operations. You can use this option instead of creating a NITRO session (using the login object) and then using that session to perform all operations,

To do this, you must specify the username and password in the request header of the NITRO request as follows:

X-NITRO-USER:<username>

X-NITRO-PASS:<password>

Note: In such cases, make sure that the request header DOES not include the following:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue>

`*Note:`*

Mandatory parameters are marked in redand placeholder content is marked in <green>.

add

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/lsnhttphdrlogprofile

HTTP Method:POST

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Content-Type:application/json

Request Payload:

```
{"lsnhttphdrlogprofile":{

<b>"httphdrlogprofilename":<String_value>,

</b>"logurl":<String_value>,

"logmethod":<String_value>,

"logversion":<String_value>,

"loghost":<String_value>

}}
```

Response:

HTTP Status Code on Success: 201 Created HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error

delete

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/lsnhttphdrlogprofile/ httphdrlogprofilename_value<String>

HTTP Method:DELETE

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue>

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error

update

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/lsnhttphdrlogprofile

HTTP Method:PUT

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Content-Type:application/json

Request Payload:

```
{"lsnhttphdrlogprofile":{

<b>"httphdrlogprofilename":<String_value>,

</b>"logurl":<String_value>,

"logmethod":<String_value>,

"logversion":<String_value>,
```

```
"loghost":<String_value>

}}
```

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error

unset

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/lsnhttphdrlogprofile?**action=unset**

HTTP Method:POST

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Content-Type:application/json

Request Payload:

```
{"lsnhttphdrlogprofile":{

<b>"httphdrlogprofilename":<String_value>,

</b>"logurl":true,

"logmethod":true,

"logversion":true,

"loghost":true

}}
```

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error

get (all)

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/lsnhttphdrlogprofile

Query-parameters:

attrs

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/lsnhttphdrlogprofile?**attrs=property-name1,property-name2**

Use this query parameter to specify the resource details that you want to retrieve.

filter

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/lsnhttphdrlogprofile?**filter=property-name1:property-val1,property-name2:property-val2**

Use this query-parameter to get the filtered set of lsnhttphdrlogprofile resources configured on Citrix ADC. Filtering can be done on any of the properties of the resource.

view

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/lsnhttphdrlogprofile?**view=summary**

Note: By default, the retrieved results are displayed in detail view (?view=detail).

pagination

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/lsnhttphdrlogprofile?**pagesize=#no;pageno=#no**

Use this query-parameter to get the lsnhttphdrlogprofile resources in chunks.

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "lsnhttphdrlogprofile": [ {

"httphdrlogprofilename":<String_value>,

"logurl":<String_value>,

"logmethod":<String_value>,

"logversion":<String_value>,

"loghost":<String_value>

}]}
```

get

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/lsnhttphdrlogprofile/ httphdrlogprofilename_value<String>

Query-parameters:

attrs

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/lsnhttphdrlogprofile/ httphdrlogprofilename_value<String>?**attrs=property-name1,property-name2**

Use this query parameter to specify the resource details that you want to retrieve.

view

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/lsnhttphdrlogprofile/ httphdrlogprofilename_value<String>?**view=summary**

Note: By default, the retrieved results are displayed in detail view (?view=detail).

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "lsnhttphdrlogprofile": [ {

"httphdrlogprofilename":<String_value>,

"logurl":<String_value>,

"logmethod":<String_value>,

"logversion":<String_value>,

"loghost":<String_value>

}]}
```

count

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/lsnhttphdrlogprofile?**count=yes**

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "lsnhttphdrlogprofile": [ { "__count": "#no"} ] }
```

lsnip6profile

Configuration for LSN ip6 Profile resource.

Properties

(click to see Operations)

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
name	<String>	Read-write	Name for the LSN ip6 profile. Must begin with an ASCII alphanumeric or underscore (_) character, and must contain only ASCII alphanumeric, underscore, hash (#), period (.), space, colon (:), at (@), equals (=), and hyphen (-) characters. Cannot be changed after the LSN ip6 profile is created. The following

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
			requirement applies only to the Citrix ADC CLI: If the name includes one or more spaces, enclose the name in double or single quotation marks (for example, "lsn ip6 profile1" or 'lsn ip6 profile1'). Minimum length = 1 Maximum length = 127
type	<String>	Read-write	IPv6 translation type for which to set the LSN IP6 profile parameters. Possible values = DS-Lite, NAT64
natprefix	<String>	Read-write	IPv6 address(es) of the LSN subscriber(s) or subscriber network(s) on whose traffic you want the Citrix ADC to perform Large Scale NAT.
network6	<String>	Read-write	IPv6 address of the Citrix ADC AFTR device.
__count	<Double>	Read-only	count parameter

Operations

(click to see Properties)

ADD| DELETE| GET (ALL)| GET| COUNT

Some options that you can use for each operations:

- **Getting warnings in response:**NITRO allows you to get warnings in an operation by specifying the "warning" query parameter as "yes". For example, to get warnings while connecting to the Citrix ADC appliance, the URL is as follows:

  http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/login?warning=yes

  If any, the warnings are displayed in the response payload with the HTTP code "209 X-NITRO-WARNING".

- **Authenticated access for individual NITRO operations:**NITRO allows you to logon to the Citrix ADC appliance to perform individual operations. You can use this option instead of creating a NITRO session (using the login object) and then using that session to perform all operations,

  To do this, you must specify the username and password in the request header of the NITRO request as follows:

  X-NITRO-USER:<username>

  X-NITRO-PASS:<password>

  Note: In such cases, make sure that the request header DOES not include the following:

  Cookie:NITRO_AUTH_TOKEN=<tokenvalue>

`*Note:`*

Mandatory parameters are marked in redand placeholder content is marked in <green>.

add

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/lsnip6profile

HTTP Method:POST

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Content-Type:application/json

Request Payload:

```
{"lsnip6profile":{

<b>"name":<String_value>,

</b><b>"type":<String_value>,

</b>"natprefix":<String_value>,

"network6":<String_value>

}}
```

Response:

HTTP Status Code on Success: 201 Created HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error

delete

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/lsnip6profile/name_value<String>

HTTP Method:DELETE

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue>

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error

get (all)

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/lsnip6profile

Query-parameters:

attrs

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/lsnip6profile?**attrs=property-name1,property-name2**

Use this query parameter to specify the resource details that you want to retrieve.

filter

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/lsnip6profile?**filter=property-name1:property-val1,property-name2:property-val2**

Use this query-parameter to get the filtered set of lsnip6profile resources configured on Citrix ADC. Filtering can be done on any of the properties of the resource.

view

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/lsnip6profile?**view=summary**

Note: By default, the retrieved results are displayed in detail view (?view=detail).

pagination

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/lsnip6profile?**pagesize=#no;pageno=#no**

Use this query-parameter to get the lsnip6profile resources in chunks.

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "lsnip6profile": [ {
```

```
"name":<String_value>,

"type":<String_value>,

"network6":<String_value>,

"natprefix":<String_value>

}]}
```

get

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/lsnip6profile/name_value<String>

Query-parameters:

attrs

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/lsnip6profile/name_value<String>?**attrs=property-name1,property-name2**

Use this query parameter to specify the resource details that you want to retrieve.

view

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/lsnip6profile/name_value<String>?**view=summary**

Note: By default, the retrieved results are displayed in detail view (?view=detail).

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "lsnip6profile": [ {

"name":<String_value>,

"type":<String_value>,

"network6":<String_value>,

"natprefix":<String_value>

}]}
```

count

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/lsnip6profile?**count=yes**

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "lsnip6profile": [ { "__count": "#no"} ] }
```

lsnlogprofile

Configuration for LSN logging Profile resource.

Properties

(click to see Operations)

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
logprofilename	<String>	Read-write	The name of the logging Profile. Minimum length = 1 Maximum length = 127
logsubscrinfo	<String>	Read-write	Subscriber ID information is logged if option is enabled. Default value: DISABLED Possible values = ENABLED, DISABLED
logcompact	<String>	Read-write	Logs in Compact Logging format if option is enabled. Default value: ENABLED Possible values = ENABLED, DISABLED
logipfix	<String>	Read-write	Logs in IPFIX format if option is enabled. Default value: DISABLED

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
			Possible values = ENABLED, DISABLED
analyticsprofile	<String>	Read-write	Name of the Analytics Profile attached to this lsn profile.
logsessdeletion	<String>	Read-write	LSN Session deletion will not be logged if disabled. Default value: ENABLED Possible values = ENABLED, DISABLED
__count	<Double>	Read-only	count parameter

Operations

(click to see Properties)

ADD| DELETE| UPDATE| UNSET| GET (ALL)| GET| COUNT

Some options that you can use for each operations:

- **Getting warnings in response:**NITRO allows you to get warnings in an operation by specifying the "warning" query parameter as "yes". For example, to get warnings while connecting to the Citrix ADC appliance, the URL is as follows:

  http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/login?warning=yes

  If any, the warnings are displayed in the response payload with the HTTP code "209 X-NITRO-WARNING".

- **Authenticated access for individual NITRO operations:**NITRO allows you to logon to the Citrix ADC appliance to perform individual operations. You can use this option instead of

creating a NITRO session (using the login object) and then using that session to perform all operations,

To do this, you must specify the username and password in the request header of the NITRO request as follows:

X-NITRO-USER:<username>

X-NITRO-PASS:<password>

Note: In such cases, make sure that the request header DOES not include the following:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue>

`*Note:`*

Mandatory parameters are marked in redand placeholder content is marked in <green>.

add

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/lsnlogprofile

HTTP Method:POST

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Content-Type:application/json

Request Payload:

```
{"lsnlogprofile":{

<b>"logprofilename":<String_value>,

</b>"logsubscrinfo":<String_value>,

"logcompact":<String_value>,

"logipfix":<String_value>,

"analyticsprofile":<String_value>,

"logsessdeletion":<String_value>
```

```
}}
```

Response:

HTTP Status Code on Success: 201 Created HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error

delete

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/lsnlogprofile/logprofilename_value<String>

HTTP Method:DELETE

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue>

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error

update

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/lsnlogprofile

HTTP Method:PUT

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Content-Type:application/json

Request Payload:

```
{"lsnlogprofile":{

<b>"logprofilename":<String_value>,
```

```
</b>"logsubscrinfo":<String_value>,

"logcompact":<String_value>,

"logipfix":<String_value>,

"analyticsprofile":<String_value>,

"logsessdeletion":<String_value>

}}
```

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error

unset

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/lsnlogprofile?**action=unset**

HTTP Method:POST

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Content-Type:application/json

Request Payload:

```
{"lsnlogprofile":{

<b>"logprofilename":<String_value>,

</b>"logsubscrinfo":true,

"logcompact":true,

"logipfix":true,
```

```
"analyticsprofile":true,

"logsessdeletion":true

}}
```

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error

get (all)

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/lsnlogprofile

Query-parameters:

attrs

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/lsnlogprofile?**attrs=property-name1,property-name2**

Use this query parameter to specify the resource details that you want to retrieve.

filter

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/lsnlogprofile?**filter=property-name1:property-val1,property-name2:property-val2**

Use this query-parameter to get the filtered set of lsnlogprofile resources configured on Citrix ADC. Filtering can be done on any of the properties of the resource.

view

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/lsnlogprofile?**view=summary**

Note: By default, the retrieved results are displayed in detail view (?view=detail).

pagination

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/lsnlogprofile?**pagesize=#no;pageno=#no**

Use this query-parameter to get the lsnlogprofile resources in chunks.

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "lsnlogprofile": [ {

"logprofilename":<String_value>,

"logsubscrinfo":<String_value>,

"logcompact":<String_value>,

"logipfix":<String_value>,

"analyticsprofile":<String_value>,

"logsessdeletion":<String_value>

}]}
```

get

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/lsnlogprofile/logprofilename_value<String>

Query-parameters:

attrs

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/lsnlogprofile/logprofilename_value<String>?**attrs=property-name1,property-name2**

Use this query parameter to specify the resource details that you want to retrieve.

view

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/lsnlogprofile/logprofilename_value<String>?**view=summary**

Note: By default, the retrieved results are displayed in detail view (?view=detail).

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "lsnlogprofile": [ {

"logprofilename":<String_value>,

"logsubscrinfo":<String_value>,

"logcompact":<String_value>,

"logipfix":<String_value>,

"analyticsprofile":<String_value>,
```

```
"logsessdeletion":<String_value>

}]}
```

count

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/lsnlogprofile?**count=yes**

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "lsnlogprofile": [ { "__count": "#no"} ] }
```

lsnparameter

Configuration for LSN parameter resource.

Properties

(click to see Operations)

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
memlimit	<Double>	Read-write	Amount of Citrix ADC memory to reserve

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
			for the LSN feature, in multiples of 2MB. Note: If you later reduce the value of this parameter, the amount of active memory is not reduced. Changing the configured memory limit can only increase the amount of active memory. This command is deprecated, use 'set extendedmemoryparam -memlimit' instead.
sessionsync	<String>	Read-write	Synchronize all LSN sessions with the secondary node in a high availability (HA) deployment (global synchronization). After a failover, established TCP connections and UDP packet flows are kept active and resumed on the secondary node (new primary). The global session synchronization parameter and session

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
			synchronization parameters (group level) of all LSN groups are enabled by default. For a group, when both the global level and the group level LSN session synchronization parameters are enabled, the primary node synchronizes information of all LSN sessions related to this LSN group with the secondary node. Default value: ENABLED Possible values = ENABLED, DISABLED
subscrsessionremoval	<String>	Read-write	LSN global setting for controlling subscriber aware session removal, when this is enabled, when ever the subscriber info is deleted from subscriber database, sessions corresponding to that

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
			subscriber will be removed. if this setting is disabled, subscriber sessions will be timed out as per the idle time out settings. Default value: DISABLED Possible values = ENABLED, DISABLED
memlimitactive	<Double>	Read-only	Amount of actual memory reserved for the LSN feature. The amount of active memory for the LSN feature might be less than the configured memory, because the available memory is shared across features.
maxmemlimit	<Double>	Read-only	Maximum amount of Citrix ADC memory that can be reserved for the LSN feature.

Operations

(click to see Properties)

UPDATE| UNSET| GET (ALL)

Some options that you can use for each operations:

- **Getting warnings in response:**NITRO allows you to get warnings in an operation by specifying the "warning" query parameter as "yes". For example, to get warnings while connecting to the Citrix ADC appliance, the URL is as follows:

  http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/login?warning=yes

  If any, the warnings are displayed in the response payload with the HTTP code "209 X-NITRO-WARNING".

- **Authenticated access for individual NITRO operations:**NITRO allows you to logon to the Citrix ADC appliance to perform individual operations. You can use this option instead of creating a NITRO session (using the login object) and then using that session to perform all operations,

  To do this, you must specify the username and password in the request header of the NITRO request as follows:

  X-NITRO-USER:<username>

  X-NITRO-PASS:<password>

  Note: In such cases, make sure that the request header DOES not include the following:

  Cookie:NITRO_AUTH_TOKEN=<tokenvalue>

`*Note:`*

Mandatory parameters are marked in redand placeholder content is marked in <green>.

update

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/lsnparameter

HTTP Method:PUT

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Content-Type:application/json

Request Payload:

```
{"lsnparameter":{
```

```
"memlimit":<Double_value>,

"sessionsync":<String_value>,

"subscrsessionremoval":<String_value>

}}
```

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error

unset

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/lsnparameter?**action=unset**

HTTP Method:POST

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Content-Type:application/json

Request Payload:

```
{"lsnparameter":{

"memlimit":true,

"sessionsync":true,

"subscrsessionremoval":true

}}
```

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error

get (all)

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/lsnparameter

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "lsnparameter": [ {

"memlimitactive":<Double_value>,

"maxmemlimit":<Double_value>,

"memlimit":<Double_value>,

"sessionsync":<String_value>,

"subscrsessionremoval":<String_value>

}]}
```

lsnpool

Configuration for LSN pool resource.

Properties

(click to see Operations)

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
poolname	<String>	Read-write	Name for the LSN pool. Must begin with an ASCII alphanumeric or underscore (_) character, and must contain only ASCII alphanumeric, underscore, hash (#), period (.), space, colon (:), at (@), equals (=), and hyphen (-) characters. Cannot be changed after the LSN pool is created. The following requirement applies only to the Citrix ADC CLI: If the name includes one or more spaces, enclose the name in double or single quotation marks (for example, "lsn pool1" or 'lsn pool1'). Minimum length = 1 Maximum length = 127
nattype	<String>	Read-write	Type of NAT IP address and port allocation (from the LSN pools bound to an LSN group) for

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
			subscribers (of the LSN client entity bound to the LSN group): Available options function as follows: * Deterministic - Allocate a NAT IP address and a block of ports to each subscriber (of the LSN client bound to the LSN group). The Citrix ADC sequentially allocates NAT resources to these subscribers. The Citrix ADC ADC assigns the first block of ports (block size determined by the port block size parameter of the LSN group) on the beginning NAT IP address to the beginning subscriber IP address. The next range of ports is assigned to the next subscriber, and so on, until the NAT address does not have enough ports

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
			for the next subscriber. In this case, the first port block on the next NAT address is used for the subscriber, and so on. Because each subscriber now receives a deterministic NAT IP address and a block of ports, a subscriber can be identified without any need for logging. For a connection, a subscriber can be identified based only on the NAT IP address and port, and the destination IP address and port. * Dynamic - Allocate a random NAT IP address and a port from the LSN NAT pool for a subscriber's connection. If port block allocation is enabled (in LSN pool) and a port block size is

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
			specified (in the LSN group), the Citrix ADC allocates a random NAT IP address and a block of ports for a subscriber when it initiates a connection for the first time. The ADC allocates this NAT IP address and a port (from the allocated block of ports) for different connections from this subscriber. If all the ports are allocated (for different subscriber's connections) from the subscriber's allocated port block, the ADC allocates a new random port block for the subscriber. Only LSN Pools and LSN groups with the same NAT type settings can be bound together. Multiples LSN pools can be bound to an

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
			LSN group. A maximum of 16 LSN pools can be bound to an LSN group. . Default value: DYNAMIC Possible values = DYNAMIC, DETERMINISTIC
portblockallocation	<String>	Read-write	Allocate a random NAT port block, from the available NAT port pool of an NAT IP address, for each subscriber when the NAT allocation is set as Dynamic NAT. For any connection initiated from a subscriber, the Citrix ADC allocates a NAT port from the subscriber's allocated NAT port block to create the LSN session. You must set the port block size in the bound LSN group. For a subscriber, if all the ports are allocated from the subscriber's allocated port block,

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
			the Citrix ADC allocates a new random port block for the subscriber. For Deterministic NAT, this parameter is enabled by default, and you cannot disable it. Default value: DISABLED Possible values = ENABLED, DISABLED
portrealloctimeout	<Double>	Read-write	The waiting time, in seconds, between deallocating LSN NAT ports (when an LSN mapping is removed) and reallocating them for a new LSN session. This parameter is necessary in order to prevent collisions between old and new mappings and sessions. It ensures that all established sessions are broken instead of redirected to a different subscriber. This is not applicable for ports used in:

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
			* Deterministic NAT * Address-Dependent filtering and Address-Port-Dependent filtering * Dynamic NAT with port block allocation In these cases, ports are immediately reallocated. Default value: 0 Minimum value = 0 Maximum value = 600
maxportrealloctmq	<Double>	Read-write	Maximum number of ports for which the port reallocation timeout applies for each NAT IP address. In other words, the maximum deallocated-port queue size for which the reallocation timeout applies for each NAT IP address. When the queue size is full, the next port deallocated is reallocated

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
			immediately for a new LSN session. Default value: 65536 Minimum value = 0 Maximum value = 65536
__count	<Double>	Read-only	count parameter

Operations

(click to see Properties)

ADD| DELETE| UPDATE| UNSET| GET (ALL)| GET| COUNT

Some options that you can use for each operations:

- **Getting warnings in response:**NITRO allows you to get warnings in an operation by specifying the "warning" query parameter as "yes". For example, to get warnings while connecting to the Citrix ADC appliance, the URL is as follows:

  http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/login?warning=yes

  If any, the warnings are displayed in the response payload with the HTTP code "209 X-NITRO-WARNING".
- **Authenticated access for individual NITRO operations:**NITRO allows you to logon to the Citrix ADC appliance to perform individual operations. You can use this option instead of creating a NITRO session (using the login object) and then using that session to perform all operations,

  To do this, you must specify the username and password in the request header of the NITRO request as follows:

  X-NITRO-USER:<username>

  X-NITRO-PASS:<password>

  Note: In such cases, make sure that the request header DOES not include the following:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue>

`*Note:` *

Mandatory parameters are marked in redand placeholder content is marked in <green>.

add

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/lsnpool

HTTP Method:POST

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Content-Type:application/json

Request Payload:

```
{"lsnpool":{

<b>"poolname":<String_value>,

</b>"nattype":<String_value>,

"portblockallocation":<String_value>,

"portrealloctimeout":<Double_value>,

"maxportrealloctmq":<Double_value>

}}
```

Response:

HTTP Status Code on Success: 201 Created HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error

delete

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/lsnpool/poolname_value<String>

HTTP Method:DELETE

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue>

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error

update

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/lsnpool

HTTP Method:PUT

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Content-Type:application/json

Request Payload:

```
{"lsnpool":{

<b>"poolname":<String_value>,

</b>"portrealloctimeout":<Double_value>,

"maxportrealloctmq":<Double_value>

}}
```

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error

unset

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/lsnpool?**action=unset**

HTTP Method:POST

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Content-Type:application/json

Request Payload:

```
{"lsnpool":{

<b>"poolname":<String_value>,

</b>"portrealloctimeout":true,

"maxportrealloctmq":true

}}
```

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error

get (all)

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/lsnpool

Query-parameters:

attrs

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/lsnpool?**attrs=property-name1,property-name2**

Use this query parameter to specify the resource details that you want to retrieve.

filter

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/lsnpool?**filter=property-name1:property-val1,property-name2:property-val2**

Use this query-parameter to get the filtered set of lsnpool resources configured on Citrix ADC. Filtering can be done on any of the properties of the resource.

view

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/lsnpool?**view=summary**

Note: By default, the retrieved results are displayed in detail view (?view=detail).

pagination

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/lsnpool?**pagesize=#no;pageno=#no**

Use this query-parameter to get the lsnpool resources in chunks.

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "lsnpool": [ {

"poolname":<String_value>,

"nattype":<String_value>,

"portrealloctimeout":<Double_value>,

"maxportrealloctmq":<Double_value>,
```

```
"portblockallocation":<String_value>

}]}
```

get

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/lsnpool/poolname_value<String>

Query-parameters:

attrs

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/lsnpool/poolname_value<String>?**attrs=property-name1,property-name2**

Use this query parameter to specify the resource details that you want to retrieve.

view

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/lsnpool/poolname_value<String>?**view=summary**

Note: By default, the retrieved results are displayed in detail view (?view=detail).

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "lsnpool": [ {
```

```
"poolname":<String_value>,

"nattype":<String_value>,

"portrealloctimeout":<Double_value>,

"maxportrealloctmq":<Double_value>,

"portblockallocation":<String_value>

}]}
```

count

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/lsnpool?**count=yes**

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "lsnpool": [ { "__count": "#no"} ] }
```

lsnpool_binding

Binding object which returns the resources bound to lsnpool.

Properties

(click to see Operations)

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
poolname	<String>	Read-write	Name for the LSN pool. Must begin with an ASCII alphanumeric or underscore (_) character, and must contain only ASCII alphanumeric, underscore, hash (#), period (.), space, colon (:), at (@), equals (=), and hyphen (-) characters. Cannot be changed after the LSN pool is created. The following requirement applies only to the Citrix ADC CLI: If the name includes one or more spaces, enclose the name in double or single quotation marks (for example, "lsn pool1" or 'lsn pool1'). Minimum length = 1 Maximum length = 127
lsnpool_lsnip_binding	<lsnpool_lsnip_binding[]>	Read-only	lsnip that can be bound to lsnpool.

Operations

(click to see Properties)

GET| GET (ALL)

Some options that you can use for each operations:

- **Getting warnings in response:**NITRO allows you to get warnings in an operation by specifying the "warning" query parameter as "yes". For example, to get warnings while connecting to the Citrix ADC appliance, the URL is as follows:

  http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/login?warning=yes

  If any, the warnings are displayed in the response payload with the HTTP code "209 X-NITRO-WARNING".

- **Authenticated access for individual NITRO operations:**NITRO allows you to logon to the Citrix ADC appliance to perform individual operations. You can use this option instead of creating a NITRO session (using the login object) and then using that session to perform all operations,

  To do this, you must specify the username and password in the request header of the NITRO request as follows:

  X-NITRO-USER:<username>

  X-NITRO-PASS:<password>

  Note: In such cases, make sure that the request header DOES not include the following:

  Cookie:NITRO_AUTH_TOKEN=<tokenvalue>

`*Note:`*

Mandatory parameters are marked in redand placeholder content is marked in <green>.

get

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/lsnpool_binding/poolname_value<String>

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "lsnpool_binding": [ {

"poolname":<String_value>,

"lsnpool_lsnip_binding":<lsnpool_lsnip_binding[]_value>

}]}
```

get (all)

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/lsnpool_binding

Query-parameters:

bulkbindings

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/lsnpool_binding?**bulkbindings=yes**

NITRO allows you to fetch bindings in bulk.

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "lsnpool_binding": [ {

"poolname":<String_value>,

"lsnpool_lsnip_binding":<lsnpool_lsnip_binding[]_value>

}]}
```

lsnpool_lsnip_binding

Binding object showing the lsnip that can be bound to lsnpool.

Properties

(click to see Operations)

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
poolname	<String>	Read-write	Name for the LSN pool. Must begin with an ASCII alphanumeric or underscore (_) character, and must contain only ASCII alphanumeric, underscore, hash (#), period (.), space, colon (:), at (@), equals (=), and hyphen (-) characters. Cannot

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
			be changed after the LSN pool is created. The following requirement applies only to the Citrix ADC CLI: If the name includes one or more spaces, enclose the name in double or single quotation marks (for example, "lsn pool1" or 'lsn pool1'). Minimum length = 1 Maximum length = 127
ownernode	<Double>	Read-write	ID(s) of cluster node(s) on which command is to be executed. Minimum value = 0 Maximum value = 31
lsnip	<String>	Read-write	IPv4 address or a range of IPv4 addresses to be used as NAT IP address(es) for LSN. After the pool is created, these IPv4 addresses are added to the Citrix ADC as Citrix ADC owned IP address of type LSN. A maximum of 4096 IP addresses can be bound to an LSN

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
			pool. An LSN IP address associated with an LSN pool cannot be shared with other LSN pools. IP addresses specified for this parameter must not already exist on the Citrix ADC as any Citrix ADC owned IP addresses. In the command line interface, separate the range with a hyphen. For example: 10.102.29.30-10.102.29.189. You can later remove some or all the LSN IP addresses from the pool, and add IP addresses to the LSN pool. By default , arp is enabled on LSN IP address but, you can disable it using command - "set ns ip" . Minimum length = 1
__count	<Double>	Read-write	count parameter

Operations

(click to see Properties)

ADD:| DELETE:| GET| GET (ALL)| COUNT

Some options that you can use for each operations:

- **Getting warnings in response:**NITRO allows you to get warnings in an operation by specifying the "warning" query parameter as "yes". For example, to get warnings while connecting to the Citrix ADC appliance, the URL is as follows:

  http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/login?warning=yes

  If any, the warnings are displayed in the response payload with the HTTP code "209 X-NITRO-WARNING".

- **Authenticated access for individual NITRO operations:**NITRO allows you to logon to the Citrix ADC appliance to perform individual operations. You can use this option instead of creating a NITRO session (using the login object) and then using that session to perform all operations,

  To do this, you must specify the username and password in the request header of the NITRO request as follows:

  X-NITRO-USER:<username>

  X-NITRO-PASS:<password>

  Note: In such cases, make sure that the request header DOES not include the following:

  Cookie:NITRO_AUTH_TOKEN=<tokenvalue>

`*Note:`*

Mandatory parameters are marked in redand placeholder content is marked in <green>.

add

URL:http://<Citrix-ADC-IP-address(NSIP)/nitro/v1/config/lsnpool_lsnip_binding

HTTP Method:PUT

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Content-Type:application/json

Request Payload:

```
{

"lsnpool_lsnip_binding":{

<b>"poolname":<String_value>,

</b>"ownernode":<Double_value>,

"lsnip":<String_value>

}}
```

Response:

HTTP Status Code on Success: 201 Created HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error

delete

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/lsnpool_lsnip_binding/poolname_value<String>

Query-parameters:

args

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/lsnpool_lsnip_binding/poolname_value<String>?**args=ownernode:<Double_value>,lsnip:<String_value>**

HTTP Method:DELETE

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue>

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error

get

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/lsnpool_lsnip_binding/poolname_value<String>

Query-parameters:

filter

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/lsnpool_lsnip_binding/poolname_value<String>?**filter=property-name1:property-value1,property-name2:property-value2**

Use this query-parameter to get the filtered set of lsnpool_lsnip_binding resources configured on Citrix ADC. Filtering can be done on any of the properties of the resource.

pagination

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/lsnpool_lsnip_binding/poolname_value<String>?**pagesize=#no;pageno=#no**

Use this query-parameter to get the lsnpool_lsnip_binding resources in chunks.

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "lsnpool_lsnip_binding": [ {

"poolname":<String_value>,
```

```
"ownernode":<Double_value>,

"lsnip":<String_value>

}]}
```

get (all)

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/lsnpool_lsnip_binding

Query-parameters:

bulkbindings

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/lsnpool_lsnip_binding?**bulkbindings=yes**

NITRO allows you to fetch bindings in bulk.

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "lsnpool_lsnip_binding": [ {

"poolname":<String_value>,

"ownernode":<Double_value>,

"lsnip":<String_value>
```

```
}]}
```

count

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/lsnpool_lsnip_binding/poolname_value<String>?**count=yes**

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{"lsnpool_lsnip_binding": [ { "__count": "#no"} ] }
```

lsnrtspalgprofile

Configuration for LSN RTSPALG Profile resource.

Properties

(click to see Operations)

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
rtspalgprofilename	<String>	Read-write	The name of the RTSPALG Profile. Minimum length = 1

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
			Maximum length = 127
rtspidletimeout	<Double>	Read-write	Idle timeout for the rtsp sessions in seconds. Default value: 120
rtspportrange	<String>	Read-write	port for the RTSP.
rtsptransportprotocol	<String>	Read-write	RTSP ALG Profile transport protocol type. Default value: TCP Possible values = TCP, UDP
__count	<Double>	Read-only	count parameter

Operations

(click to see Properties)

ADD| UPDATE| UNSET| DELETE| GET (ALL)| GET| COUNT

Some options that you can use for each operations:

- **Getting warnings in response:**NITRO allows you to get warnings in an operation by specifying the "warning" query parameter as "yes". For example, to get warnings while connecting to the Citrix ADC appliance, the URL is as follows:

  http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/login?warning=yes

  If any, the warnings are displayed in the response payload with the HTTP code "209 X-NITRO-WARNING".

- **Authenticated access for individual NITRO operations:**NITRO allows you to logon to the Citrix ADC appliance to perform individual operations. You can use this option instead of creating a NITRO session (using the login object) and then using that session to perform all operations,

To do this, you must specify the username and password in the request header of the NITRO request as follows:

X-NITRO-USER:<username>

X-NITRO-PASS:<password>

Note: In such cases, make sure that the request header DOES not include the following:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue>

`*Note:`*

Mandatory parameters are marked in redand placeholder content is marked in <green>.

add

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/lsnrtspalgprofile

HTTP Method:POST

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Content-Type:application/json

Request Payload:

```
{"lsnrtspalgprofile":{

<b>"rtspalgprofilename":<String_value>,

</b>"rtspidletimeout":<Double_value>,

<b>"rtspportrange":<String_value>,

</b>"rtsptransportprotocol":<String_value>

}}
```

Response:

HTTP Status Code on Success: 201 Created HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error

update

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/lsnrtspalgprofile

HTTP Method:PUT

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Content-Type:application/json

Request Payload:

```
{"lsnrtspalgprofile":{

<b>"rtspalgprofilename":<String_value>,

</b>"rtspidletimeout":<Double_value>,

"rtspportrange":<String_value>,

"rtsptransportprotocol":<String_value>

}}
```

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error

unset

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/lsnrtspalgprofile?**action=unset**

HTTP Method:POST

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Content-Type:application/json

Request Payload:

```
{"lsnrtspalgprofile":{

<b>"rtspalgprofilename":<String_value>,

</b>"rtspidletimeout":true,

"rtspportrange":true,

"rtsptransportprotocol":true

}}
```

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error

delete

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/lsnrtspalgprofile/rtspalgprofilename_value<String>

HTTP Method:DELETE

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue>

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error

get (all)

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/lsnrtspalgprofile

Query-parameters:

attrs

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/lsnrtspalgprofile?**attrs=property-name1,property-name2**

Use this query parameter to specify the resource details that you want to retrieve.

filter

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/lsnrtspalgprofile?**filter=property-name1:property-val1,property-name2:property-val2**

Use this query-parameter to get the filtered set of lsnrtspalgprofile resources configured on Citrix ADC. Filtering can be done on any of the properties of the resource.

view

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/lsnrtspalgprofile?**view=summary**

Note: By default, the retrieved results are displayed in detail view (?view=detail).

pagination

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/lsnrtspalgprofile?**pagesize=#no;pageno=#no**

Use this query-parameter to get the lsnrtspalgprofile resources in chunks.

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "lsnrtspalgprofile": [ {

"rtspalgprofilename":<String_value>,

"rtspidletimeout":<Double_value>,

"rtspportrange":<String_value>,

"rtsptransportprotocol":<String_value>

}]}
```

get

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/lsnrtspalgprofile/rtspalgprofilename_value<String>

Query-parameters:

attrs

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/lsnrtspalgprofile/rtspalgprofilename_value<String>?**attrs=property-name1,property-name2**

Use this query parameter to specify the resource details that you want to retrieve.

view

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/lsnrtspalgprofile/rtspalgprofilename_value<String>?**view=summary**

Note: By default, the retrieved results are displayed in detail view (?view=detail).

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "lsnrtspalgprofile": [ {

"rtspalgprofilename":<String_value>,

"rtspidletimeout":<Double_value>,

"rtspportrange":<String_value>,

"rtsptransportprotocol":<String_value>

}]}
```

count

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/lsnrtspalgprofile?**count=yes**

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "lsnrtspalgprofile": [ { "__count": "#no"} ] }
```

lsnrtspalgsession

Configuration for LSN RTSPALG session resource.

Properties

(click to see Operations)

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
sessionid	<String>	Read-write	Session ID for the RTSP call.
nodeid	<Double>	Read-write	Unique number that identifies the cluster node. Minimum value = 0 Maximum value = 31
callflags	<Double>	Read-only	Flags for the call entry.
xlatip	<String>	Read-only	XLAT IP Address if its XLAT session.
callrefcount	<Double>	Read-only	Reference count for the call entry.
calltimer	<Double>	Read-only	Timer for the call entry.
__count	<Double>	Read-only	count parameter

Operations

(click to see Properties)

FLUSH| GET (ALL)| GET| COUNT

Some options that you can use for each operations:

- **Getting warnings in response:**NITRO allows you to get warnings in an operation by specifying the "warning" query parameter as "yes". For example, to get warnings while connecting to the Citrix ADC appliance, the URL is as follows:

  http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/login?warning=yes

  If any, the warnings are displayed in the response payload with the HTTP code "209 X-NITRO-WARNING".

- **Authenticated access for individual NITRO operations:**NITRO allows you to logon to the Citrix ADC appliance to perform individual operations. You can use this option instead of creating a NITRO session (using the login object) and then using that session to perform all operations,

  To do this, you must specify the username and password in the request header of the NITRO request as follows:

  X-NITRO-USER:<username>

  X-NITRO-PASS:<password>

  Note: In such cases, make sure that the request header DOES not include the following:

  Cookie:NITRO_AUTH_TOKEN=<tokenvalue>

`*Note:`*

Mandatory parameters are marked in redand placeholder content is marked in <green>.

flush

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/lsnrtspalgsession?**action=flush**

HTTP Method:POST

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Content-Type:application/json

Request Payload:

```
{"lsnrtspalgsession":{

<b>"sessionid":<String_value>

</b>}}
```

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error

get (all)

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/lsnrtspalgsession

Query-parameters:

args

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/lsnrtspalgsession?**args=sessionid:<String_value>,nodeid:<Double_value>**

Use this query-parameter to get lsnrtspalgsession resources based on additional properties.

attrs

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/lsnrtspalgsession?**attrs=property-name1,property-name2**

Use this query parameter to specify the resource details that you want to retrieve.

filter

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/lsnrtspalgsession?**filter=property-name1:property-val1,property-name2:property-val2**

Use this query-parameter to get the filtered set of lsnrtspalgsession resources configured on Citrix ADC. Filtering can be done on any of the properties of the resource.

view

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/lsnrtspalgsession?**view=summary**

Note: By default, the retrieved results are displayed in detail view (?view=detail).

pagination

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/lsnrtspalgsession?**pagesize=#no;pageno=#no**

Use this query-parameter to get the lsnrtspalgsession resources in chunks.

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "lsnrtspalgsession": [ {

sessionid:<String_value>,nodeid:<Double_value>"callflags":<Double_value>,

"xlatip":<String_value>,

"callrefcount":<Double_value>,

"calltimer":<Double_value>

}]}
```

get

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/lsnrtspalgsession/sessionid_value<String>

Query-parameters:

attrs

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/lsnrtspalgsession/sessionid_value<String>?**attrs=property-name1,property-name2**

Use this query parameter to specify the resource details that you want to retrieve.

view

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/lsnrtspalgsession/sessionid_value<String>?**view=summary**

Note: By default, the retrieved results are displayed in detail view (?view=detail).

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "lsnrtspalgsession": [ {

sessionid:<String_value>,nodeid:<Double_value>"callflags":<Double_value>,

"xlatip":<String_value>,
```

```
"callrefcount":<Double_value>,

"calltimer":<Double_value>

}]}
```

count

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/lsnrtspalgsession?**count=yes**

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "lsnrtspalgsession": [ { "__count": "#no"} ] }
```

lsnrtspalgsession_binding

Binding object which returns the resources bound to lsnrtspalgsession.

Properties

(click to see Operations)

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
sessionid	<String>	Read-write	Session ID for the RTSP call.
lsnrtspalgsession_datachannel_binding	<lsnrtspalgsession_datachannel_binding[]>	Read-only	datachannel that can be bound to lsnrtspalgsession.

Operations

(click to see Properties)

GET| GET (ALL)

Some options that you can use for each operations:

- **Getting warnings in response:**NITRO allows you to get warnings in an operation by specifying the "warning" query parameter as "yes". For example, to get warnings while connecting to the Citrix ADC appliance, the URL is as follows:

  http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/login?warning=yes

  If any, the warnings are displayed in the response payload with the HTTP code "209 X-NITRO-WARNING".

- **Authenticated access for individual NITRO operations:**NITRO allows you to logon to the Citrix ADC appliance to perform individual operations. You can use this option instead of creating a NITRO session (using the login object) and then using that session to perform all operations,

  To do this, you must specify the username and password in the request header of the NITRO request as follows:

  X-NITRO-USER:<username>

  X-NITRO-PASS:<password>

  Note: In such cases, make sure that the request header DOES not include the following:

  Cookie:NITRO_AUTH_TOKEN=<tokenvalue>

`*Note:`*

Mandatory parameters are marked in redand placeholder content is marked in <green>.

get

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/lsnrtspalgsession_binding/sessionid_value<String>

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "lsnrtspalgsession_binding": [ {

"sessionid":<String_value>,

"lsnrtspalgsession_datachannel_binding":<lsnrtspalgsession_datachannel_binding[]_value>

}]}
```

get (all)

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/lsnrtspalgsession_binding

Query-parameters:

bulkbindings

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/lsnrtspalgsession_binding?**bulkbindings=yes**

NITRO allows you to fetch bindings in bulk.

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "lsnrtspalgsession_binding": [ {

"sessionid":<String_value>,

"lsnrtspalgsession_datachannel_binding":<lsnrtspalgsession_datachannel_binding[]_value>

}]}
```

lsnrtspalgsession_datachannel_binding

Binding object showing the datachannel that can be bound to lsnrtspalgsession.

Properties

(click to see Operations)

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
channelip	<String>	Read-write	IP address of the channel.
sessionid	<String>	Read-write	Session ID for the RTSP call.

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
channelnatip	<String>	Read-only	Natted IP address of the channel.
channelport	<Integer>	Read-only	port for the channel.
channelflags	<Double>	Read-only	Flags for the call entry.
channeltimeout	<Double>	Read-only	Timeout for the channel.
channelnatport	<Integer>	Read-only	Natted port for the channel.
channelprotocol	<String>	Read-only	Channel transport protocol. Possible values = HTTP, FTP, TCP, UDP, SSL, SSL_BRIDGE, SSL_TCP, NNTP, RPCSVR, DNS, ADNS, SNMP, RTSP, DHCPRA, ANY, SIP_UDP, SIP_TCP, SIP_SSL, DNS_TCP, ADNS_TCP, HTTPSVR, HTTPCLIENT, NAT, HA, AAA, SINCTCP, VPN AFTP, MONITORS, SSLVPN UDP, SINCUDP, RIP, UDP CLNT, SASP, RPCCLNTS, ROUTE, AUDIT,

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
			STA HTTP, STA SSL, DNS RESOLVE, RDP, MYSQL, MSSQL, DIAMETER, SSL_DIAMETER, TFTP, ORACLE, ICA, RADIUS, RADIUSListener, SMPP, PPTP, GRE, SYSLOGTCP, SYSLOGUDP, FIX, SSL_FIX, QUIC, TFTP_DATA, USER_TCP, USER_SSL_TCP
__count	<Double>	Read-only	count parameter

Operations

(click to see Properties)

GET| GET (ALL)| COUNT

Some options that you can use for each operations:

- **Getting warnings in response:**NITRO allows you to get warnings in an operation by specifying the "warning" query parameter as "yes". For example, to get warnings while connecting to the Citrix ADC appliance, the URL is as follows:

  http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/login?warning=yes

  If any, the warnings are displayed in the response payload with the HTTP code "209 X-NITRO-WARNING".

- **Authenticated access for individual NITRO operations:**NITRO allows you to logon to the Citrix ADC appliance to perform individual operations. You can use this option instead of

creating a NITRO session (using the login object) and then using that session to perform all operations,

To do this, you must specify the username and password in the request header of the NITRO request as follows:

X-NITRO-USER:<username>

X-NITRO-PASS:<password>

Note: In such cases, make sure that the request header DOES not include the following:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue>

`*Note:`*

Mandatory parameters are marked in redand placeholder content is marked in <green>.

get

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/lsnrtspalgsession_datachannel_binding/sessionid_value<String>

Query-parameters:

filter

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/lsnrtspalgsession_datachannel_binding/sessionid_value<String>?**filter=property-name1:property-value1,property-name2:property-value2**

Use this query-parameter to get the filtered set of lsnrtspalgsession_datachannel_binding resources configured on Citrix ADC. Filtering can be done on any of the properties of the resource.

pagination

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/lsnrtspalgsession_datachannel_binding/sessionid_value<String>?**pagesize=#no;pageno=#no**

Use this query-parameter to get the lsnrtspalgsession_datachannel_binding resources in chunks.

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "lsnrtspalgsession_datachannel_binding": [ {

"channelip":<String_value>,

"sessionid":<String_value>,

"channelnatip":<String_value>,

"channelport":<Integer_value>,

"channelflags":<Double_value>,

"channeltimeout":<Double_value>,

"channelnatport":<Integer_value>,

"channelprotocol":<String_value>

}]}
```

get (all)

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/lsnrtspalgsession_datachannel_binding

Query-parameters:

bulkbindings

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/lsnrtspalgsession_datachannel_binding?**bulkbindings=yes**

NITRO allows you to fetch bindings in bulk.

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "lsnrtspalgsession_datachannel_binding": [ {

"channelip":<String_value>,

"sessionid":<String_value>,

"channelnatip":<String_value>,

"channelport":<Integer_value>,

"channelflags":<Double_value>,

"channeltimeout":<Double_value>,

"channelnatport":<Integer_value>,

"channelprotocol":<String_value>

}]}
```

count

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/lsnrtspalgsession_datachannel_binding/sessionid_value<String>?**count=yes**

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{"lsnrtspalgsession_datachannel_binding": [ { "__count": "#no"} ] }
```

lsnsession

Configuration for lsn session resource.

Properties

(click to see Operations)

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
nattype	<String>	Read-write	Type of sessions to be displayed. Default value: NAT44

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
			Possible values = NAT44, DS-Lite, NAT64
clientname	<String>	Read-write	Name of the LSN Client entity.
network	<String>	Read-write	IP address or network address of subscriber(s).
netmask	<String>	Read-write	Subnet mask for the IP address specified by the network parameter.
network6	<String>	Read-write	IPv6 address of the LSN subscriber or B4 device.
td	<Double>	Read-write	Traffic domain ID of the LSN client entity. Default value: 0 Minimum value = 0 Maximum value = 4094
natip	<String>	Read-write	Mapped NAT IP address used in LSN sessions.
nodeid	<Double>	Read-write	Unique number that identifies the cluster node. Minimum value = 0 Maximum value = 31

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
natport2	<Integer>	Read-write	Mapped NAT port used in the LSN sessions.
natprefix	<String>	Read-only	IPv6 address of the LSN subscriber(s) or subscriber network(B4-Device) on whose traffic the Citrix ADC perform Large Scale NAT.
subscrip	<String>	Read-only	The Source IP address.
subscrport	<Integer>	Read-only	The Source Port.
destip	<String>	Read-only	The Destination IP address.
destport	<Integer>	Read-only	The Destination Port.
natport	<Integer>	Read-only	The NAT Port.
transportprotocol	<String>	Read-only	The Transport Protocol for the session. Possible values = TCP, UDP, ICMP
sessionestdir	<String>	Read-only	The Session establishment direction, session was established for outbound or inbound packet. Possible values = OUT, IN
dsttd	<Double>	Read-only	.

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
srctd	<Double>	Read-only	.
ipv6address	<String>	Read-only	IPv6 address of v6 vserver.
__count	<Double>	Read-only	count parameter

Operations

(click to see Properties)

FLUSH| GET (ALL)| COUNT

Some options that you can use for each operations:

- **Getting warnings in response:**NITRO allows you to get warnings in an operation by specifying the "warning" query parameter as "yes". For example, to get warnings while connecting to the Citrix ADC appliance, the URL is as follows:

  http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/login?warning=yes

  If any, the warnings are displayed in the response payload with the HTTP code "209 X-NITRO-WARNING".

- **Authenticated access for individual NITRO operations:**NITRO allows you to logon to the Citrix ADC appliance to perform individual operations. You can use this option instead of creating a NITRO session (using the login object) and then using that session to perform all operations,

  To do this, you must specify the username and password in the request header of the NITRO request as follows:

  X-NITRO-USER:<username>

  X-NITRO-PASS:<password>

  Note: In such cases, make sure that the request header DOES not include the following:

  Cookie:NITRO_AUTH_TOKEN=<tokenvalue>

`*Note:` *

Mandatory parameters are marked in redand placeholder content is marked in <green>.

flush

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/lsnsession?**action=flush**

HTTP Method:POST

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Content-Type:application/json

Request Payload:

```
{"lsnsession":{

"nattype":<String_value>,

"clientname":<String_value>,

"network":<String_value>,

"netmask":<String_value>,

"network6":<String_value>,

"td":<Double_value>,

"natip":<String_value>,

"natport2":<Integer_value>,

"nodeid":<Double_value>

}}
```

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error

get (all)

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/lsnsession

Query-parameters:

args

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/lsnsession?**args=nattype:<String_value>,clientname:<String_value>,network:<String_value>,netmask:<String_valu**

Use this query-parameter to get lsnsession resources based on additional properties.

attrs

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/lsnsession?**attrs=property-name1,property-name2**

Use this query parameter to specify the resource details that you want to retrieve.

filter

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/lsnsession?**filter=property-name1:property-val1,property-name2:property-val2**

Use this query-parameter to get the filtered set of lsnsession resources configured on Citrix ADC. Filtering can be done on any of the properties of the resource.

view

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/lsnsession?**view=summary**

Note: By default, the retrieved results are displayed in detail view (?view=detail).

pagination

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/lsnsession?**pagesize=#no;pageno=#no**

Use this query-parameter to get the lsnsession resources in chunks.

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "lsnsession": [ {

nattype:<String_value>,clientname:<String_value>,network:<String_value>,ne
tmask:<String_value>,network6:<String_value>,td:<Double_value>,natip:<Stri
ng_value>,nodeid:<Double_value>"natprefix":<String_value>,

"subscrip":<String_value>,

"subscrport":<Integer_value>,

"destip":<String_value>,

"destport":<Integer_value>,

"natport":<Integer_value>,

"transportprotocol":<String_value>,

"sessionestdir":<String_value>,

"dsttd":<Double_value>,

"srctd":<Double_value>,

"ipv6address":<String_value>

}]}
```

count

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/lsnsession?**count=yes**

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "lsnsession": [ { "__count": "#no"} ] }
```

lsnsipalgcall

Configuration for LSN SIPALG call resource.

Properties

(click to see Operations)

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
callid	<String>	Read-write	Call ID for the SIP call.
nodeid	<Double>	Read-write	Unique number that identifies the cluster node. Minimum value = 0 Maximum value = 31

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
callflags	<Double>	Read-only	Flags for the call entry.
xlatip	<String>	Read-only	XLAT IP Address if its XLAT session.
callrefcount	<Double>	Read-only	Reference count for the call entry.
calltimer	<Double>	Read-only	Timer for the call entry.
__count	<Double>	Read-only	count parameter

Operations

(click to see Properties)

FLUSH| GET (ALL)| GET| COUNT

Some options that you can use for each operations:

- **Getting warnings in response:**NITRO allows you to get warnings in an operation by specifying the "warning" query parameter as "yes". For example, to get warnings while connecting to the Citrix ADC appliance, the URL is as follows:

  http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/login?warning=yes

  If any, the warnings are displayed in the response payload with the HTTP code "209 X-NITRO-WARNING".

- **Authenticated access for individual NITRO operations:**NITRO allows you to logon to the Citrix ADC appliance to perform individual operations. You can use this option instead of creating a NITRO session (using the login object) and then using that session to perform all operations,

  To do this, you must specify the username and password in the request header of the NITRO request as follows:

  X-NITRO-USER:<username>

  X-NITRO-PASS:<password>

Note: In such cases, make sure that the request header DOES not include the following:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue>

`*Note:` *

Mandatory parameters are marked in redand placeholder content is marked in <green>.

flush

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/lsnsipalgcall?**action=flush**

HTTP Method:POST

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Content-Type:application/json

Request Payload:

```
{"lsnsipalgcall":{

<b>"callid":<String_value>

</b>}}
```

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error

get (all)

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/lsnsipalgcall

Query-parameters:

args

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/lsnsipalgcall?**args=callid:<String_value>,nodeid:<Double_value>**

Use this query-parameter to get lsnsipalgcall resources based on additional properties.

attrs

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/lsnsipalgcall?**attrs=property-name1,property-name2**

Use this query parameter to specify the resource details that you want to retrieve.

filter

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/lsnsipalgcall?**filter=property-name1:property-val1,property-name2:property-val2**

Use this query-parameter to get the filtered set of lsnsipalgcall resources configured on Citrix ADC. Filtering can be done on any of the properties of the resource.

view

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/lsnsipalgcall?**view=summary**

Note: By default, the retrieved results are displayed in detail view (?view=detail).

pagination

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/lsnsipalgcall?**pagesize=#no;pageno=#no**

Use this query-parameter to get the lsnsipalgcall resources in chunks.

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "lsnsipalgcall": [ {

callid:<String_value>,nodeid:<Double_value>"callflags":<Double_value>,

"xlatip":<String_value>,

"callrefcount":<Double_value>,

"calltimer":<Double_value>

}]}
```

get

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/lsnsipalgcall/callid_value<String>

Query-parameters:

attrs

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/lsnsipalgcall/callid_value<String>?**attrs=property-name1,property-name2**

Use this query parameter to specify the resource details that you want to retrieve.

view

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/lsnsipalgcall/callid_value<String>?**view=summary**

Note: By default, the retrieved results are displayed in detail view (?view=detail).

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "lsnsipalgcall": [ {

callid:<String_value>,nodeid:<Double_value>"callflags":<Double_value>,

"xlatip":<String_value>,

"callrefcount":<Double_value>,

"calltimer":<Double_value>

}]}
```

count

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/lsnsipalgcall?**count=yes**

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "lsnsipalgcall": [ { "__count": "#no"} ] }
```

lsnsipalgcall_binding

Binding object which returns the resources bound to lsnsipalgcall.

Properties

(click to see Operations)

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
callid	<String>	Read-write	Call ID for the SIP call.
lsnsipalgcall_datachannel_binding	<lsnsipalgcall_datachannel_binding[]>	Read-only	datachannel that can be bound to lsnsipalgcall.
lsnsipalgcall_controlchannel_binding	<lsnsipalgcall_controlchannel_binding[]>	Read-only	controlchannel that can be bound to lsnsipalgcall.

Operations

(click to see Properties)

GET| GET (ALL)

Some options that you can use for each operations:

- **Getting warnings in response:**NITRO allows you to get warnings in an operation by specifying the "warning" query parameter as "yes". For example, to get warnings while connecting to the Citrix ADC appliance, the URL is as follows:

  http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/login?warning=yes

  If any, the warnings are displayed in the response payload with the HTTP code "209 X-NITRO-WARNING".

- **Authenticated access for individual NITRO operations:**NITRO allows you to logon to the Citrix ADC appliance to perform individual operations. You can use this option instead of creating a NITRO session (using the login object) and then using that session to perform all operations,

  To do this, you must specify the username and password in the request header of the NITRO request as follows:

  X-NITRO-USER:<username>

  X-NITRO-PASS:<password>

  Note: In such cases, make sure that the request header DOES not include the following:

  Cookie:NITRO_AUTH_TOKEN=<tokenvalue>

`*Note:`*

Mandatory parameters are marked in redand placeholder content is marked in <green>.

get

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/lsnsipalgcall_binding/callid_value<String>

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "lsnsipalgcall_binding": [ {
```

```
"callid":<String_value>,

"lsnsipalgcall_controlchannel_binding":<lsnsipalgcall_controlchannel_binding[]_value>,

"lsnsipalgcall_datachannel_binding":<lsnsipalgcall_datachannel_binding[]_value>

}]}
```

get (all)

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/lsnsipalgcall_binding

Query-parameters:

bulkbindings

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/lsnsipalgcall_binding?**bulkbindings=yes**

NITRO allows you to fetch bindings in bulk.

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "lsnsipalgcall_binding": [ {

"callid":<String_value>,
```

```
"lsnsipalgcall_controlchannel_binding":<lsnsipalgcall_controlchannel_bindi
ng[]_value>,

"lsnsipalgcall_datachannel_binding":<lsnsipalgcall_datachannel_binding[]_v
alue>

}]}
```

lsnsipalgcall_controlchannel_binding

Binding object showing the controlchannel that can be bound to lsnsipalgcall.

Properties

(click to see Operations)

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
channelip	<String>	Read-write	IP address of the channel.
callid	<String>	Read-write	Call ID for the SIP call.
channelnatip	<String>	Read-only	Natted IP address of the channel.
channelport	<Integer>	Read-only	port for the channel.
channelflags	<Double>	Read-only	Flags for the call entry.
channeltimeout	<Double>	Read-only	Timeout for the channel.
channelnatport	<Integer>	Read-only	Natted port for the channel.
channelprotocol	<String>	Read-only	Channel transport protocol.

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
			Possible values = HTTP, FTP, TCP, UDP, SSL, SSL_BRIDGE, SSL_TCP, NNTP, RPCSVR, DNS, ADNS, SNMP, RTSP, DHCPRA, ANY, SIP_UDP, SIP_TCP, SIP_SSL, DNS_TCP, ADNS_TCP, HTTPSVR, HTTPCLIENT, NAT, HA, AAA, SINCTCP, VPN AFTP, MONITORS, SSLVPN UDP, SINCUDP, RIP, UDP CLNT, SASP, RPCCLNTS, ROUTE, AUDIT, STA HTTP, STA SSL, DNS RESOLVE, RDP, MYSQL, MSSQL, DIAMETER, SSL_DIAMETER, TFTP, ORACLE, ICA, RADIUS, RADIUSListener, SMPP, PPTP, GRE, SYSLOGTCP, SYSLOGUDP, FIX, SSL_FIX, QUIC, TFTP_DATA,

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
			USER_TCP, USER_SSL_TCP
__count	<Double>	Read-only	count parameter

Operations

(click to see Properties)

GET| GET (ALL)| COUNT

Some options that you can use for each operations:

- **Getting warnings in response:**NITRO allows you to get warnings in an operation by specifying the "warning" query parameter as "yes". For example, to get warnings while connecting to the Citrix ADC appliance, the URL is as follows:

  http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/login?warning=yes

  If any, the warnings are displayed in the response payload with the HTTP code "209 X-NITRO-WARNING".

- **Authenticated access for individual NITRO operations:**NITRO allows you to logon to the Citrix ADC appliance to perform individual operations. You can use this option instead of creating a NITRO session (using the login object) and then using that session to perform all operations,

  To do this, you must specify the username and password in the request header of the NITRO request as follows:

  X-NITRO-USER:<username>

  X-NITRO-PASS:<password>

  Note: In such cases, make sure that the request header DOES not include the following:

  Cookie:NITRO_AUTH_TOKEN=<tokenvalue>

`*Note:`*

Mandatory parameters are marked in redand placeholder content is marked in <green>.

get

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/lsnsipalgcall_controlchannel_binding/callid_value<String>

Query-parameters:

filter

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/lsnsipalgcall_controlchannel_binding/callid_value<String>?**filter=property-name1:property-value1,property-name2:property-value2**

Use this query-parameter to get the filtered set of lsnsipalgcall_controlchannel_binding resources configured on Citrix ADC. Filtering can be done on any of the properties of the resource.

pagination

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/lsnsipalgcall_controlchannel_binding/callid_value<String>?**pagesize=#no;pageno=#no**

Use this query-parameter to get the lsnsipalgcall_controlchannel_binding resources in chunks.

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "lsnsipalgcall_controlchannel_binding": [ {

"channelip":<String_value>,

"callid":<String_value>,
```

```
"channelnatip":<String_value>,

"channelport":<Integer_value>,

"channelflags":<Double_value>,

"channeltimeout":<Double_value>,

"channelnatport":<Integer_value>,

"channelprotocol":<String_value>

}]}
```

get (all)

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/lsnsipalgcall_controlchannel_binding

Query-parameters:

bulkbindings

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/lsnsipalgcall_controlchannel_binding?**bulkbindings=yes**

NITRO allows you to fetch bindings in bulk.

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "lsnsipalgcall_controlchannel_binding": [ {

"channelip":<String_value>,

"callid":<String_value>,

"channelnatip":<String_value>,

"channelport":<Integer_value>,

"channelflags":<Double_value>,

"channeltimeout":<Double_value>,

"channelnatport":<Integer_value>,

"channelprotocol":<String_value>

}]}
```

count

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/lsnsipalgcall_controlchannel_binding/callid_value<String>?**count=yes**

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{"lsnsipalgcall_controlchannel_binding": [ { "__count": "#no"} ] }
```

lsnsipalgcall_datachannel_binding

Binding object showing the datachannel that can be bound to lsnsipalgcall.

Properties

(click to see Operations)

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
channelip	<String>	Read-write	IP address of the channel.
callid	<String>	Read-write	Call ID for the SIP call.
channelnatip	<String>	Read-only	Natted IP address of the channel.
channelport	<Integer>	Read-only	port for the channel.
channelflags	<Double>	Read-only	Flags for the call entry.
channeltimeout	<Double>	Read-only	Timeout for the channel.
channelnatport	<Integer>	Read-only	Natted port for the channel.
channelprotocol	<String>	Read-only	Channel transport protocol. Possible values = HTTP, FTP, TCP, UDP, SSL, SSL_BRIDGE, SSL_TCP, NNTP,

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
			RPCSVR, DNS, ADNS, SNMP, RTSP, DHCPRA, ANY, SIP_UDP, SIP_TCP, SIP_SSL, DNS_TCP, ADNS_TCP, HTTPSVR, HTTPCLIENT, NAT, HA, AAA, SINCTCP, VPN AFTP, MONITORS, SSLVPN UDP, SINCUDP, RIP, UDP CLNT, SASP, RPCCLNTS, ROUTE, AUDIT, STA HTTP, STA SSL, DNS RESOLVE, RDP, MYSQL, MSSQL, DIAMETER, SSL_DIAMETER, TFTP, ORACLE, ICA, RADIUS, RADIUSListener, SMPP, PPTP, GRE, SYSLOGTCP, SYSLOGUDP, FIX, SSL_FIX, QUIC, TFTP_DATA, USER_TCP, USER_SSL_TCP
__count	<Double>	Read-only	count parameter

Operations

(click to see Properties)

GET| GET (ALL)| COUNT

Some options that you can use for each operations:

- **Getting warnings in response:**NITRO allows you to get warnings in an operation by specifying the "warning" query parameter as "yes". For example, to get warnings while connecting to the Citrix ADC appliance, the URL is as follows:

  http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/login?warning=yes

  If any, the warnings are displayed in the response payload with the HTTP code "209 X-NITRO-WARNING".

- **Authenticated access for individual NITRO operations:**NITRO allows you to logon to the Citrix ADC appliance to perform individual operations. You can use this option instead of creating a NITRO session (using the login object) and then using that session to perform all operations,

  To do this, you must specify the username and password in the request header of the NITRO request as follows:

  X-NITRO-USER:<username>

  X-NITRO-PASS:<password>

  Note: In such cases, make sure that the request header DOES not include the following:

  Cookie:NITRO_AUTH_TOKEN=<tokenvalue>

`*Note:`*

Mandatory parameters are marked in redand placeholder content is marked in <green>.

get

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/lsnsipalgcall_datachannel_binding/callid_value<String>

Query-parameters:

filter

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/lsnsipalgcall_datachannel_binding/callid_value<String>?**filter=property-name1:property-value1,property-name2:property-value2**

Use this query-parameter to get the filtered set of lsnsipalgcall_datachannel_binding resources configured on Citrix ADC. Filtering can be done on any of the properties of the resource.

pagination

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/lsnsipalgcall_datachannel_binding/callid_value<String>?**pagesize=#no;pageno=#no**

Use this query-parameter to get the lsnsipalgcall_datachannel_binding resources in chunks.

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "lsnsipalgcall_datachannel_binding": [ {

"channelip":<String_value>,

"callid":<String_value>,

"channelnatip":<String_value>,

"channelport":<Integer_value>,

"channelflags":<Double_value>,

"channeltimeout":<Double_value>,
```

```
"channelnatport":<Integer_value>,

"channelprotocol":<String_value>

}]}
```

get (all)

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/lsnsipalgcall_datachannel_binding

Query-parameters:

bulkbindings

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/lsnsipalgcall_datachannel_binding?**bulkbindings=yes**

NITRO allows you to fetch bindings in bulk.

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "lsnsipalgcall_datachannel_binding": [ {

"channelip":<String_value>,

"callid":<String_value>,
```

```
"channelnatip":<String_value>,

"channelport":<Integer_value>,

"channelflags":<Double_value>,

"channeltimeout":<Double_value>,

"channelnatport":<Integer_value>,

"channelprotocol":<String_value>

}]}
```

count

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/lsnsipalgcall_datachannel_binding/callid_value<String>?**count=yes**

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{"lsnsipalgcall_datachannel_binding": [ { "__count": "#no"} ] }
```

lsnsipalgprofile

Configuration for LSN SIPALG Profile resource.

Properties

(click to see Operations)

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
sipalgprofilename	<String>	Read-write	The name of the SIPALG Profile. Minimum length = 1 Maximum length = 127
datasessionidletimeout	<Double>	Read-write	Idle timeout for the data channel sessions in seconds. Default value: 120
sipsessiontimeout	<Double>	Read-write	SIP control channel session timeout in seconds. Default value: 600
registrationtimeout	<Double>	Read-write	SIP registration timeout in seconds. Default value: 60
sipsrcportrange	<String>	Read-write	Source port range for SIP_UDP and SIP_TCP.
sipdstportrange	<String>	Read-write	Destination port range for SIP_UDP and SIP_TCP.
openregisterpinhole	<String>	Read-write	ENABLE/DISABLE RegisterPinhole creation.

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
			Default value: ENABLED Possible values = ENABLED, DISABLED
opencontactpinhole	<String>	Read-write	ENABLE/DISABLE ContactPinhole creation. Default value: ENABLED Possible values = ENABLED, DISABLED
openviapinhole	<String>	Read-write	ENABLE/DISABLE ViaPinhole creation. Default value: ENABLED Possible values = ENABLED, DISABLED
openrecordroutepinhole	<String>	Read-write	ENABLE/DISABLE RecordRoutePinhole creation. Default value: ENABLED Possible values = ENABLED, DISABLED
siptransportprotocol	<String>	Read-write	SIP ALG Profile transport protocol type.

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
			Possible values = TCP, UDP
openroutepinhole	<String>	Read-write	ENABLE/DISABLE RoutePinhole creation. Default value: ENABLED Possible values = ENABLED, DISABLED
rport	<String>	Read-write	ENABLE/DISABLE rport. Default value: ENABLED Possible values = ENABLED, DISABLED
__count	<Double>	Read-only	count parameter

Operations

(click to see Properties)

ADD| UPDATE| UNSET| DELETE| GET (ALL)| GET| COUNT

Some options that you can use for each operations:

- **Getting warnings in response:**NITRO allows you to get warnings in an operation by specifying the "warning" query parameter as "yes". For example, to get warnings while connecting to the Citrix ADC appliance, the URL is as follows:

  http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/login?warning=yes

  If any, the warnings are displayed in the response payload with the HTTP code "209 X-NITRO-WARNING".

- **Authenticated access for individual NITRO operations:**NITRO allows you to logon to the Citrix ADC appliance to perform individual operations. You can use this option instead of creating a NITRO session (using the login object) and then using that session to perform all operations,

  To do this, you must specify the username and password in the request header of the NITRO request as follows:

  X-NITRO-USER:<username>

  X-NITRO-PASS:<password>

  Note: In such cases, make sure that the request header DOES not include the following:

  Cookie:NITRO_AUTH_TOKEN=<tokenvalue>

`*Note:`*

Mandatory parameters are marked in redand placeholder content is marked in <green>.

add

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/lsnsipalgprofile

HTTP Method:POST

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Content-Type:application/json

Request Payload:

```
{"lsnsipalgprofile":{

<b>"sipalgprofilename":<String_value>,

</b>"datasessionidletimeout":<Double_value>,

"sipsessiontimeout":<Double_value>,

"registrationtimeout":<Double_value>,

"sipsrcportrange":<String_value>,
```

```
"sipdstportrange":<String_value>,

"openregisterpinhole":<String_value>,

"opencontactpinhole":<String_value>,

"openviapinhole":<String_value>,

"openrecordroutepinhole":<String_value>,

<b>"siptransportprotocol":<String_value>,

</b>"openroutepinhole":<String_value>,

"rport":<String_value>

}}
```

Response:

HTTP Status Code on Success: 201 Created HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error

update

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/lsnsipalgprofile

HTTP Method:PUT

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Content-Type:application/json

Request Payload:

```
{"lsnsipalgprofile":{

<b>"sipalgprofilename":<String_value>,
```

```
</b>"datasessionidletimeout":<Double_value>,

"sipsessiontimeout":<Double_value>,

"registrationtimeout":<Double_value>,

"sipsrcportrange":<String_value>,

"sipdstportrange":<String_value>,

"openregisterpinhole":<String_value>,

"opencontactpinhole":<String_value>,

"openviapinhole":<String_value>,

"openrecordroutepinhole":<String_value>,

"siptransportprotocol":<String_value>,

"openroutepinhole":<String_value>,

"rport":<String_value>

}}
```

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error

unset

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/lsnsipalgprofile?**action=unset**

HTTP Method:POST

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Content-Type:application/json

Request Payload:

```
{"lsnsipalgprofile":{

<b>"sipalgprofilename":<String_value>,

</b>"datasessionidletimeout":true,

"sipsessiontimeout":true,

"registrationtimeout":true,

"sipsrcportrange":true,

"sipdstportrange":true,

"openregisterpinhole":true,

"opencontactpinhole":true,

"openviapinhole":true,

"openrecordroutepinhole":true,

"siptransportprotocol":true,

"openroutepinhole":true,

"rport":true

}}
```

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error

delete

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/lsnsipalgprofile/sipalgprofilename_value<String>

HTTP Method:DELETE

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue>

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error

get (all)

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/lsnsipalgprofile

Query-parameters:

attrs

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/lsnsipalgprofile?**attrs=property-name1,property-name2**

Use this query parameter to specify the resource details that you want to retrieve.

filter

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/lsnsipalgprofile?**filter=property-name1:property-val1,property-name2:property-val2**

Use this query-parameter to get the filtered set of lsnsipalgprofile resources configured on Citrix ADC. Filtering can be done on any of the properties of the resource.

view

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/lsnsipalgprofile?**view=summary**

Note: By default, the retrieved results are displayed in detail view (?view=detail).

pagination

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/lsnsipalgprofile?**pagesize=#no;pageno=#no**

Use this query-parameter to get the lsnsipalgprofile resources in chunks.

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "lsnsipalgprofile": [ {

"sipalgprofilename":<String_value>,

"datasessionidletimeout":<Double_value>,

"sipsessiontimeout":<Double_value>,

"registrationtimeout":<Double_value>,

"sipsrcportrange":<String_value>,

"sipdstportrange":<String_value>,

"openregisterpinhole":<String_value>,

"opencontactpinhole":<String_value>,

"openviapinhole":<String_value>,

"openrecordroutepinhole":<String_value>,

"siptransportprotocol":<String_value>,
```

```
"openroutepinhole":<String_value>,

"rport":<String_value>

}]}
```

get

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/lsnsipalgprofile/sipalgprofilename_value<String>

Query-parameters:

attrs

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/lsnsipalgprofile/sipalgprofilename_value<String>?**attrs=property-name1,property-name2**

Use this query parameter to specify the resource details that you want to retrieve.

view

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/lsnsipalgprofile/sipalgprofilename_value<String>?**view=summary**

Note: By default, the retrieved results are displayed in detail view (?view=detail).

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "lsnsipalgprofile": [ {
"sipalgprofilename":<String_value>,
"datasessionidletimeout":<Double_value>,
"sipsessiontimeout":<Double_value>,
"registrationtimeout":<Double_value>,
"sipsrcportrange":<String_value>,
"sipdstportrange":<String_value>,
"openregisterpinhole":<String_value>,
"opencontactpinhole":<String_value>,
"openviapinhole":<String_value>,
"openrecordroutepinhole":<String_value>,
"siptransportprotocol":<String_value>,
"openroutepinhole":<String_value>,
"rport":<String_value>
}]}
```

count

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/lsnsipalgprofile?**count=yes**

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "lsnsipalgprofile": [ { "__count": "#no"} ] }
```

lsnstatic

Configuration for static mapping resource.

Properties

(click to see Operations)

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
name	<String>	Read-write	Name for the LSN static mapping entry. Must begin with an ASCII alphanumeric or underscore (_) character, and must contain only ASCII alphanumeric, underscore, hash (#), period (.), space, colon (:), at (@), equals (=), and hyphen (-) characters. Cannot be changed after the LSN group is created. The following requirement applies only to the Citrix

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
			ADC CLI: If the name includes one or more spaces, enclose the name in double or single quotation marks (for example, "lsn static1" or 'lsn static1'). Minimum length = 1 Maximum length = 127
transportprotocol	<String>	Read-write	Protocol for the LSN mapping entry. Possible values = TCP, UDP, ICMP, ALL
subscrip	<String>	Read-write	IPv4(NAT44 ; DS-Lite)/IPv6(NAT64) address of an LSN subscriber for the LSN static mapping entry.
subscrport	<Integer>	Read-write	Port of the LSN subscriber for the LSN mapping entry. * represents all ports being used. Used in case of static wildcard. Minimum value = 0 Maximum value = 65535 Range 1 - 65535

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
			* in CLI is represented as 65535 in NITRO API
network6	<String>	Read-write	B4 address in DS-Lite setup. Minimum length = 1
td	<Double>	Read-write	ID of the traffic domain to which the subscriber belongs. If you do not specify an ID, the subscriber is assumed to be a part of the default traffic domain. Default value: 0 Minimum value = 0 Maximum value = 4094
natip	<String>	Read-write	IPv4 address, already existing on the Citrix ADC as type LSN, to be used as NAT IP address for this mapping entry.
natport	<Integer>	Read-write	NAT port for this LSN mapping entry. * represents all ports being used. Used in case of static wildcard.

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
			Minimum value = 0 Maximum value = 65535 Range 1 - 65535 * in CLI is represented as 65535 in NITRO API
destip	<String>	Read-write	Destination IP address for the LSN mapping entry.
dsttd	<Double>	Read-write	ID of the traffic domain through which the destination IP address for this LSN mapping entry is reachable from the Citrix ADC. If you do not specify an ID, the destination IP address is assumed to be reachable through the default traffic domain, which has an ID of 0. Default value: 0 Minimum value = 0 Maximum value = 4094

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
nattype	<String>	Read-write	Type of sessions to be displayed. Possible values = NAT44, DS-Lite, NAT64
status	<String>	Read-only	The status of the Mapping. Status could be Inactive, if mapping addition failed due to already existing dynamic/ static mapping, port allocation failure. Possible values = ACTIVE, INACTIVE
__count	<Double>	Read-only	count parameter

Operations

(click to see Properties)

ADD| DELETE| GET (ALL)| GET| COUNT

Some options that you can use for each operations:

- **Getting warnings in response:**NITRO allows you to get warnings in an operation by specifying the "warning" query parameter as "yes". For example, to get warnings while connecting to the Citrix ADC appliance, the URL is as follows:

  http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/login?warning=yes

  If any, the warnings are displayed in the response payload with the HTTP code "209 X-NITRO-WARNING".
- **Authenticated access for individual NITRO operations:**NITRO allows you to logon to the Citrix ADC appliance to perform individual operations. You can use this option instead of creating a NITRO session (using the login object) and then using that session to perform all operations,

To do this, you must specify the username and password in the request header of the NITRO request as follows:

X-NITRO-USER:<username>

X-NITRO-PASS:<password>

Note: In such cases, make sure that the request header DOES not include the following:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue>

`*Note:`*

Mandatory parameters are marked in redand placeholder content is marked in <green>.

add

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/lsnstatic

HTTP Method:POST

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Content-Type:application/json

Request Payload:

```
{"lsnstatic":{

<b>"name":<String_value>,

</b><b>"transportprotocol":<String_value>,

</b><b>"subscrip":<String_value>,

</b><b>"subscrport":<Integer_value>,

</b>"network6":<String_value>,

"td":<Double_value>,

"natip":<String_value>,
```

```
"natport":<Integer_value>,

"destip":<String_value>,

"dsttd":<Double_value>

}}
```

Response:

HTTP Status Code on Success: 201 Created HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error

delete

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/lsnstatic/name_value<String>

HTTP Method:DELETE

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue>

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error

get (all)

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/lsnstatic

Query-parameters:

args

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/lsnstatic?
args=name:<String_value>,nattype:<String_value>

Use this query-parameter to get lsnstatic resources based on additional properties.

attrs

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/lsnstatic?**attrs=property-name1,property-name2**

Use this query parameter to specify the resource details that you want to retrieve.

filter

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/lsnstatic?**filter=property-name1:property-val1,property-name2:property-val2**

Use this query-parameter to get the filtered set of lsnstatic resources configured on Citrix ADC. Filtering can be done on any of the properties of the resource.

view

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/lsnstatic?**view=summary**

Note: By default, the retrieved results are displayed in detail view (?view=detail).

pagination

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/lsnstatic?**pagesize=#no;pageno=#no**

Use this query-parameter to get the lsnstatic resources in chunks.

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "lsnstatic": [ {
```

```
name:<String_value>,nattype:<String_value>"transportprotocol":<String_valu
e>,

"subscrip":<String_value>,

"subscrport":<Integer_value>,

"network6":<String_value>,

"natip":<String_value>,

"natport":<Integer_value>,

"td":<Double_value>,

"destip":<String_value>,

"dsttd":<Double_value>,

"status":<String_value>

}]}
```

get

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/lsnstatic/name_value<String>

Query-parameters:

attrs

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/lsnstatic/name_value<String>?**attrs=property-name1,property-name2**

Use this query parameter to specify the resource details that you want to retrieve.

view

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/lsnstatic/name_value<String>?**view=summary**

Note: By default, the retrieved results are displayed in detail view (?view=detail).

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "lsnstatic": [ {

name:<String_value>,nattype:<String_value>"transportprotocol":<String_valu
e>,

"subscrip":<String_value>,

"subscrport":<Integer_value>,

"network6":<String_value>,

"natip":<String_value>,

"natport":<Integer_value>,

"td":<Double_value>,

"destip":<String_value>,

"dsttd":<Double_value>,

"status":<String_value>

}]}
```

count

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/lsnstatic?**count=yes**

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "lsnstatic": [ { "__count": "#no"} ] }
```

lsntransportprofile

Configuration for LSN Transport Profile resource.

Properties

(click to see Operations)

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
transportprofilename	<String>	Read-write	Name for the LSN transport profile. Must begin with an ASCII alphanumeric or underscore (_) character, and must contain only ASCII alphanumeric, underscore, hash

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
			(#), period (.), space, colon (:), at (@), equals (=), and hyphen (-) characters. Cannot be changed after the LSN transport profile is created. The following requirement applies only to the Citrix ADC CLI: If the name includes one or more spaces, enclose the name in double or single quotation marks (for example, "lsn transport profile1" or 'lsn transport profile1'). Minimum length = 1 Maximum length = 127
transportprotocol	<String>	Read-write	Protocol for which to set the LSN transport profile parameters. Possible values = TCP, UDP, ICMP
sessiontimeout	<Double>	Read-write	Timeout, in seconds, for an idle LSN session. If an LSN session is idle for a time that exceeds this value, the Citrix ADC removes the session.

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
			This timeout does not apply for a TCP LSN session when a FIN or RST message is received from either of the endpoints. . Default value: 120 Minimum value = 60
finrsttimeout	<Double>	Read-write	Timeout, in seconds, for a TCP LSN session after a FIN or RST message is received from one of the endpoints. If a TCP LSN session is idle (after the Citrix ADC receives a FIN or RST message) for a time that exceeds this value, the Citrix ADC ADC removes the session. Since the LSN feature of the Citrix ADC does not maintain state information of any TCP LSN sessions, this timeout accommodates the transmission of the FIN or RST, and

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
			ACK messages from the other endpoint so that both endpoints can properly close the connection. Default value: 30
stuntimeout	<Double>	Read-write	STUN protocol timeout. Default value: 600 Minimum value = 120 Maximum value = 1200
synidletimeout	<Double>	Read-write	SYN Idle timeout. Default value: 60 Minimum value = 30 Maximum value = 120
portquota	<Double>	Read-write	Maximum number of LSN NAT ports to be used at a time by each subscriber for the specified protocol. For example, each subscriber can be limited to a maximum of 500 TCP NAT ports. When the LSN NAT mappings for a subscriber reach the limit, the Citrix ADC

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
			does not allocate additional NAT ports for that subscriber. Default value: 0 Minimum value = 0 Maximum value = 65535
sessionquota	<Double>	Read-write	Maximum number of concurrent LSN sessions allowed for each subscriber for the specified protocol. When the number of LSN sessions reaches the limit for a subscriber, the Citrix ADC does not allow the subscriber to open additional sessions. Default value: 0 Minimum value = 0 Maximum value = 65535
groupsessionlimit	<Double>	Read-write	Maximum number of concurrent LSN sessions(for the specified protocol) allowed for all subscriber of a group to which this profile has bound. This limit will get split across

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
			the Citrix ADCs packet engines and rounded down. When the number of LSN sessions reaches the limit for a group in packet engine, the Citrix ADC does not allow the subscriber of that group to open additional sessions through that packet engine. Default value: 0
portpreserveparity	<String>	Read-write	Enable port parity between a subscriber port and its mapped LSN NAT port. For example, if a subscriber initiates a connection from an odd numbered port, the Citrix ADC allocates an odd numbered LSN NAT port for this connection. You must set this parameter for proper functioning of protocols that require the source port to be even or odd numbered, for example, in peer-to-peer applications

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
			that use RTP or RTCP protocol. Default value: DISABLED Possible values = ENABLED, DISABLED
portpreserverange	<String>	Read-write	If a subscriber initiates a connection from a well-known port (0-1023), allocate a NAT port from the well-known port range (0-1023) for this connection. For example, if a subscriber initiates a connection from port 80, the Citrix ADC can allocate port 100 as the NAT port for this connection. This parameter applies to dynamic NAT without port block allocation. It also applies to Deterministic NAT if the range of ports allocated includes well-known ports. When all the well-known ports of all the available NAT IP addresses are used

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
			in different subscriber's connections (LSN sessions), and a subscriber initiates a connection from a well-known port, the Citrix ADC drops this connection. Default value: DISABLED Possible values = ENABLED, DISABLED
syncheck	<String>	Read-write	Silently drop any non-SYN packets for connections for which there is no LSN-NAT session present on the Citrix ADC. If you disable this parameter, the Citrix ADC accepts any non-SYN packets and creates a new LSN session entry for this connection. Following are some reasons for the Citrix ADC to receive such packets:

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
			* LSN session for a connection existed but the Citrix ADC removed this session because the LSN session was idle for a time that exceeded the configured session timeout. * Such packets can be a part of a DoS attack. Default value: ENABLED Possible values = ENABLED, DISABLED
__count	<Double>	Read-only	count parameter

Operations

(click to see Properties)

ADD| DELETE| UPDATE| UNSET| GET (ALL)| GET| COUNT

Some options that you can use for each operations:

- **Getting warnings in response:**NITRO allows you to get warnings in an operation by specifying the "warning" query parameter as "yes". For example, to get warnings while connecting to the Citrix ADC appliance, the URL is as follows:

  http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/login?warning=yes

  If any, the warnings are displayed in the response payload with the HTTP code "209 X-NITRO-WARNING".

- **Authenticated access for individual NITRO operations:**NITRO allows you to logon to the Citrix ADC appliance to perform individual operations. You can use this option instead of creating a NITRO session (using the login object) and then using that session to perform all operations,

  To do this, you must specify the username and password in the request header of the NITRO request as follows:

  X-NITRO-USER:<username>

  X-NITRO-PASS:<password>

  Note: In such cases, make sure that the request header DOES not include the following:

  Cookie:NITRO_AUTH_TOKEN=<tokenvalue>

`*Note:`*

Mandatory parameters are marked in redand placeholder content is marked in <green>.

add

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/lsntransportprofile

HTTP Method:POST

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Content-Type:application/json

Request Payload:

```
{"lsntransportprofile":{

<b>"transportprofilename":<String_value>,

</b><b>"transportprotocol":<String_value>,

</b>"sessiontimeout":<Double_value>,

"finrsttimeout":<Double_value>,

"stuntimeout":<Double_value>,
```

```
"synidletimeout":<Double_value>,

"portquota":<Double_value>,

"sessionquota":<Double_value>,

"groupsessionlimit":<Double_value>,

"portpreserveparity":<String_value>,

"portpreserverange":<String_value>,

"syncheck":<String_value>

}}
```

Response:

HTTP Status Code on Success: 201 Created HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error

delete

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/lsntransportprofile/transportprofilename_value<String>

HTTP Method:DELETE

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue>

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error

update

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/lsntransportprofile

HTTP Method:PUT

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Content-Type:application/json

Request Payload:

```
{"lsntransportprofile":{

<b>"transportprofilename":<String_value>,

</b>"sessiontimeout":<Double_value>,

"finrsttimeout":<Double_value>,

"stuntimeout":<Double_value>,

"synidletimeout":<Double_value>,

"portquota":<Double_value>,

"sessionquota":<Double_value>,

"groupsessionlimit":<Double_value>,

"portpreserveparity":<String_value>,

"portpreserverange":<String_value>,

"syncheck":<String_value>

}}
```

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error

unset

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/lsntransportprofile?**action=unset**

HTTP Method:POST

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Content-Type:application/json

Request Payload:

```
{"lsntransportprofile":{

<b>"transportprofilename":<String_value>,

</b>"sessiontimeout":true,

"finrsttimeout":true,

"stuntimeout":true,

"synidletimeout":true,

"portquota":true,

"sessionquota":true,

"groupsessionlimit":true,

"portpreserveparity":true,

"portpreserverange":true,

"syncheck":true

}}
```

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error

get (all)

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/lsntransportprofile

Query-parameters:

attrs

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/lsntransportprofile?**attrs=property-name1,property-name2**

Use this query parameter to specify the resource details that you want to retrieve.

filter

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/lsntransportprofile?**filter=property-name1:property-val1,property-name2:property-val2**

Use this query-parameter to get the filtered set of lsntransportprofile resources configured on Citrix ADC. Filtering can be done on any of the properties of the resource.

view

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/lsntransportprofile?**view=summary**

Note: By default, the retrieved results are displayed in detail view (?view=detail).

pagination

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/lsntransportprofile?**pagesize=#no;pageno=#no**

Use this query-parameter to get the lsntransportprofile resources in chunks.

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "lsntransportprofile": [ {

"transportprofilename":<String_value>,

"transportprotocol":<String_value>,

"sessiontimeout":<Double_value>,

"finrsttimeout":<Double_value>,

"stuntimeout":<Double_value>,

"synidletimeout":<Double_value>,

"portquota":<Double_value>,

"sessionquota":<Double_value>,

"groupsessionlimit":<Double_value>,

"portpreserveparity":<String_value>,

"portpreserverange":<String_value>,

"syncheck":<String_value>

}]}
```

get

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/lsntransportprofile/transportprofilename_value<String>

Query-parameters:

attrs

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/lsntransportprofile/transportprofilename_value<String>?**attrs=property-name1,property-name2**

Use this query parameter to specify the resource details that you want to retrieve.

view

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/lsntransportprofile/transportprofilename_value<String>?**view=summary**

Note: By default, the retrieved results are displayed in detail view (?view=detail).

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "lsntransportprofile": [ {

"transportprofilename":<String_value>,

"transportprotocol":<String_value>,

"sessiontimeout":<Double_value>,

"finrsttimeout":<Double_value>,

"stuntimeout":<Double_value>,
```

```
"synidletimeout":<Double_value>,

"portquota":<Double_value>,

"sessionquota":<Double_value>,

"groupsessionlimit":<Double_value>,

"portpreserveparity":<String_value>,

"portpreserverange":<String_value>,

"syncheck":<String_value>

}]}
```

count

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/lsntransportprofile?**count=yes**

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "lsntransportprofile": [ { "__count": "#no"} ] }
```

Network

Network

Networking commands.

NAME	DESCRIPTION
Interface	Configuration for interface
appalgparam	Configuration for AppAlg Param
arp	Configuration for arp
arpparam	Configuration for Global arp parameters
bridgegroup	Configuration for bridge group
bridgegroup_binding	Binding object showing the resources that can be bound to bridgegroup
bridgegroup_nsip6_binding	Binding object showing the nsip6 that can be bound to bridgegroup
bridgegroup_nsip_binding	Binding object showing the nsip that can be bound to bridgegroup
bridgegroup_vlan_binding	Binding object showing the vlan that can be bound to bridgegroup
bridgetable	Configuration for bridge table entry
channel	Configuration for channel
channel_binding	Binding object showing the resources that can be bound to channel
channel_interface_binding	Binding object showing the interface that can be bound to channel
ci	Configuration for "CI"
fis	Configuration for "FIS"

NAME	DESCRIPTION
fis_binding	Binding object showing the resources that can be bound to fis
fis_channel_binding	Binding object showing the channel that can be bound to fis
fis_interface_binding	Binding object showing the interface that can be bound to fis
forwardingsession	Configuration for session forward
inat	Configuration for inbound nat
inatparam	Configuration for INAT parameter
interfacepair	Configuration for "Interface Pair"
ip6tunnel	Configuration for ip6 Tunnel
ip6tunnelparam	Configuration for ip6 tunnel parameter
ipset	Configuration for network ipset
ipset_binding	Binding object showing the resources that can be bound to ipset
ipset_nsip6_binding	Binding object showing the nsip6 that can be bound to ipset
ipset_nsip_binding	Binding object showing the nsip that can be bound to ipset
iptunnel	Configuration for ip Tunnel
iptunnelparam	Configuration for ip tunnel parameter
ipv6	Configuration for ip v6
l2param	Configuration for Layer 2 related parameter
l3param	Configuration for Layer 3 related parameter

NAME	DESCRIPTION
l4param	Configuration for Layer 4 related parameter
lacp	Configuration for Link aggregation control protocol
linkset	Configuration for link set
linkset_binding	Binding object showing the resources that can be bound to linkset
linkset_channel_binding	Binding object showing the channel that can be bound to linkset
linkset_interface_binding	Binding object showing the interface that can be bound to linkset
mapbmr	Configuration for MAP-T Basic Mapping rule
mapbmr_binding	Binding object showing the resources that can be bound to mapbmr
mapbmr_bmrv4network_binding	Binding object showing the bmrv4network that can be bound to mapbmr
mapdmr	Configuration for MAP-T Default Mapping rule
mapdomain	Configuration for MAP-T Map Domain
mapdomain_binding	Binding object showing the resources that can be bound to mapdomain
mapdomain_mapbmr_binding	Binding object showing the mapbmr that can be bound to mapdomain
nat64	Configuration for nat64 config
nat64param	Configuration for NAT64 parameter
nd6	Configuration for nd6
nd6ravariables	Configuration for nd6 Router Advertisment configuration variables

NAME	DESCRIPTION
nd6ravariables_binding	Binding object showing the resources that can be bound to nd6ravariables
nd6ravariables_onlinkipv6prefix_binding	Binding object showing the onlinkipv6prefix that can be bound to nd6ravariables
netbridge	Configuration for network bridge
netbridge_binding	Binding object showing the resources that can be bound to netbridge
netbridge_iptunnel_binding	Binding object showing the iptunnel that can be bound to netbridge
netbridge_nsip6_binding	Binding object showing the nsip6 that can be bound to netbridge
netbridge_nsip_binding	Binding object showing the nsip that can be bound to netbridge
netbridge_vlan_binding	Binding object showing the vlan that can be bound to netbridge
netprofile	Configuration for Network profile
netprofile_binding	Binding object showing the resources that can be bound to netprofile
netprofile_natrule_binding	Binding object showing the natrule that can be bound to netprofile
netprofile_srcportset_binding	Binding object showing the srcportset that can be bound to netprofile
onlinkipv6prefix	Configuration for on-link IPv6 global prefixes for Router Advertisment
ptp	Configuration for Precision Time Protocol
rnat	Configuration for RNAT configured route
rnat6	Configuration for IPv6 RNAT configured route

NAME	DESCRIPTION
rnat6_binding	Binding object showing the resources that can be bound to rnat6
rnat6_nsip6_binding	Binding object showing the nsip6 that can be bound to rnat6
rnat_binding	Binding object showing the resources that can be bound to rnat
rnat_nsip_binding	Binding object showing the nsip that can be bound to rnat
rnatglobal_auditsyslogpolicy_binding	Binding object showing the auditsyslogpolicy that can be bound to rnatglobal
rnatglobal_binding	Binding object showing the resources that can be bound to rnatglobal
rnatparam	Configuration for RNAT parameter
rnatsession	Configuration for RNAT session
route	Configuration for route
route6	Configuration for route 6
rsskeytype	Configuration for RSS key type
vlan	Configuration for "VLAN"
vlan_binding	Binding object showing the resources that can be bound to vlan
vlan_channel_binding	Binding object showing the channel that can be bound to vlan
vlan_interface_binding	Binding object showing the interface that can be bound to vlan
vlan_linkset_binding	Binding object showing the linkset that can be bound to vlan

NAME	DESCRIPTION
vlan_nsip6_binding	Binding object showing the nsip6 that can be bound to vlan
vlan_nsip_binding	Binding object showing the nsip that can be bound to vlan
vrid	Configuration for Virtual Router ID
vrid6	Configuration for Virtual Router ID for IPv6
vrid6_binding	Binding object showing the resources that can be bound to vrid6
vrid6_channel_binding	Binding object showing the channel that can be bound to vrid6
vrid6_interface_binding	Binding object showing the interface that can be bound to vrid6
vrid6_nsip6_binding	Binding object showing the nsip6 that can be bound to vrid6
vrid6_nsip_binding	Binding object showing the nsip that can be bound to vrid6
vrid6_trackinterface_binding	Binding object showing the trackinterface that can be bound to vrid6
vrid_binding	Binding object showing the resources that can be bound to vrid
vrid_channel_binding	Binding object showing the channel that can be bound to vrid
vrid_interface_binding	Binding object showing the interface that can be bound to vrid
vrid_nsip6_binding	Binding object showing the nsip6 that can be bound to vrid
vrid_nsip_binding	Binding object showing the nsip that can be bound to vrid

NAME	DESCRIPTION
vrid_trackinterface_binding	Binding object showing the trackinterface that can be bound to vrid
vridparam	Configuration for VR ID parameter
vxlan	Configuration for "VXLAN"
vxlan_binding	Binding object showing the resources that can be bound to vxlan
vxlan_iptunnel_binding	Binding object showing the iptunnel that can be bound to vxlan
vxlan_nsip6_binding	Binding object showing the nsip6 that can be bound to vxlan
vxlan_nsip_binding	Binding object showing the nsip that can be bound to vxlan
vxlan_srcip_binding	Binding object showing the srcip that can be bound to vxlan
vxlanvlanmap	Configuration for vxlan vlan mapping
vxlanvlanmap_binding	Binding object showing the resources that can be bound to vxlanvlanmap
vxlanvlanmap_vxlan_binding	Binding object showing the vxlan that can be bound to vxlanvlanmap

appalgparam

Configuration for AppAlg Param resource.

Properties

(click to see Operations)

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
pptpgreidletimeout	<Double>	Read-write	Interval in sec, after which data sessions

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
			of PPTP GRE is cleared. Default value: 9000 Minimum value = 1 Maximum value = 9000

Operations

(click to see Properties)

UPDATE| UNSET| GET (ALL)

Some options that you can use for each operations:

- **Getting warnings in response:**NITRO allows you to get warnings in an operation by specifying the "warning" query parameter as "yes". For example, to get warnings while connecting to the Citrix ADC appliance, the URL is as follows:

  http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/login?warning=yes

  If any, the warnings are displayed in the response payload with the HTTP code "209 X-NITRO-WARNING".

- **Authenticated access for individual NITRO operations:**NITRO allows you to logon to the Citrix ADC appliance to perform individual operations. You can use this option instead of creating a NITRO session (using the login object) and then using that session to perform all operations,

  To do this, you must specify the username and password in the request header of the NITRO request as follows:

  X-NITRO-USER:<username>

  X-NITRO-PASS:<password>

  Note: In such cases, make sure that the request header DOES not include the following:

  Cookie:NITRO_AUTH_TOKEN=<tokenvalue>

`*Note:`*

Mandatory parameters are marked in redand placeholder content is marked in <green>.

update

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/appalgparam

HTTP Method:PUT

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Content-Type:application/json

Request Payload:

```
{"appalgparam":{

"pptpgreidletimeout":<Double_value>

}}
```

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error

unset

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/appalgparam?**action=unset**

HTTP Method:POST

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Content-Type:application/json

Request Payload:

```
{"appalgparam":{
```

```
"pptpgreidletimeout":true

}}
```

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error

get (all)

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/appalgparam

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "appalgparam": [ {

"pptpgreidletimeout":<Double_value>

}]}
```

arp

Configuration for arp resource.

Properties

(click to see Operations)

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
ipaddress	<String>	Read-write	IP address of the network device that you want to add to the ARP table. Minimum length = 1
td	<Double>	Read-write	Integer value that uniquely identifies the traffic domain in which you want to configure the entity. If you do not specify an ID, the entity becomes part of the default traffic domain, which has an ID of 0. Minimum value = 0 Maximum value = 4094
mac	<String>	Read-write	MAC address of the network device.
ifnum	<String>	Read-write	Interface through which the network device is accessible. Specify the interface in (slot/port) notation. For example, 1/3.
vxlan	<Double>	Read-write	ID of the VXLAN on which the IP address of this ARP entry is reachable.

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
			Minimum value = 1 Maximum value = 16777215
vtep	<String>	Read-write	IP address of the VXLAN tunnel endpoint (VTEP) through which the IP address of this ARP entry is reachable. Minimum length = 1
vlan	<Double>	Read-write	The VLAN ID through which packets are to be sent after matching the ARP entry. This is a numeric value.
ownernode	<Double>	Read-write	The owner node for the Arp entry. Minimum value = 0 Maximum value = 31
all	<Boolean>	Read-write	Remove all ARP entries from the ARP table of the Citrix ADC.
nodeid	<Double>	Read-write	Unique number that identifies the cluster node. Minimum value = 0 Maximum value = 31

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
timeout	<Double>	Read-only	The time, in seconds, after which the entry times out.
state	<Double>	Read-only	The state of the ARP entry.
flags	<Double>	Read-only	The flags for the entry.
type	<String>	Read-only	Indicates whether this ARP entry was added manually or dynamically. When you manually add an ARP entry, the value for this parameter is STATIC. Otherwise, it is DYNAMIC. For the NSIP and loopback IP addresses, the value is PERMANENT. Possible values = STATIC, PERMANENT, DYNAMIC
channel	<Double>	Read-only	The tunnel, channel, or physical interface through which the ARP entry is identified.
controlplane	<Boolean>	Read-only	This arp entry is populated by a control plane protocol.
__count	<Double>	Read-only	count parameter

Operations

(click to see Properties)

ADD| DELETE| SEND| GET (ALL)| COUNT

Some options that you can use for each operations:

- **Getting warnings in response:**NITRO allows you to get warnings in an operation by specifying the "warning" query parameter as "yes". For example, to get warnings while connecting to the Citrix ADC appliance, the URL is as follows:

  http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/login?warning=yes

  If any, the warnings are displayed in the response payload with the HTTP code "209 X-NITRO-WARNING".

- **Authenticated access for individual NITRO operations:**NITRO allows you to logon to the Citrix ADC appliance to perform individual operations. You can use this option instead of creating a NITRO session (using the login object) and then using that session to perform all operations,

  To do this, you must specify the username and password in the request header of the NITRO request as follows:

  X-NITRO-USER:<username>

  X-NITRO-PASS:<password>

  Note: In such cases, make sure that the request header DOES not include the following:

  Cookie:NITRO_AUTH_TOKEN=<tokenvalue>

`*Note:`*

Mandatory parameters are marked in redand placeholder content is marked in <green>.

add

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/arp

HTTP Method:POST

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Content-Type:application/json

Request Payload:

```
{"arp":{

<b>"ipaddress":<String_value>,

</b>"td":<Double_value>,

<b>"mac":<String_value>,

</b>"ifnum":<String_value>,

"vxlan":<Double_value>,

"vtep":<String_value>,

"vlan":<Double_value>,

"ownernode":<Double_value>

}}
```

Response:

HTTP Status Code on Success: 201 Created HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error

delete

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/arp/ipaddress_value<String>

HTTP Method:DELETE

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue>

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error

send

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/arp?**action=send**

HTTP Method:POST

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Content-Type:application/json

Request Payload:

```
{"arp":{

"ipaddress":<String_value>,

"td":<Double_value>,

"all":<Boolean_value>

}}
```

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error

get (all)

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/arp

Query-parameters:

args

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/arp?
args=ipaddress:<String_value>,td:<Double_value>,ownernode:<Double_value>,nodeid:<Double_value>

Use this query-parameter to get arp resources based on additional properties.

attrs

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/arp?**attrs=property-name1,property-name2**

Use this query parameter to specify the resource details that you want to retrieve.

filter

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/arp?**filter=property-name1:property-val1,property-name2:property-val2**

Use this query-parameter to get the filtered set of arp resources configured on Citrix ADC. Filtering can be done on any of the properties of the resource.

view

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/arp?**view=summary**

Note: By default, the retrieved results are displayed in detail view (?view=detail).

pagination

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/arp?**pagesize=#no;pageno=#no**

Use this query-parameter to get the arp resources in chunks.

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "arp": [ {

ipaddress:<String_value>,td:<Double_value>,ownernode:<Double_value>,nodeid
:<Double_value>"mac":<String_value>,

"ifnum":<String_value>,

"timeout":<Double_value>,

"state":<Double_value>,

"flags":<Double_value>,

"type":<String_value>,

"vlan":<Double_value>,

"vxlan":<Double_value>,

"vtep":<String_value>,

"channel":<Double_value>,

"controlplane":<Boolean_value>

}]}
```

count

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/arp?**count=yes**

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "arp": [ { "__count": "#no"} ] }
```

arpparam

Configuration for Global arp parameters resource.

Properties

(click to see Operations)

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
timeout	<Double>	Read-write	Time-out value (aging time) for the dynamically learned ARP entries, in seconds. The new value applies only to ARP entries that are dynamically learned after the new value is set. Previously existing ARP entries expire after the previously configured aging time. Default value: 1200 Minimum value = 5

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
			Maximum value = 1200
spoofvalidation	<String>	Read-write	enable/disable arp spoofing validation. Default value: DISABLED Possible values = ENABLED, DISABLED

Operations

(click to see Properties)

UPDATE| UNSET| GET (ALL)

Some options that you can use for each operations:

- **Getting warnings in response:**NITRO allows you to get warnings in an operation by specifying the "warning" query parameter as "yes". For example, to get warnings while connecting to the Citrix ADC appliance, the URL is as follows:

  http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/login?warning=yes

  If any, the warnings are displayed in the response payload with the HTTP code "209 X-NITRO-WARNING".

- **Authenticated access for individual NITRO operations:**NITRO allows you to logon to the Citrix ADC appliance to perform individual operations. You can use this option instead of creating a NITRO session (using the login object) and then using that session to perform all operations,

  To do this, you must specify the username and password in the request header of the NITRO request as follows:

  X-NITRO-USER:<username>

  X-NITRO-PASS:<password>

  Note: In such cases, make sure that the request header DOES not include the following:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue>

`*Note:` *

Mandatory parameters are marked in redand placeholder content is marked in <green>.

update

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/arpparam

HTTP Method:PUT

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Content-Type:application/json

Request Payload:

```
{"arpparam":{

"timeout":<Double_value>,

"spoofvalidation":<String_value>

}}
```

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error

unset

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/arpparam?**action=unset**

HTTP Method:POST

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Content-Type:application/json

Request Payload:

```
{"arpparam":{

"timeout":true,

"spoofvalidation":true

}}
```

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error

get (all)

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/arpparam

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "arpparam": [ {

"timeout":<Double_value>,

"spoofvalidation":<String_value>
```

```
}]}
```

bridgegroup

Configuration for bridge group resource.

Properties

(click to see Operations)

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
id	<Double>	Read-write	An integer that uniquely identifies the bridge group. Minimum value = 1 Maximum value = 1000
dynamicrouting	<String>	Read-write	Enable dynamic routing for this bridgegroup. Default value: DISABLED Possible values = ENABLED, DISABLED
ipv6dynamicrouting	<String>	Read-write	Enable all IPv6 dynamic routing protocols on all VLANs bound to this bridgegroup. Note: For the ENABLED setting to work, you must configure IPv6 dynamic routing protocols from the

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
			VTYSH command line. Default value: DISABLED Possible values = ENABLED, DISABLED
flags	<Boolean>	Read-only	Temporary flag used for internal purpose.
portbitmap	<Double>	Read-only	Member interfaces of this bridge group.
tagbitmap	<Double>	Read-only	Tagged members of this bridge group.
ifaces	<String>	Read-only	Names of all member interfaces of this bridge group.
tagifaces	<String>	Read-only	Names of all tagged member interfaces of this bridge group.
rnat	<Boolean>	Read-only	Temporary flag used for internal purpose.
partitionname	<String>	Read-only	Name of the Partition to which this vlan bound to. Minimum length = 1
__count	<Double>	Read-only	count parameter

Operations

(click to see Properties)

ADD| DELETE| UPDATE| UNSET| GET (ALL)| GET| COUNT

Some options that you can use for each operations:

- **Getting warnings in response:**NITRO allows you to get warnings in an operation by specifying the "warning" query parameter as "yes". For example, to get warnings while connecting to the Citrix ADC appliance, the URL is as follows:

  http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/login?warning=yes

  If any, the warnings are displayed in the response payload with the HTTP code "209 X-NITRO-WARNING".

- **Authenticated access for individual NITRO operations:**NITRO allows you to logon to the Citrix ADC appliance to perform individual operations. You can use this option instead of creating a NITRO session (using the login object) and then using that session to perform all operations,

  To do this, you must specify the username and password in the request header of the NITRO request as follows:

  X-NITRO-USER:<username>

  X-NITRO-PASS:<password>

  Note: In such cases, make sure that the request header DOES not include the following:

  Cookie:NITRO_AUTH_TOKEN=<tokenvalue>

`*Note:`*

Mandatory parameters are marked in redand placeholder content is marked in <green>.

add

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/bridgegroup

HTTP Method:POST

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Content-Type:application/json

Request Payload:

```
{"bridgegroup":{

<b>"id":<Double_value>,

</b>"dynamicrouting":<String_value>,

"ipv6dynamicrouting":<String_value>

}}
```

Response:

HTTP Status Code on Success: 201 Created HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error

delete

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/bridgegroup/id_value<Double>

HTTP Method:DELETE

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue>

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error

update

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/bridgegroup

HTTP Method:PUT

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Content-Type:application/json

Request Payload:

```
{"bridgegroup":{

<b>"id":<Double_value>,

</b>"dynamicrouting":<String_value>,

"ipv6dynamicrouting":<String_value>

}}
```

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error

unset

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/bridgegroup?**action=unset**

HTTP Method:POST

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Content-Type:application/json

Request Payload:

```
{"bridgegroup":{

<b>"id":<Double_value>,

</b>"dynamicrouting":true,

"ipv6dynamicrouting":true

}}
```

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error

get (all)

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/bridgegroup

Query-parameters:

attrs

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/bridgegroup?**attrs=property-name1,property-name2**

Use this query parameter to specify the resource details that you want to retrieve.

filter

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/bridgegroup?**filter=property-name1:property-val1,property-name2:property-val2**

Use this query-parameter to get the filtered set of bridgegroup resources configured on Citrix ADC. Filtering can be done on any of the properties of the resource.

view

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/bridgegroup?**view=summary**

Note: By default, the retrieved results are displayed in detail view (?view=detail).

pagination

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/bridgegroup?**pagesize=#no;pageno=#no**

Use this query-parameter to get the bridgegroup resources in chunks.

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "bridgegroup": [ {

"id":<Double_value>,

"flags":<Boolean_value>,

"portbitmap":<Double_value>,

"tagbitmap":<Double_value>,

"ifaces":<String_value>,

"tagifaces":<String_value>,

"dynamicrouting":<String_value>,

"ipv6dynamicrouting":<String_value>,

"rnat":<Boolean_value>,

"partitionname":<String_value>

}]}
```

get

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/bridgegroup/id_value<Double>

Query-parameters:

attrs

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/bridgegroup/id_value<Double>?**attrs=property-name1,property-name2**

Use this query parameter to specify the resource details that you want to retrieve.

view

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/bridgegroup/id_value<Double>?**view=summary**

Note: By default, the retrieved results are displayed in detail view (?view=detail).

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "bridgegroup": [ {

"id":<Double_value>,

"flags":<Boolean_value>,

"portbitmap":<Double_value>,

"tagbitmap":<Double_value>,

"ifaces":<String_value>,

"tagifaces":<String_value>,
```

```
"dynamicrouting":<String_value>,

"ipv6dynamicrouting":<String_value>,

"rnat":<Boolean_value>,

"partitionname":<String_value>

}]}
```

count

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/bridgegroup?**count=yes**

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "bridgegroup": [ { "__count": "#no"} ] }
```

bridgegroup_binding

Binding object which returns the resources bound to bridgegroup.

Properties

(click to see Operations)

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
id	<Double>	Read-write	The name of the bridge group. Minimum value = 1 Maximum value = 1000
bridgegroup_nsip_binding	<bridgegroup_nsip_binding[]>	Read-only	nsip that can be bound to bridgegroup.
bridgegroup_vlan_binding	<bridgegroup_vlan_binding[]>	Read-only	vlan that can be bound to bridgegroup.
bridgegroup_nsip6_binding	<bridgegroup_nsip6_binding[]>	Read-only	nsip6 that can be bound to bridgegroup.

Operations

(click to see Properties)

GET| GET (ALL)

Some options that you can use for each operations:

- **Getting warnings in response:**NITRO allows you to get warnings in an operation by specifying the "warning" query parameter as "yes". For example, to get warnings while connecting to the Citrix ADC appliance, the URL is as follows:

  http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/login?warning=yes

  If any, the warnings are displayed in the response payload with the HTTP code "209 X-NITRO-WARNING".
- **Authenticated access for individual NITRO operations:**NITRO allows you to logon to the Citrix ADC appliance to perform individual operations. You can use this option instead of creating a NITRO session (using the login object) and then using that session to perform all operations,

To do this, you must specify the username and password in the request header of the NITRO request as follows:

X-NITRO-USER:<username>

X-NITRO-PASS:<password>

Note: In such cases, make sure that the request header DOES not include the following:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue>

`*Note:`*

Mandatory parameters are marked in redand placeholder content is marked in <green>.

get

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/bridgegroup_binding/id_value<Double>

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "bridgegroup_binding": [ {

"id":<Double_value>,

"bridgegroup_nsip_binding":<bridgegroup_nsip_binding[]_value>,

"bridgegroup_vlan_binding":<bridgegroup_vlan_binding[]_value>,

"bridgegroup_nsip6_binding":<bridgegroup_nsip6_binding[]_value>
```

```
}]}
```

get (all)

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/bridgegroup_binding

Query-parameters:

bulkbindings

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/bridgegroup_binding?**bulkbindings=yes**

NITRO allows you to fetch bindings in bulk.

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "bridgegroup_binding": [ {

"id":<Double_value>,

"bridgegroup_nsip_binding":<bridgegroup_nsip_binding[]_value>,

"bridgegroup_vlan_binding":<bridgegroup_vlan_binding[]_value>,

"bridgegroup_nsip6_binding":<bridgegroup_nsip6_binding[]_value>

}]}
```

bridgegroup_nsip6_binding

Binding object showing the nsip6 that can be bound to bridgegroup.

Properties

(click to see Operations)

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
ownergroup	<String>	Read-write	The owner node group in a Cluster for this vlan. Default value: DEFAULT_NG Minimum length = 1
netmask	<String>	Read-write	A subnet mask associated with the network address. Minimum length = 1
id	<Double>	Read-write	The integer that uniquely identifies the bridge group. Minimum value = 1 Maximum value = 1000
td	<Double>	Read-write	Integer value that uniquely identifies the traffic domain in which you want to configure the entity. If you do not specify an ID, the entity becomes part of the default traffic

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
			domain, which has an ID of 0. Minimum value = 0 Maximum value = 4094
ipaddress	<String>	Read-write	The IP address assigned to the bridge group.
rnat	<Boolean>	Read-only	Temporary flag used for internal purpose.
__count	<Double>	Read-only	count parameter

Operations

(click to see Properties)

ADD:| DELETE:| GET| GET (ALL)| COUNT

Some options that you can use for each operations:

- **Getting warnings in response:**NITRO allows you to get warnings in an operation by specifying the "warning" query parameter as "yes". For example, to get warnings while connecting to the Citrix ADC appliance, the URL is as follows:

  http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/login?warning=yes

  If any, the warnings are displayed in the response payload with the HTTP code "209 X-NITRO-WARNING".

- **Authenticated access for individual NITRO operations:**NITRO allows you to logon to the Citrix ADC appliance to perform individual operations. You can use this option instead of creating a NITRO session (using the login object) and then using that session to perform all operations,

  To do this, you must specify the username and password in the request header of the NITRO request as follows:

  X-NITRO-USER:<username>

X-NITRO-PASS:<password>

Note: In such cases, make sure that the request header DOES not include the following:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue>

`*Note:`*

Mandatory parameters are marked in redand placeholder content is marked in <green>.

add

URL:http://<Citrix-ADC-IP-address(NSIP)/nitro/v1/config/bridgegroup_nsip6_binding

HTTP Method:PUT

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Content-Type:application/json

Request Payload:

```
{
"bridgegroup_nsip6_binding":{
<b>"id":<Double_value>,
</b>"ipaddress":<String_value>,
"netmask":<String_value>,
"td":<Double_value>,
"ownergroup":<String_value>
}}
```

Response:

HTTP Status Code on Success: 201 Created HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error

delete

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/bridgegroup_nsip6_binding/id_value<Double>

Query-parameters:

args

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/bridgegroup_nsip6_binding/id_value<Double>?**args=ipaddress:<String_value>,netmask:<String_value>,td:<Double_value>,ownergroup:<String_value**

HTTP Method:DELETE

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue>

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error

get

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/bridgegroup_nsip6_binding/id_value<Double>

Query-parameters:

filter

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/bridgegroup_nsip6_binding/id_value<Double>?**filter=property-name1:property-value1,property-name2:property-value2**

Use this query-parameter to get the filtered set of bridgegroup_nsip6_binding resources configured on Citrix ADC. Filtering can be done on any of the properties of the resource.

pagination

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/bridgegroup_nsip6_binding/id_value<Double>?**pagesize=#no;pageno=#no**

Use this query-parameter to get the bridgegroup_nsip6_binding resources in chunks.

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "bridgegroup_nsip6_binding": [ {

"ownergroup":<String_value>,

"netmask":<String_value>,

"id":<Double_value>,

"td":<Double_value>,

"ipaddress":<String_value>,

"rnat":<Boolean_value>

}]}
```

get (all)

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/bridgegroup_nsip6_binding

Query-parameters:

bulkbindings

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/bridgegroup_nsip6_binding?**bulkbindings=yes**

NITRO allows you to fetch bindings in bulk.

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "bridgegroup_nsip6_binding": [ {

"ownergroup":<String_value>,

"netmask":<String_value>,

"id":<Double_value>,

"td":<Double_value>,

"ipaddress":<String_value>,

"rnat":<Boolean_value>

}]}
```

count

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/bridgegroup_nsip6_binding/id_value<Double>?**count=yes**

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{"bridgegroup_nsip6_binding": [ { "__count": "#no"} ] }
```

bridgegroup_nsip_binding

Binding object showing the nsip that can be bound to bridgegroup.

Properties

(click to see Operations)

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
ownergroup	<String>	Read-write	The owner node group in a Cluster for this vlan. Default value: DEFAULT_NG Minimum length = 1

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
id	<Double>	Read-write	The integer that uniquely identifies the bridge group. Minimum value = 1 Maximum value = 1000
netmask	<String>	Read-write	The network mask for the subnet defined for the bridge group.
td	<Double>	Read-write	Integer value that uniquely identifies the traffic domain in which you want to configure the entity. If you do not specify an ID, the entity becomes part of the default traffic domain, which has an ID of 0. Minimum value = 0 Maximum value = 4094
ipaddress	<String>	Read-write	The IP address assigned to the bridge group.
rnat	<Boolean>	Read-only	Temporary flag used for internal purpose.
__count	<Double>	Read-only	count parameter

Operations

(click to see Properties)

ADD:| DELETE:| GET| GET (ALL)| COUNT

Some options that you can use for each operations:

- **Getting warnings in response:**NITRO allows you to get warnings in an operation by specifying the "warning" query parameter as "yes". For example, to get warnings while connecting to the Citrix ADC appliance, the URL is as follows:

  http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/login?warning=yes

  If any, the warnings are displayed in the response payload with the HTTP code "209 X-NITRO-WARNING".

- **Authenticated access for individual NITRO operations:**NITRO allows you to logon to the Citrix ADC appliance to perform individual operations. You can use this option instead of creating a NITRO session (using the login object) and then using that session to perform all operations,

  To do this, you must specify the username and password in the request header of the NITRO request as follows:

  X-NITRO-USER:<username>

  X-NITRO-PASS:<password>

  Note: In such cases, make sure that the request header DOES not include the following:

  Cookie:NITRO_AUTH_TOKEN=<tokenvalue>

`*Note:`*

Mandatory parameters are marked in redand placeholder content is marked in <green>.

add

URL:http://<Citrix-ADC-IP-address(NSIP)/nitro/v1/config/bridgegroup_nsip_binding

HTTP Method:PUT

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Content-Type:application/json

Request Payload:

```
{

"bridgegroup_nsip_binding":{

<b>"id":<Double_value>,

</b>"ipaddress":<String_value>,

"netmask":<String_value>,

"td":<Double_value>,

"ownergroup":<String_value>

}}
```

Response:

HTTP Status Code on Success: 201 Created HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error

delete

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/bridgegroup_nsip_binding/id_value<Double>

Query-parameters:

args

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/bridgegroup_nsip_binding/id_value<Double>?**args=ipaddress:<String_value>,netmask:<String_value>,td:<Double_value>,ownergroup:<String_value**

HTTP Method:DELETE

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue>

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error

get

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/bridgegroup_nsip_binding/id_value<Double>

Query-parameters:

filter

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/bridgegroup_nsip_binding/id_value<Double>?**filter=property-name1:property-value1,property-name2:property-value2**

Use this query-parameter to get the filtered set of bridgegroup_nsip_binding resources configured on Citrix ADC. Filtering can be done on any of the properties of the resource.

pagination

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/bridgegroup_nsip_binding/id_value<Double>?**pagesize=#no;pageno=#no**

Use this query-parameter to get the bridgegroup_nsip_binding resources in chunks.

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "bridgegroup_nsip_binding": [ {
```

```
"ownergroup":<String_value>,

"id":<Double_value>,

"netmask":<String_value>,

"td":<Double_value>,

"ipaddress":<String_value>,

"rnat":<Boolean_value>

}]}
```

get (all)

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/bridgegroup_nsip_binding

Query-parameters:

bulkbindings

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/bridgegroup_nsip_binding?**bulkbindings=yes**

NITRO allows you to fetch bindings in bulk.

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "bridgegroup_nsip_binding": [ {

"ownergroup":<String_value>,

"id":<Double_value>,

"netmask":<String_value>,

"td":<Double_value>,

"ipaddress":<String_value>,

"rnat":<Boolean_value>

}]}
```

count

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/bridgegroup_nsip_binding/id_value<Double>?**count=yes**

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{"bridgegroup_nsip_binding": [ { "__count": "#no"} ] }
```

bridgegroup_vlan_binding

Binding object showing the vlan that can be bound to bridgegroup.

Properties

(click to see Operations)

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
vlan	<Double>	Read-write	Names of all member VLANs.
id	<Double>	Read-write	The integer that uniquely identifies the bridge group. Minimum value = 1 Maximum value = 1000
rnat	<Boolean>	Read-only	Temporary flag used for internal purpose.
__count	<Double>	Read-only	count parameter

Operations

(click to see Properties)

ADD:| DELETE:| GET| GET (ALL)| COUNT

Some options that you can use for each operations:

- **Getting warnings in response:**NITRO allows you to get warnings in an operation by specifying the "warning" query parameter as "yes". For example, to get warnings while connecting to the Citrix ADC appliance, the URL is as follows:

  http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/login?warning=yes

  If any, the warnings are displayed in the response payload with the HTTP code "209 X-NITRO-WARNING".

- **Authenticated access for individual NITRO operations:**NITRO allows you to logon to the Citrix ADC appliance to perform individual operations. You can use this option instead of creating a NITRO session (using the login object) and then using that session to perform all operations,

  To do this, you must specify the username and password in the request header of the NITRO request as follows:

  X-NITRO-USER:<username>

  X-NITRO-PASS:<password>

  Note: In such cases, make sure that the request header DOES not include the following:

  Cookie:NITRO_AUTH_TOKEN=<tokenvalue>

`*Note:`*

Mandatory parameters are marked in redand placeholder content is marked in <green>.

add

URL:http://<Citrix-ADC-IP-address(NSIP)/nitro/v1/config/bridgegroup_vlan_binding

HTTP Method:PUT

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Content-Type:application/json

Request Payload:

```
{

"bridgegroup_vlan_binding":{

<b>"id":<Double_value>,

</b>"vlan":<Double_value>

}}
```

Response:

HTTP Status Code on Success: 201 Created HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error

delete

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/bridgegroup_vlan_binding/id_value<Double>

Query-parameters:

args

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/bridgegroup_vlan_binding/id_value<Double>?**args=vlan:<Double_value>**

HTTP Method:DELETE

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue>

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error

get

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/bridgegroup_vlan_binding/id_value<Double>

Query-parameters:

filter

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/bridgegroup_vlan_binding/id_value<Double>?**filter=property-name1:property-value1,property-name2:property-value2**

Use this query-parameter to get the filtered set of bridgegroup_vlan_binding resources configured on Citrix ADC. Filtering can be done on any of the properties of the resource.

pagination

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/bridgegroup_vlan_binding/id_value<Double>?**pagesize=#no;pageno=#no**

Use this query-parameter to get the bridgegroup_vlan_binding resources in chunks.

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "bridgegroup_vlan_binding": [ {

"vlan":<Double_value>,

"id":<Double_value>,

"rnat":<Boolean_value>

}]}
```

get (all)

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/bridgegroup_vlan_binding

Query-parameters:

bulkbindings

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/bridgegroup_vlan_binding?**bulkbindings=yes**

NITRO allows you to fetch bindings in bulk.

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "bridgegroup_vlan_binding": [ {

"vlan":<Double_value>,

"id":<Double_value>,

"rnat":<Boolean_value>

}]}
```

count

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/bridgegroup_vlan_binding/id_value<Double>?**count=yes**

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{"bridgegroup_vlan_binding": [ { "__count": "#no"} ] }
```

bridgetable

Configuration for bridge table entry resource.

Properties

(click to see Operations)

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
mac	<String>	Read-write	The MAC address of the target.
vxlan	<Double>	Read-write	The VXLAN to which this address is associated. Minimum value = 1 Maximum value = 16777215
vtep	<String>	Read-write	The IP address of the destination VXLAN tunnel endpoint where the Ethernet MAC ADDRESS resides. Minimum length = 1

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
vni	<Double>	Read-write	The VXLAN VNI Network Identifier (or VXLAN Segment ID) to use to connect to the remote VXLAN tunnel endpoint. If omitted the value specified as vxlan will be used. Minimum value = 1 Maximum value = 16777215
devicevlan	<Double>	Read-write	The vlan on which to send multicast packets when the VXLAN tunnel endpoint is a muticast group address. Minimum value = 1 Maximum value = 4094
bridgeage	<Double>	Read-write	Time-out value for the bridge table entries, in seconds. The new value applies only to the entries that are dynamically learned after the new value is set. Previously existing bridge table entries expire after the previously configured time-out value.

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
			Default value: 300 Minimum value = 60 Maximum value = 300
nodeid	<Double>	Read-write	Unique number that identifies the cluster node. Minimum value = 0 Maximum value = 31
vlan	<Double>	Read-write	VLAN whose entries are to be removed. Minimum value = 1 Maximum value = 4094
ifnum	<String>	Read-write	INTERFACE whose entries are to be removed.
flags	<Double>	Read-only	Display flags,.
type	<String>	Read-only	Whether static or dynamic. Possible values = STATIC, PERMANENT, DYNAMIC
channel	<Double>	Read-only	The Tunnel through which bridge entry is learned.
controlplane	<Boolean>	Read-only	This bridge table entry is populated by

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
			a control plane protocol.
__count	<Double>	Read-only	count parameter

Operations

(click to see Properties)

ADD| DELETE| UPDATE| UNSET| CLEAR| GET (ALL)| COUNT

Some options that you can use for each operations:

- **Getting warnings in response:**NITRO allows you to get warnings in an operation by specifying the "warning" query parameter as "yes". For example, to get warnings while connecting to the Citrix ADC appliance, the URL is as follows:

  http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/login?warning=yes

  If any, the warnings are displayed in the response payload with the HTTP code "209 X-NITRO-WARNING".
- **Authenticated access for individual NITRO operations:**NITRO allows you to logon to the Citrix ADC appliance to perform individual operations. You can use this option instead of creating a NITRO session (using the login object) and then using that session to perform all operations,

  To do this, you must specify the username and password in the request header of the NITRO request as follows:

  X-NITRO-USER:<username>

  X-NITRO-PASS:<password>

  Note: In such cases, make sure that the request header DOES not include the following:

  Cookie:NITRO_AUTH_TOKEN=<tokenvalue>

`*Note:`*

Mandatory parameters are marked in redand placeholder content is marked in <green>.

add

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/bridgetable

HTTP Method:POST

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Content-Type:application/json

Request Payload:

```
{"bridgetable":{

<b>"mac":<String_value>,

</b><b>"vxlan":<Double_value>,

</b><b>"vtep":<String_value>,

</b>"vni":<Double_value>,

"devicevlan":<Double_value>

}}
```

Response:

HTTP Status Code on Success: 201 Created HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error

delete

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/bridgetable

HTTP Method:DELETE

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue>

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error

update

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/bridgetable

HTTP Method:PUT

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Content-Type:application/json

Request Payload:

```
{"bridgetable":{

"bridgeage":<Double_value>

}}
```

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error

unset

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/bridgetable?**action=unset**

HTTP Method:POST

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Content-Type:application/json

Request Payload:

```
{"bridgetable":{

"bridgeage":true

}}
```

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error

clear

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/bridgetable?**action=clear**

HTTP Method:POST

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Content-Type:application/json

Request Payload:

```
{"bridgetable":{

"vlan":<Double_value>,

"ifnum":<String_value>,

"vxlan":<Double_value>

}}
```

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error

get (all)

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/bridgetable

Query-parameters:

args

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/bridgetable?**args=nodeid:<Double_value>**

Use this query-parameter to get bridgetable resources based on additional properties.

attrs

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/bridgetable?**attrs=property-name1,property-name2**

Use this query parameter to specify the resource details that you want to retrieve.

filter

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/bridgetable?**filter=property-name1:property-val1,property-name2:property-val2**

Use this query-parameter to get the filtered set of bridgetable resources configured on Citrix ADC. Filtering can be done on any of the properties of the resource.

view

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/bridgetable?**view=summary**

Note: By default, the retrieved results are displayed in detail view (?view=detail).

pagination

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/bridgetable?**pagesize=#no;pageno=#no**

Use this query-parameter to get the bridgetable resources in chunks.

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "bridgetable": [ {

nodeid:<Double_value>"bridgeage":<Double_value>,

"mac":<String_value>,

"ifnum":<String_value>,

"vlan":<Double_value>,

"vxlan":<Double_value>,

"vtep":<String_value>,

"vni":<Double_value>,

"devicevlan":<Double_value>,

"flags":<Double_value>,

"type":<String_value>,

"channel":<Double_value>,

"controlplane":<Boolean_value>

}]}
```

count

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/bridgetable?**count=yes**

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "bridgetable": [ { "__count": "#no"} ] }
```

channel

Configuration for channel resource.

Properties

(click to see Operations)

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
id	<String>	Read-write	ID for the LA channel or cluster LA channel or LR channel to be created. Specify an LA channel in LA/x notation, where x can range from 1 to 8 or cluster LA channel in CLA/x

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
			notation or Link redundant channel in LR/x notation, where x can range from 1 to 4. Cannot be changed after the LA channel is created.
ifnum	<String[]>	Read-write	Interfaces to be bound to the LA channel of a Citrix ADC or to the LA channel of a cluster configuration. For an LA channel of a Citrix ADC, specify an interface in C/U notation (for example, 1/3). For an LA channel of a cluster configuration, specify an interface in N/C/U notation (for example, 2/1/3). where C can take one of the following values: * 0 - Indicates a management interface. * 1 - Indicates a 1 Gbps port.

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
			* 10 - Indicates a 10 Gbps port. U is a unique integer for representing an interface in a particular port group. N is the ID of the node to which an interface belongs in a cluster configuration. Use spaces to separate multiple entries.
state	<String>	Read-write	Enable or disable the LA channel. Default value: ENABLED Possible values = ENABLED, DISABLED
mode	<String>	Read-write	The initital mode for the LA channel. Possible values = MANUAL, AUTO
conndistr	<String>	Read-write	The 'connection' distribution mode for the LA channel.

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
			Possible values = DISABLED, ENABLED
macdistr	<String>	Read-write	The 'MAC' distribution mode for the LA channel. Possible values = SOURCE, DESTINATION, BOTH
lamac	<String>	Read-write	Specifies a MAC address for the LA channels configured in Citrix ADC virtual appliances (VPX). This MAC address is persistent after each reboot. If you don't specify this parameter, a MAC address is generated randomly for each LA channel. These MAC addresses change after each reboot.
speed	<String>	Read-write	Ethernet speed of the channel, in Mbps. If the speed of any bound interface is greater than or equal to the value set for this parameter, the state of the interface is UP.

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
			Otherwise, the state is INACTIVE. Bound Interfaces whose state is INACTIVE do not process any traffic. Default value: AUTO Possible values = AUTO, 10, 100, 1000, 10000, 25000, 40000, 50000, 100000
flowctl	<String>	Read-write	Specifies the flow control type for this LA channel to manage the flow of frames. Flow control is a function as mentioned in clause 31 of the IEEE 802.3 standard. Flow control allows congested ports to pause traffic from the peer device. Flow control is achieved by sending PAUSE frames. Default value: OFF Possible values = OFF, RX, TX, RXTX, ON
hamonitor	<String>	Read-write	In a High Availability (HA) configuration, monitor the LA

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
			channel for failure events. Failure of any LA channel that has HA MON enabled triggers HA failover. Default value: ON Possible values = ON, OFF
haheartbeat	<String>	Read-write	In a High Availability (HA) configuration, configure the LA channel for sending heartbeats. LA channel that has HA Heartbeat disabled should not send the heartbeats. Default value: ON Possible values = OFF, ON
tagall	<String>	Read-write	Adds a four-byte 802.1q tag to every packet sent on this channel. The ON setting applies tags for all VLANs that are bound to this channel. OFF applies the tag for all VLANs other than the native VLAN. Default value: OFF Possible values = ON, OFF

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
trunk	<String>	Read-write	This is deprecated by tagall. Default value: OFF Possible values = ON, OFF
ifalias	<String>	Read-write	Alias name for the LA channel. Used only to enhance readability. To perform any operations, you have to specify the LA channel ID. Default value: " " Maximum length = 31
throughput	<Double>	Read-write	Low threshold value for the throughput of the LA channel, in Mbps. In an high availability (HA) configuration, failover is triggered when the LA channel has HA MON enabled and the throughput is below the specified threshold. Minimum value = 0 Maximum value = 160000
bandwidthhigh	<Double>	Read-write	High threshold value for the bandwidth usage of the LA

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
			channel, in Mbps. The Citrix ADC generates an SNMP trap message when the bandwidth usage of the LA channel is greater than or equal to the specified high threshold value. Minimum value = 0 Maximum value = 160000
bandwidthnormal	<Double>	Read-write	Normal threshold value for the bandwidth usage of the LA channel, in Mbps. When the bandwidth usage of the LA channel returns to less than or equal to the specified normal threshold after exceeding the high threshold, the Citrix ADC generates an SNMP trap message to indicate that the bandwidth usage has returned to normal. Minimum value = 0 Maximum value = 160000
mtu	<Double>	Read-write	The maximum transmission unit (MTU) is the largest packet size,

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
			measured in bytes excluding 14 bytes ethernet header and 4 bytes crc, that can be transmitted and received by this interface. Default value of MTU is 1500 on all the interface of Citrix ADC any value configured more than 1500 on the interface will make the interface as jumbo enabled. In case of cluster backplane interface MTU value will be changed to 1514 by default, user has to change the backplane interface value to maximum mtu configured on any of the interface in cluster system plus 14 bytes more for backplane interface if Jumbo is enabled on any of the interface in a cluster system. Changing the backplane will bring back the MTU of backplane interface to default value of 1500. If a channel is configured as backplane then the same holds true for

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
			channel as well as member interfaces. Default value: 1500 Minimum value = 1500 Maximum value = 9216
lrminthroughput	<Double>	Read-write	Specifies the minimum throughput threshold (in Mbps) to be met by the active subchannel. Setting this parameter automatically divides an LACP channel into logical subchannels, with one subchannel active and the others in standby mode. When the maximum supported throughput of the active channel falls below the lrMinThroughput value, link failover occurs and a standby subchannel becomes active. Minimum value = 0 Maximum value = 80000
linkredundancy	<String>	Read-write	Link Redundancy for Cluster LAG.

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
			Default value: OFF Possible values = ON, OFF
devicename	<String>	Read-only	LA channel name in form LA/x, where x is channel ID, which ranges from 1 to 8 or LR channel name in form LR/x, where x is channel ID, which ranges from 1 to 4.
unit	<Double>	Read-only	Unit number of the channel. This is an internal reference number that the Citrix ADC uses to identify the channel.
description	<String>	Read-only	The IEEE standard that the channel is based on.
flags	<Double>	Read-only	Flags of this channel.
actualmtu	<Double>	Read-only	MTU of the channel. This is the maximum frame size that the channel can process.
vlan	<Double>	Read-only	Native VLAN of the channel.
mac	<String>	Read-only	MAC address of the channel.
uptime	<Double>	Read-only	Duration for which the channel is UP. (Example: 3 hours 1 minute 1 second).

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
			This value is reset when the channel state changes to DOWN.
downtime	<Double>	Read-only	Duration for which the channel is DOWN. (Example: 3 hours 1 minute 1 second). This value is reset when the channel state changes to UP.
reqmedia	<String>	Read-only	Requested media setting for this channel. Since there is no media associated with LA, the displayed values carry no significance. Possible values = AUTO, UTP, FIBER
reqspeed	<String>	Read-only	Requested speed setting for this channel. Since no media are associated with LA, this speed is used to determine the threshold for the slave interfaces. If the speed of the member interface is less than the requested speed, that interface is considered inactive.

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
			Possible values = AUTO, 10, 100, 1000, 10000, 25000, 40000, 50000, 100000
reqduplex	<String>	Read-only	Requested duplex setting for this channel. Since no media are associated with LA, the displayed values carry no significance. Possible values = AUTO, HALF, FULL
reqflowcontrol	<String>	Read-only	Requested flow control setting for this channel. Since no media are associated with LA, the displayed values carry no significance. Possible values = OFF, RX, TX, RXTX, ON
media	<String>	Read-only	Requested media setting for this interface. Possible values = AUTO, UTP, FIBER
actspeed	<String>	Read-only	Actual speed setting for this channel. Possible values = AUTO, 10, 100, 1000, 10000,

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
			25000, 40000, 50000, 100000
duplex	<String>	Read-only	Actualduplex setting for this interface. Possible values = AUTO, HALF, FULL
actflowctl	<String>	Read-only	Actual flow control setting for this channel. Possible values = OFF, RX, TX, RXTX, ON
lamode	<String>	Read-only	The mode(AUTO/MANNUAL) for the LA channel. Possible values = MANUAL, AUTO
autoneg	<Double>	Read-only	Requested auto negotiation setting for this channel. Since no media are associated with LA, this setting has no effect.
autonegresult	<Double>	Read-only	Actual auto negotiation setting for this channel.
tagged	<Double>	Read-only	VLAN tags setting on this channel.
taggedany	<Double>	Read-only	Channel setting to accept/drop all tagged packets.

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
taggedautolearn	<Double>	Read-only	Dynaminc vlan membership on this channel.
hangdetect	<Double>	Read-only	Hang detect for this channel.
hangreset	<Double>	Read-only	Hang reset for this channel.
linkstate	<Double>	Read-only	The current state of the link associated with the interface. For logical interfaces (LA), the state of the link is dependent on the state of the slave interfaces. For the link to be UP at least one of the slave interfaces needs to be UP.
intfstate	<Double>	Read-only	Current state of the specified interface. The interface state set to UP only if the link state is UP and administrative state is ENABLED.
rxpackets	<Double>	Read-only	Number of bytes received by all the slave interfaces of the channel since the Citrix ADC was started or the interface statistics were cleared.

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
rxbytes	<Double>	Read-only	Number of packets received by all member interfaces since the Citrix ADC was started or the interface statistics were cleared.
rxerrors	<Double>	Read-only	Number of inbound packets dropped by the hardware of the slave interfaces since the Citrix ADC was started or the interface statistics were cleared. Possible causes of dropped packets are CRC, length (undersize or oversize), and alignment errors.
rxdrops	<Double>	Read-only	Number of inbound packets dropped by the channel's slave interfaces. Commonly dropped packets are multicast frames, spanning tree BPDUs, packets destined to a MAC not owned by the Citrix ADC when L2 mode is disabled, or packets tagged for a VLAN that is not bound to the interface. In most healthy networks, this statistic

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
			increments at a steady rate regardless of traffic load. A sharp spike in dropped packets generally indicates an issue with connected L2 switches, such as a forwarding database overflow resulting in packets being broadcast on all ports.
txpackets	<Double>	Read-only	Number of packets transmitted by slave interfaces of a channel since the Citrix ADC was started or the interface statistics were cleared.
txbytes	<Double>	Read-only	Number of bytes transmitted by slave interfaces of a channel since the Citrix ADC was started or the interface statistics were cleared.
txerrors	<Double>	Read-only	Number of outbound packets dropped by the hardware of a channel's slave interfaces since the Citrix ADC was started or the interface statistics

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
			were cleared. Possible causes of dropped packets are length (undersize or oversize) errors and lack of resources.
txdrops	<Double>	Read-only	Number of packets dropped in transmission by a channel's slave interfaces for one of the following reasons: (1) VLAN mismatch. (2) Oversized packets. (3) Interface congestion. (4) Loopback packets sent on non-loopback interface.
indisc	<Double>	Read-only	Number of error-free inbound packets discarded by a channel's slave interfaces because of a lack of resources (for example, insufficient receive buffers).
outdisc	<Double>	Read-only	Number of error-free outbound packets discarded by a channel's slave

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
			interfaces because of a lack of resources. This statistic is not available on: (1) 10G ports of Citrix ADC MPX 12500/12500/15500 -10G platforms. (2) 10G data ports on Citrix ADC MPX 17500/19500/21500 platforms.
fctls	<Double>	Read-only	Number of times flow control is performed on a channel's slave interfaces because of pause frames.
hangs	<Double>	Read-only	Number of hangs that occurred on the channel's slave interfaces.
stsstalls	<Double>	Read-only	Number of status stalls that occurred on the channel's slave interfaces.
txstalls	<Double>	Read-only	Number of Tx stalls happened that occurred on the channel's slave interfaces.
rxstalls	<Double>	Read-only	Number of Rx stalls that occurred on the

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
			channel's slave interfaces.
bdgmuted	<Double>	Read-only	Number of times a channel's slave interfaces stopped transmitting and receiving packets because of MAC moves between ports.
vmac	<String>	Read-only	Virtual MAC of this channel.
vmac6	<String>	Read-only	Virtual MAC for IPv6 on this interface.
reqthroughput	<Double>	Read-only	Minimum required throughput for an interface. Failover is triggered if the operating throughput of a Link Aggregation (LA) channel for which HAMON is ON falls below this value. Minimum value = 0 Maximum value = 160000
actthroughput	<Double>	Read-only	Actual throughput for the interface.
backplane	<String>	Read-only	The cluster backplane status of the LA. If the status is enabled, the LA is part of the cluster backplane. By default, the

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
			backplane status is disabled. Possible values = ENABLED, DISABLED
cleartime	<Double>	Read-only	Time since the interface stats are cleared last time.
lacpmode	<String>	Read-only	The LACP mode of the specified interface. The possible values are: 1. Active: A port in active mode generates LACP protocol messages on a regular basis, regardless of any need expressed by its partner to receive them. 2. Passive: A port in passive mode generally does not transmit LACP messages unless its partner is in the active mode; that is, it does not speak unless spoken to. 3. Disabled: Removes the interface from the LA channel. If this is

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
			only interface in the LA channel, the LA channel is also deleted. . Possible values = DISABLED, ACTIVE, PASSIVE
lacptimeout	<String>	Read-only	Time to wait for the LACPDU. If a LACPDU is not received within this interval, the Citrix ADC markes the link partner port as DOWN. Possible values: Long and Short. Long lacptimeout is 90 sec and Short LACP timeout is 3 sec. Possible values = LONG, SHORT
lacpactorpriority	<Double>	Read-only	LACP Actor Priority. A LACP port priority is configured on each port using LACP. LACP uses the port priority with the port number to form the port identifier. The port priority determines which ports should be put in standby mode when there is

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
			a hardware limitation that prevents all compatible ports from aggregating.
lacpactorportno	<Double>	Read-only	LACP Actor port number. LACP uses the port priority with the port number to form the port identifier.
lacppartnerstate	<String>	Read-only	LACP Partner State. Whether the port is in Active or Passive negotiating state. Possible values = MANUAL, AUTO
lacppartnertimeout	<String>	Read-only	The timeout value for the information revieved in LACPDUs. It can have values as SHORT or LONG. The SHORT timeout is 3s and the LONG timeout is 90s. Possible values = LONG, SHORT
lacppartneraggregation	<String>	Read-only	The Aggregation flag indicates that the participant will allow the link to be used as part of an aggregate. Otherwise the link is to be used as an individual link, i.e.

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
			not aggregated with any other. Possible values = NS_EMPTY_STR, AGG
lacppartnerinsync	<String>	Read-only	The Synchronization flag indicates that the transmitting participant.s mux component is in sync with the system id and key information transmitted. Possible values = NS_EMPTY_STR, SYNC
lacppartnercollecting	<String>	Read-only	The Collecting flag indicates that the participant.s collector, i.e. the reception component of the mux, is definitely on. If set the flag communicates collecting. Possible values = NS_EMPTY_STR, COLLECTING
lacppartnerdistributing	<String>	Read-only	The Distributing flag indicates that the participant.s distributor is not definitely off. If reset the flag indicates not distributing.

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
			Possible values = NS_EMPTY_STR, DISTRIBUTING
lacppartnerdefaulted	<String>	Read-only	If the timer expires in the Expired state, the Receive Machine enters the Defaulted state. Possible values = NS_EMPTY_STR, DEFAULTED
lacppartnerexpired	<String>	Read-only	If the LACPDUs are received for timeout period, the Receive Machine enters the Expired state and the timer is restarted with the timeout value of SHORT timeout. Possible values = NS_EMPTY_STR, EXPIRED
lacppartnerpriority	<Double>	Read-only	LACP Partner Priority. A LACP port priority is configured on each port using LACP. LACP uses the port priority with the port number to form the port identifier. The port priority determines which ports should be put in standby mode

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
			when there is a hardware limitation that prevents all compatible ports from aggregating.
lacppartnersystemmac	<String>	Read-only	LACP Partner System MAC.
lacppartnersystempriority	<Double>	Read-only	LACP Partner System Priority. The LACP partner's system priority. The values for the priority range from 0 to 65535. The lower the value, the higher the system priority. The switch with the lower system priority value determines which links between LACP partner are active and which are in the standby for each LACP Channel.
lacppartnerportno	<Double>	Read-only	LACP Partner Port number. LACP uses the port priority with the port number to form the port identifier.
lacppartnerkey	<Double>	Read-only	LACP Partner Key. The LACP key used by the partner port.
lacpactoraggregation	<String>	Read-only	The Aggregation flag indicates that the participant will allow

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
			the link to be used as part of an aggregate. Otherwise the link is to be used as an individual link, i.e. not aggregated with any other. Possible values = NS_EMPTY_STR, AGG
lacpactorinsync	<String>	Read-only	The Synchronization flag indicates that the transmitting participant.s mux component is in sync with the system id and key information transmitted. Possible values = NS_EMPTY_STR, SYNC
lacpactorcollecting	<String>	Read-only	The Collecting flag indicates that the participant.s collector, i.e. the reception component of the mux, is definitely on. If set the flag communicates collecting. Possible values = NS_EMPTY_STR, COLLECTING
lacpactordistributing	<String>	Read-only	The Distributing flag indicates that the participant.s

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
			distributor is not definitely off. If reset the flag indicates not distributing. Possible values = NS_EMPTY_STR, DISTRIBUTING
lacpportmuxstate	<String>	Read-only	LACP Port MUX state. The state of the MUX control machine. The Mux Control Machine attaches the physical port to an aggregate port, using the Selection Logic to choose an appropriate port, and turns the distributor and collector for the physical port on or off as required by protocol information. Possible values = DETACHED, WAITING, ATTACHED, COLLECTING, DISTRIBUTING
lacpportrxstat	<String>	Read-only	LACP Port RX state. The state of the Receive machine. The Receive Machine maintains partner information, recording protocol information from LACPDUs sent by

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
			remote partner(s). Received information is subject to a timeout, and if sufficient time elapses the receive machine will revert to using default partner information. Possible values = INIT, PORT_DISABLED, LACP_DISABLED, EXPIRED, DEFAULTED, CURRENT
lacpportselectstate	<String>	Read-only	LACP Port SELECT state. The state of the SELECT state machine, It could be SELECTED or UNSELECTED. Possible values = UNSELECTED, SELECTED, STANDBY
lldpmode	<String>	Read-only	Link Layer Discovery Protocol (LLDP) mode for an interface. The resultant LLDP mode of an interface depends on the LLDP mode configured at the global and the interface levels.

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
			Possible values = NONE, TRANSMITTER, RECEIVER, TRANSCEIVER
__count	<Double>	Read-only	count parameter

Operations

(click to see Properties)

ADD| DELETE| UPDATE| UNSET| GET (ALL)| GET| COUNT

Some options that you can use for each operations:

- **Getting warnings in response:**NITRO allows you to get warnings in an operation by specifying the "warning" query parameter as "yes". For example, to get warnings while connecting to the Citrix ADC appliance, the URL is as follows:

  http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/login?warning=yes

  If any, the warnings are displayed in the response payload with the HTTP code "209 X-NITRO-WARNING".

- **Authenticated access for individual NITRO operations:**NITRO allows you to logon to the Citrix ADC appliance to perform individual operations. You can use this option instead of creating a NITRO session (using the login object) and then using that session to perform all operations,

  To do this, you must specify the username and password in the request header of the NITRO request as follows:

  X-NITRO-USER:<username>

  X-NITRO-PASS:<password>

  Note: In such cases, make sure that the request header DOES not include the following:

  Cookie:NITRO_AUTH_TOKEN=<tokenvalue>

`*Note:`*

Mandatory parameters are marked in redand placeholder content is marked in <green>.

add

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/channel

HTTP Method:POST

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Content-Type:application/json

Request Payload:

```
{"channel":{

<b>"id":<String_value>,

</b>"ifnum":<String[]_value>,

"state":<String_value>,

"mode":<String_value>,

"conndistr":<String_value>,

"macdistr":<String_value>,

"lamac":<String_value>,

"speed":<String_value>,

"flowctl":<String_value>,

"hamonitor":<String_value>,

"haheartbeat":<String_value>,

"tagall":<String_value>,

"trunk":<String_value>,

"ifalias":<String_value>,
```

```
"throughput":<Double_value>,

"bandwidthhigh":<Double_value>,

"bandwidthnormal":<Double_value>

}}
```

Response:

HTTP Status Code on Success: 201 Created HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error

delete

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/channel/id_value<String>

HTTP Method:DELETE

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue>

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error

update

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/channel

HTTP Method:PUT

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Content-Type:application/json

Request Payload:

```
{"channel":{
<b>"id":<String_value>,
</b>"state":<String_value>,
"mode":<String_value>,
"conndistr":<String_value>,
"macdistr":<String_value>,
"lamac":<String_value>,
"speed":<String_value>,
"mtu":<Double_value>,
"flowctl":<String_value>,
"hamonitor":<String_value>,
"haheartbeat":<String_value>,
"tagall":<String_value>,
"trunk":<String_value>,
"ifalias":<String_value>,
"throughput":<Double_value>,
"lrminthroughput":<Double_value>,
"linkredundancy":<String_value>,
"bandwidthhigh":<Double_value>,
"bandwidthnormal":<Double_value>
}}
```

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error

unset

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/channel?**action=unset**

HTTP Method:POST

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Content-Type:application/json

Request Payload:

```
{"channel":{

<b>"id":<String_value>,

</b>"state":true,

"mode":true,

"conndistr":true,

"macdistr":true,

"speed":true,

"mtu":true,

"flowctl":true,

"hamonitor":true,

"haheartbeat":true,
```

```
"tagall":true,

"trunk":true,

"ifalias":true,

"throughput":true,

"lrminthroughput":true,

"linkredundancy":true,

"bandwidthhigh":true,

"bandwidthnormal":true

}}
```

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error

get (all)

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/channel

Query-parameters:

attrs

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/channel?**attrs=property-name1,property-name2**

Use this query parameter to specify the resource details that you want to retrieve.

filter

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/channel?**filter=property-name1:property-val1,property-name2:property-val2**

Use this query-parameter to get the filtered set of channel resources configured on Citrix ADC. Filtering can be done on any of the properties of the resource.

view

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/channel?**view=summary**

Note: By default, the retrieved results are displayed in detail view (?view=detail).

pagination

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/channel?**pagesize=#no;pageno=#no**

Use this query-parameter to get the channel resources in chunks.

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "channel": [ {

"id":<String_value>,

"devicename":<String_value>,

"unit":<Double_value>,

"description":<String_value>,

"flags":<Double_value>,
```

```
"mtu":<Double_value>,

"actualmtu":<Double_value>,

"vlan":<Double_value>,

"mac":<String_value>,

"lamac":<String_value>,

"uptime":<Double_value>,

"downtime":<Double_value>,

"reqmedia":<String_value>,

"reqspeed":<String_value>,

"reqduplex":<String_value>,

"reqflowcontrol":<String_value>,

"media":<String_value>,

"actspeed":<String_value>,

"duplex":<String_value>,

"actflowctl":<String_value>,

"mode":<String_value>,

"lamode":<String_value>,

"hamonitor":<String_value>,

"haheartbeat":<String_value>,

"state":<String_value>,

"autoneg":<Double_value>,

"autonegresult":<Double_value>,
```

```
"tagged":<Double_value>,

"tagall":<String_value>,

"trunk":<String_value>,

"taggedany":<Double_value>,

"taggedautolearn":<Double_value>,

"hangdetect":<Double_value>,

"hangreset":<Double_value>,

"linkstate":<Double_value>,

"intfstate":<Double_value>,

"rxpackets":<Double_value>,

"rxbytes":<Double_value>,

"rxerrors":<Double_value>,

"rxdrops":<Double_value>,

"txpackets":<Double_value>,

"txbytes":<Double_value>,

"txerrors":<Double_value>,

"txdrops":<Double_value>,

"indisc":<Double_value>,

"outdisc":<Double_value>,

"fctls":<Double_value>,

"hangs":<Double_value>,

"stsstalls":<Double_value>,
```

```
"txstalls":<Double_value>,

"rxstalls":<Double_value>,

"bdgmuted":<Double_value>,

"vmac":<String_value>,

"vmac6":<String_value>,

"ifalias":<String_value>,

"reqthroughput":<Double_value>,

"lrminthroughput":<Double_value>,

"linkredundancy":<String_value>,

"actthroughput":<Double_value>,

"bandwidthhigh":<Double_value>,

"bandwidthnormal":<Double_value>,

"ifnum":<String[]_value>,

"backplane":<String_value>,

"cleartime":<Double_value>,

"lacpmode":<String_value>,

"lacptimeout":<String_value>,

"lacpactorpriority":<Double_value>,

"lacpactorportno":<Double_value>,

"lacppartnerstate":<String_value>,

"lacppartnertimeout":<String_value>,

"lacppartneraggregation":<String_value>,
```

```
"lacppartnerinsync":<String_value>,

"lacppartnercollecting":<String_value>,

"lacppartnerdistributing":<String_value>,

"lacppartnerdefaulted":<String_value>,

"lacppartnerexpired":<String_value>,

"lacppartnerpriority":<Double_value>,

"lacppartnersystemmac":<String_value>,

"lacppartnersystempriority":<Double_value>,

"lacppartnerportno":<Double_value>,

"lacppartnerkey":<Double_value>,

"lacpactoraggregation":<String_value>,

"lacpactorinsync":<String_value>,

"lacpactorcollecting":<String_value>,

"lacpactordistributing":<String_value>,

"lacpportmuxstate":<String_value>,

"lacpportrxstat":<String_value>,

"lacpportselectstate":<String_value>,

"lldpmode":<String_value>

}]}
```

get

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/channel/id_value<String>

Query-parameters:

attrs

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/channel/id_value<String>?**attrs=property-name1,property-name2**

Use this query parameter to specify the resource details that you want to retrieve.

view

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/channel/id_value<String>?**view=summary**

Note: By default, the retrieved results are displayed in detail view (?view=detail).

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "channel": [ {

"id":<String_value>,

"devicename":<String_value>,

"unit":<Double_value>,

"description":<String_value>,

"flags":<Double_value>,
```

```
"mtu":<Double_value>,

"actualmtu":<Double_value>,

"vlan":<Double_value>,

"mac":<String_value>,

"lamac":<String_value>,

"uptime":<Double_value>,

"downtime":<Double_value>,

"reqmedia":<String_value>,

"reqspeed":<String_value>,

"reqduplex":<String_value>,

"reqflowcontrol":<String_value>,

"media":<String_value>,

"actspeed":<String_value>,

"duplex":<String_value>,

"actflowctl":<String_value>,

"mode":<String_value>,

"lamode":<String_value>,

"hamonitor":<String_value>,

"haheartbeat":<String_value>,

"state":<String_value>,

"autoneg":<Double_value>,

"autonegresult":<Double_value>,
```

```
"tagged":<Double_value>,

"tagall":<String_value>,

"trunk":<String_value>,

"taggedany":<Double_value>,

"taggedautolearn":<Double_value>,

"hangdetect":<Double_value>,

"hangreset":<Double_value>,

"linkstate":<Double_value>,

"intfstate":<Double_value>,

"rxpackets":<Double_value>,

"rxbytes":<Double_value>,

"rxerrors":<Double_value>,

"rxdrops":<Double_value>,

"txpackets":<Double_value>,

"txbytes":<Double_value>,

"txerrors":<Double_value>,

"txdrops":<Double_value>,

"indisc":<Double_value>,

"outdisc":<Double_value>,

"fctls":<Double_value>,

"hangs":<Double_value>,

"stsstalls":<Double_value>,
```

```
"txstalls":<Double_value>,

"rxstalls":<Double_value>,

"bdgmuted":<Double_value>,

"vmac":<String_value>,

"vmac6":<String_value>,

"ifalias":<String_value>,

"reqthroughput":<Double_value>,

"lrminthroughput":<Double_value>,

"linkredundancy":<String_value>,

"actthroughput":<Double_value>,

"bandwidthhigh":<Double_value>,

"bandwidthnormal":<Double_value>,

"ifnum":<String[]_value>,

"backplane":<String_value>,

"cleartime":<Double_value>,

"lacpmode":<String_value>,

"lacptimeout":<String_value>,

"lacpactorpriority":<Double_value>,

"lacpactorportno":<Double_value>,

"lacppartnerstate":<String_value>,

"lacppartnertimeout":<String_value>,

"lacppartneraggregation":<String_value>,
```

```
"lacppartnerinsync":<String_value>,

"lacppartnercollecting":<String_value>,

"lacppartnerdistributing":<String_value>,

"lacppartnerdefaulted":<String_value>,

"lacppartnerexpired":<String_value>,

"lacppartnerpriority":<Double_value>,

"lacppartnersystemmac":<String_value>,

"lacppartnersystempriority":<Double_value>,

"lacppartnerportno":<Double_value>,

"lacppartnerkey":<Double_value>,

"lacpactoraggregation":<String_value>,

"lacpactorinsync":<String_value>,

"lacpactorcollecting":<String_value>,

"lacpactordistributing":<String_value>,

"lacpportmuxstate":<String_value>,

"lacpportrxstat":<String_value>,

"lacpportselectstate":<String_value>,

"lldpmode":<String_value>

}]}
```

count

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/channel?**count=yes**

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "channel": [ { "__count": "#no"} ] }
```

channel_binding

Binding object which returns the resources bound to channel.

Properties

(click to see Operations)

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
id	<String>	Read-write	ID of an LA channel or LA channel in cluster configuration whose details you want the Citrix ADC to display. Specify an LA channel in LA/x notation, where x can range from 1 to 8 or a cluster LA channel in CLA/x

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
			notation or Link redundant channel in LR/x notation , where x can range from 1 to 4. Minimum length = 1
channel_interface_binding	<channel_interface_binding[]>	Read-only	interface that can be bound to channel.

Operations

(click to see Properties)

GET| GET (ALL)

Some options that you can use for each operations:

- **Getting warnings in response:**NITRO allows you to get warnings in an operation by specifying the "warning" query parameter as "yes". For example, to get warnings while connecting to the Citrix ADC appliance, the URL is as follows:

  http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/login?warning=yes

  If any, the warnings are displayed in the response payload with the HTTP code "209 X-NITRO-WARNING".

- **Authenticated access for individual NITRO operations:**NITRO allows you to logon to the Citrix ADC appliance to perform individual operations. You can use this option instead of creating a NITRO session (using the login object) and then using that session to perform all operations,

  To do this, you must specify the username and password in the request header of the NITRO request as follows:

  X-NITRO-USER:<username>

  X-NITRO-PASS:<password>

  Note: In such cases, make sure that the request header DOES not include the following:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue>

`*Note:`*

Mandatory parameters are marked in redand placeholder content is marked in <green>.

get

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/channel_binding/id_value<String>

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "channel_binding": [ {

"id":<String_value>,

"channel_interface_binding":<channel_interface_binding[]_value>

}]}
```

get (all)

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/channel_binding

Query-parameters:

bulkbindings

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/channel_binding?**bulkbindings=yes**

NITRO allows you to fetch bindings in bulk.

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "channel_binding": [ {

"id":<String_value>,

"channel_interface_binding":<channel_interface_binding[]_value>

}]}
```

channel_interface_binding

Binding object showing the interface that can be bound to channel.

Properties

(click to see Operations)

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
svmcmd	<Double>	Read-write	New attribute added to identify the source of cmd, when SVM fires the nitro cmd, it

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
			will set the value of SVMCMD to be 1. . Default value: 0
ifnum	<String[]>	Read-write	Interfaces to be bound to the LA channel of a Citrix ADC or to the LA channel of a cluster configuration. For an LA channel of a Citrix ADC, specify an interface in C/U notation (for example, 1/3). For an LA channel of a cluster configuration, specify an interface in N/C/U notation (for example, 2/1/3). where C can take one of the following values: * 0 - Indicates a management interface. * 1 - Indicates a 1 Gbps port. * 10 - Indicates a 10 Gbps port. U is a unique integer for representing an interface in a particular port group. N is the ID of the node to which an interface belongs in a cluster configuration. Use spaces to separate multiple entries.

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
id	<String>	Read-write	ID of the LA channel or the cluster LA channel to which you want to bind interfaces. Specify an LA channel in LA/x notation, where x can range from 1 to 8 or a cluster LA channel in CLA/x notation or Link redundant channel in LR/x notation , where x can range from 1 to 4.
lamode	<String>	Read-only	The mode(AUTO/MANNUAL) for the LA channel. Possible values = MANUAL, AUTO
slavespeed	<Double>	Read-only	Speed of the member interfaces.
slavemedia	<Double>	Read-only	Media type of the member interfaces.
slaveduplex	<Double>	Read-only	Duplex of the member interfaces.
lractiveintf	<Boolean>	Read-only	LR set member interface state(active/inactive).
slaveflowctl	<Double>	Read-only	Flowcontrol of the member interfaces.
slavetime	<Double>	Read-only	UP time of the member interfaces.

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
slavestate	<Double>	Read-only	State of the member interfaces.
__count	<Double>	Read-only	count parameter

Operations

(click to see Properties)

ADD:| DELETE:| GET| GET (ALL)| COUNT

Some options that you can use for each operations:

- **Getting warnings in response:**NITRO allows you to get warnings in an operation by specifying the "warning" query parameter as "yes". For example, to get warnings while connecting to the Citrix ADC appliance, the URL is as follows:

  http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/login?warning=yes

  If any, the warnings are displayed in the response payload with the HTTP code "209 X-NITRO-WARNING".

- **Authenticated access for individual NITRO operations:**NITRO allows you to logon to the Citrix ADC appliance to perform individual operations. You can use this option instead of creating a NITRO session (using the login object) and then using that session to perform all operations,

  To do this, you must specify the username and password in the request header of the NITRO request as follows:

  X-NITRO-USER:<username>

  X-NITRO-PASS:<password>

  Note: In such cases, make sure that the request header DOES not include the following:

  Cookie:NITRO_AUTH_TOKEN=<tokenvalue>

`*Note:`*

Mandatory parameters are marked in redand placeholder content is marked in <green>.

add

URL:http://<Citrix-ADC-IP-address(NSIP)/nitro/v1/config/channel_interface_binding

HTTP Method:PUT

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Content-Type:application/json

Request Payload:

```
{

"channel_interface_binding":{

<b>"id":<String_value>,

</b><b>"ifnum":<String[]_value></b>

}}
```

Response:

HTTP Status Code on Success: 201 Created HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error

delete

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/channel_interface_binding/id_value<String>

Query-parameters:

args

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/channel_interface_binding/id_value<String>?**args=ifnum:<String_value>**

HTTP Method:DELETE

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue>

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error

get

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/channel_interface_binding/id_value<String>

Query-parameters:

filter

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/channel_interface_binding/id_value<String>?**filter=property-name1:property-value1,property-name2:property-value2**

Use this query-parameter to get the filtered set of channel_interface_binding resources configured on Citrix ADC. Filtering can be done on any of the properties of the resource.

pagination

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/channel_interface_binding/id_value<String>?**pagesize=#no;pageno=#no**

Use this query-parameter to get the channel_interface_binding resources in chunks.

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "channel_interface_binding": [ {

"svmcmd":<Double_value>,

"ifnum":<String[]_value>,

"id":<String_value>,

"lamode":<String_value>,

"slavespeed":<Double_value>,

"slavemedia":<Double_value>,

"slaveduplex":<Double_value>,

"lractiveintf":<Boolean_value>,

"slaveflowctl":<Double_value>,

"slavetime":<Double_value>,

"slavestate":<Double_value>

}]}
```

get (all)

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/channel_interface_binding

Query-parameters:

bulkbindings

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/channel_interface_binding?
bulkbindings=yes

NITRO allows you to fetch bindings in bulk.

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "channel_interface_binding": [ {

"svmcmd":<Double_value>,

"ifnum":<String[]_value>,

"id":<String_value>,

"lamode":<String_value>,

"slavespeed":<Double_value>,

"slavemedia":<Double_value>,

"slaveduplex":<Double_value>,

"lractiveintf":<Boolean_value>,

"slaveflowctl":<Double_value>,

"slavetime":<Double_value>,

"slavestate":<Double_value>

}]}
```

count

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/channel_interface_binding/id_value<String>?**count=yes**

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{"channel_interface_binding": [ { "__count": "#no"} ] }
```

ci

Configuration for "CI" resource.

Properties

(click to see Operations)

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
ifaces	<String>	Read-only	Interfaces that are critical for the appliance to operate in high availability mode.

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
__count	<Double>	Read-only	count parameter

Operations

(click to see Properties)

GET (ALL)| COUNT

Some options that you can use for each operations:

- **Getting warnings in response:**NITRO allows you to get warnings in an operation by specifying the "warning" query parameter as "yes". For example, to get warnings while connecting to the Citrix ADC appliance, the URL is as follows:

  http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/login?warning=yes

  If any, the warnings are displayed in the response payload with the HTTP code "209 X-NITRO-WARNING".
- **Authenticated access for individual NITRO operations:**NITRO allows you to logon to the Citrix ADC appliance to perform individual operations. You can use this option instead of creating a NITRO session (using the login object) and then using that session to perform all operations,

  To do this, you must specify the username and password in the request header of the NITRO request as follows:

  X-NITRO-USER:<username>

  X-NITRO-PASS:<password>

  Note: In such cases, make sure that the request header DOES not include the following:

  Cookie:NITRO_AUTH_TOKEN=<tokenvalue>

`*Note:`*

Mandatory parameters are marked in redand placeholder content is marked in <green>.

get (all)

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/ci

Query-parameters:

attrs

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/ci?**attrs=property-name1,property-name2**

Use this query parameter to specify the resource details that you want to retrieve.

filter

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/ci?**filter=property-name1:property-val1,property-name2:property-val2**

Use this query-parameter to get the filtered set of ci resources configured on Citrix ADC. Filtering can be done on any of the properties of the resource.

view

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/ci?**view=summary**

Note: By default, the retrieved results are displayed in detail view (?view=detail).

pagination

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/ci?**pagesize=#no;pageno=#no**

Use this query-parameter to get the ci resources in chunks.

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "ci": [ {
```

```
"ifaces":<String_value>

}]}
```

count

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/ci?**count=yes**

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "ci": [ { "__count": "#no"} ] }
```

fis

Configuration for "FIS" resource.

Properties

(click to see Operations)

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
name	<String>	Read-write	Name for the FIS to be created. Leading character must be a

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
			number or letter. Other characters allowed, after the first character, are @ _ - . (period) : (colon) # and space ( ). Note: In a cluster setup, the FIS name on each node must be unique. Minimum length = 1
ownernode	<Double>	Read-write	ID of the cluster node for which you are creating the FIS. Can be configured only through the cluster IP address. Minimum value = 0 Maximum value = 31
ifaces	<String>	Read-only	Interfaces to be bound to the FIS, in slot/port notation (for example, 1/3).
__count	<Double>	Read-only	count parameter

Operations

(click to see Properties)

ADD| DELETE| GET (ALL)| GET| COUNT

Some options that you can use for each operations:

- **Getting warnings in response:**NITRO allows you to get warnings in an operation by specifying the "warning" query parameter as "yes". For example, to get warnings while connecting to the Citrix ADC appliance, the URL is as follows:

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/login?warning=yes

If any, the warnings are displayed in the response payload with the HTTP code "209 X-NITRO-WARNING".

- **Authenticated access for individual NITRO operations:**NITRO allows you to logon to the Citrix ADC appliance to perform individual operations. You can use this option instead of creating a NITRO session (using the login object) and then using that session to perform all operations,

  To do this, you must specify the username and password in the request header of the NITRO request as follows:

  X-NITRO-USER:<username>

  X-NITRO-PASS:<password>

  Note: In such cases, make sure that the request header DOES not include the following:

  Cookie:NITRO_AUTH_TOKEN=<tokenvalue>

`*Note:`*

Mandatory parameters are marked in redand placeholder content is marked in <green>.

add

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/fis

HTTP Method:POST

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Content-Type:application/json

Request Payload:

```
{"fis":{

<b>"name":<String_value>,

</b>"ownernode":<Double_value>

}}
```

Response:

HTTP Status Code on Success: 201 Created HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error

delete

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/fis/name_value<String>

HTTP Method:DELETE

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue>

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error

get (all)

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/fis

Query-parameters:

attrs

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/fis?**attrs=property-name1,property-name2**

Use this query parameter to specify the resource details that you want to retrieve.

filter

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/fis?**filter=property-name1:property-val1,property-name2:property-val2**

Use this query-parameter to get the filtered set of fis resources configured on Citrix ADC. Filtering can be done on any of the properties of the resource.

view

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/fis?**view=summary**

Note: By default, the retrieved results are displayed in detail view (?view=detail).

pagination

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/fis?**pagesize=#no;pageno=#no**

Use this query-parameter to get the fis resources in chunks.

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "fis": [ {

"name":<String_value>,

"ifaces":<String_value>,

"ownernode":<Double_value>

}]}
```

get

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/fis/name_value<String>

Query-parameters:

attrs

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/fis/name_value<String>?**attrs=property-name1,property-name2**

Use this query parameter to specify the resource details that you want to retrieve.

view

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/fis/name_value<String>?**view=summary**

Note: By default, the retrieved results are displayed in detail view (?view=detail).

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "fis": [ {

"name":<String_value>,

"ifaces":<String_value>,

"ownernode":<Double_value>

}]}
```

count

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/fis?**count=yes**

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "fis": [ { "__count": "#no"} ] }
```

fis_binding

Binding object which returns the resources bound to fis.

Properties

(click to see Operations)

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
name	<String>	Read-write	The name of the FIS configured on the appliance. . Minimum length = 1
fis_channel_binding	<fis_channel_binding []>	Read-only	channel that can be bound to fis.

Operations

(click to see Properties)

GET| GET (ALL)

Some options that you can use for each operations:

- **Getting warnings in response:**NITRO allows you to get warnings in an operation by specifying the "warning" query parameter as "yes". For example, to get warnings while connecting to the Citrix ADC appliance, the URL is as follows:

  http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/login?warning=yes

  If any, the warnings are displayed in the response payload with the HTTP code "209 X-NITRO-WARNING".

- **Authenticated access for individual NITRO operations:**NITRO allows you to logon to the Citrix ADC appliance to perform individual operations. You can use this option instead of creating a NITRO session (using the login object) and then using that session to perform all operations,

  To do this, you must specify the username and password in the request header of the NITRO request as follows:

  X-NITRO-USER:<username>

  X-NITRO-PASS:<password>

  Note: In such cases, make sure that the request header DOES not include the following:

  Cookie:NITRO_AUTH_TOKEN=<tokenvalue>

`*Note:`*

Mandatory parameters are marked in redand placeholder content is marked in <green>.

get

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/fis_binding/name_value<String>

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "fis_binding": [ {

"name":<String_value>,

"fis_channel_binding":<fis_channel_binding[]_value>

}]}
```

get (all)

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/fis_binding

Query-parameters:

bulkbindings

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/fis_binding?**bulkbindings=yes**

NITRO allows you to fetch bindings in bulk.

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "fis_binding": [ {

"name":<String_value>,

"fis_channel_binding":<fis_channel_binding[]_value>

}]}
```

fis_channel_binding

Binding object showing the channel that can be bound to fis.

Properties

(click to see Operations)

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
ownernode	<Double>	Read-write	ID of the cluster node for which you are creating the FIS. Can be configured only through the cluster IP address. Minimum value = 0 Maximum value = 31
name	<String>	Read-write	The name of the FIS to which you want to bind interfaces. Minimum length = 1
ifnum	<String>	Read-write	Interface to be bound to the FIS, specified in slot/port notation (for example, 1/3).

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
__count	<Double>	Read-write	count parameter

Operations

(click to see Properties)

ADD:| DELETE:| GET| GET (ALL)| COUNT

Some options that you can use for each operations:

- **Getting warnings in response:**NITRO allows you to get warnings in an operation by specifying the "warning" query parameter as "yes". For example, to get warnings while connecting to the Citrix ADC appliance, the URL is as follows:

  http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/login?warning=yes

  If any, the warnings are displayed in the response payload with the HTTP code "209 X-NITRO-WARNING".
- **Authenticated access for individual NITRO operations:**NITRO allows you to logon to the Citrix ADC appliance to perform individual operations. You can use this option instead of creating a NITRO session (using the login object) and then using that session to perform all operations,

  To do this, you must specify the username and password in the request header of the NITRO request as follows:

  X-NITRO-USER:<username>

  X-NITRO-PASS:<password>

  Note: In such cases, make sure that the request header DOES not include the following:

  Cookie:NITRO_AUTH_TOKEN=<tokenvalue>

`*Note:`*

Mandatory parameters are marked in redand placeholder content is marked in <green>.

add

URL:http://<Citrix-ADC-IP-address(NSIP)/nitro/v1/config/fis_channel_binding

HTTP Method:PUT

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Content-Type:application/json

Request Payload:

```
{

"fis_channel_binding":{

<b>"name":<String_value>,

</b><b>"ifnum":<String_value></b>

}}
```

Response:

HTTP Status Code on Success: 201 Created HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error

delete

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/fis_channel_binding/name_value<String>

Query-parameters:

args

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/fis_channel_binding/name_value<String>?**args=ifnum:<String_value>**

HTTP Method:DELETE

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue>

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error

get

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/fis_channel_binding/name_value<String>

Query-parameters:

filter

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/fis_channel_binding/name_value<String>?**filter=property-name1:property-value1,property-name2:property-value2**

Use this query-parameter to get the filtered set of fis_channel_binding resources configured on Citrix ADC. Filtering can be done on any of the properties of the resource.

pagination

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/fis_channel_binding/name_value<String>?**pagesize=#no;pageno=#no**

Use this query-parameter to get the fis_channel_binding resources in chunks.

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "fis_channel_binding": [ {
```

```
"ownernode":<Double_value>,

"name":<String_value>,

"ifnum":<String_value>

}]}
```

get (all)

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/fis_channel_binding

Query-parameters:

bulkbindings

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/fis_channel_binding?**bulkbindings=yes**

NITRO allows you to fetch bindings in bulk.

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "fis_channel_binding": [ {

"ownernode":<Double_value>,

"name":<String_value>,
```

```
"ifnum":<String_value>

}]}
```

count

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/fis_channel_binding/name_value<String>?**count=yes**

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{"fis_channel_binding": [ { "__count": "#no"} ] }
```

fis_interface_binding

Binding object showing the interface that can be bound to fis.

Properties

(click to see Operations)

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
ownernode	<Double>	Read-write	ID of the cluster node for which you

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
			are creating the FIS. Can be configured only through the cluster IP address. Minimum value = 0 Maximum value = 31
name	<String>	Read-write	The name of the FIS to which you want to bind interfaces. Minimum length = 1
ifnum	<String>	Read-write	Interface to be bound to the FIS, specified in slot/port notation (for example, 1/3).

Operations

(click to see Properties)

ADD:| DELETE:

Some options that you can use for each operations:

- **Getting warnings in response:**NITRO allows you to get warnings in an operation by specifying the "warning" query parameter as "yes". For example, to get warnings while connecting to the Citrix ADC appliance, the URL is as follows:

  http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/login?warning=yes

  If any, the warnings are displayed in the response payload with the HTTP code "209 X-NITRO-WARNING".
- **Authenticated access for individual NITRO operations:**NITRO allows you to logon to the Citrix ADC appliance to perform individual operations. You can use this option instead of creating a NITRO session (using the login object) and then using that session to perform all operations,

To do this, you must specify the username and password in the request header of the NITRO request as follows:

X-NITRO-USER:<username>

X-NITRO-PASS:<password>

Note: In such cases, make sure that the request header DOES not include the following:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue>

`*Note:`*

Mandatory parameters are marked in redand placeholder content is marked in <green>.

add

URL:http://<Citrix-ADC-IP-address(NSIP)/nitro/v1/config/fis_interface_binding

HTTP Method:PUT

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Content-Type:application/json

Request Payload:

```
{

"fis_interface_binding":{

<b>"name":<String_value>,

</b><b>"ifnum":<String_value></b>

}}
```

Response:

HTTP Status Code on Success: 201 Created HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error

delete

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/fis_interface_binding/name_value<String>

Query-parameters:

args

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/fis_interface_binding/name_value<String>?**args=ifnum:<String_value>**

HTTP Method:DELETE

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue>

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error

forwardingsession

Configuration for session forward resource.

Properties

(click to see Operations)

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
name	<String>	Read-write	Name for the forwarding session rule. Can begin with a letter, number, or the underscore character (_), and can consist of letters, numbers, and the hyphen (-), period (.) pound (#), space ( ), at (@), equals (=),

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
			colon (:), and underscore characters. Cannot be changed after the rule is created. The following requirement applies only to the Citrix ADC CLI: If the name includes one or more spaces, enclose the name in double or single quotation marks (for example, "my rule" or 'my rule'). Minimum length = 1
network	<String>	Read-write	An IPv4 network address or IPv6 prefix of a network from which the forwarded traffic originates or to which it is destined. Minimum length = 1
netmask	<String>	Read-write	Subnet mask associated with the network. Minimum length = 1
acl6name	<String>	Read-write	Name of any configured ACL6 whose action is ALLOW. The rule of the ACL6 is used as

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
			a forwarding session rule. Minimum length = 1
aclname	<String>	Read-write	Name of any configured ACL whose action is ALLOW. The rule of the ACL is used as a forwarding session rule. Minimum length = 1
td	<Double>	Read-write	Integer value that uniquely identifies the traffic domain in which you want to configure the entity. If you do not specify an ID, the entity becomes part of the default traffic domain, which has an ID of 0. Minimum value = 0 Maximum value = 4094
connfailover	<String>	Read-write	Synchronize connection information with the secondary appliance in a high availability (HA) pair. That is, synchronize all connection-related information for the forwarding session.

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
			Default value: DISABLED Possible values = ENABLED, DISABLED
sourceroutecache	<String>	Read-write	Cache the source ip address and mac address of the DA servers. Default value: DISABLED Possible values = ENABLED, DISABLED
processlocal	<String>	Read-write	Enabling this option on forwarding session will not steer the packet to flow processor. Instead, packet will be routed. Default value: DISABLED Possible values = ENABLED, DISABLED
__count	<Double>	Read-only	count parameter

Operations

(click to see Properties)

ADD| UPDATE| DELETE| GET (ALL)| GET| COUNT

Some options that you can use for each operations:

- **Getting warnings in response:**NITRO allows you to get warnings in an operation by specifying the "warning" query parameter as "yes". For example, to get warnings while connecting to the Citrix ADC appliance, the URL is as follows:

  http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/login?warning=yes

  If any, the warnings are displayed in the response payload with the HTTP code "209 X-NITRO-WARNING".

- **Authenticated access for individual NITRO operations:**NITRO allows you to logon to the Citrix ADC appliance to perform individual operations. You can use this option instead of creating a NITRO session (using the login object) and then using that session to perform all operations,

  To do this, you must specify the username and password in the request header of the NITRO request as follows:

  X-NITRO-USER:<username>

  X-NITRO-PASS:<password>

  Note: In such cases, make sure that the request header DOES not include the following:

  Cookie:NITRO_AUTH_TOKEN=<tokenvalue>

`*Note:`*

Mandatory parameters are marked in redand placeholder content is marked in <green>.

add

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/forwardingsession

HTTP Method:POST

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Content-Type:application/json

Request Payload:

```
{"forwardingsession":{
```

```
<b>"name":<String_value>,

</b>"network":<String_value>,

"netmask":<String_value>,

"acl6name":<String_value>,

"aclname":<String_value>,

"td":<Double_value>,

"connfailover":<String_value>,

"sourceroutecache":<String_value>,

"processlocal":<String_value>

}}
```

Response:

HTTP Status Code on Success: 201 Created HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error

update

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/forwardingsession

HTTP Method:PUT

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Content-Type:application/json

Request Payload:

```
{"forwardingsession":{
```

```
<b>"name":<String_value>,

</b>"connfailover":<String_value>,

"sourceroutecache":<String_value>,

"processlocal":<String_value>,

"acl6name":<String_value>,

"aclname":<String_value>

}}
```

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error

delete

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/forwardingsession/name_value<String>

HTTP Method:DELETE

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue>

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error

get (all)

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/forwardingsession

Query-parameters:

attrs

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/forwardingsession?**attrs=property-name1,property-name2**

Use this query parameter to specify the resource details that you want to retrieve.

filter

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/forwardingsession?**filter=property-name1:property-val1,property-name2:property-val2**

Use this query-parameter to get the filtered set of forwardingsession resources configured on Citrix ADC. Filtering can be done on any of the properties of the resource.

view

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/forwardingsession?**view=summary**

Note: By default, the retrieved results are displayed in detail view (?view=detail).

pagination

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/forwardingsession?**pagesize=#no;pageno=#no**

Use this query-parameter to get the forwardingsession resources in chunks.

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "forwardingsession": [ {
```

```
"name":<String_value>,

"network":<String_value>,

"netmask":<String_value>,

"td":<Double_value>,

"aclname":<String_value>,

"acl6name":<String_value>,

"connfailover":<String_value>,

"sourceroutecache":<String_value>,

"processlocal":<String_value>

}]}
```

get

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/forwardingsession/name_value<String>

Query-parameters:

attrs

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/forwardingsession/name_value<String>?**attrs=property-name1,property-name2**

Use this query parameter to specify the resource details that you want to retrieve.

view

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/forwardingsession/name_value<String>?**view=summary**

Note: By default, the retrieved results are displayed in detail view (?view=detail).

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "forwardingsession": [ {

"name":<String_value>,

"network":<String_value>,

"netmask":<String_value>,

"td":<Double_value>,

"aclname":<String_value>,

"acl6name":<String_value>,

"connfailover":<String_value>,

"sourceroutecache":<String_value>,

"processlocal":<String_value>

}]}
```

count

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/forwardingsession?**count=yes**

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "forwardingsession": [ { "__count": "#no"} ] }
```

inat

Configuration for inbound nat resource.

Properties

(click to see Operations)

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
name	<String>	Read-write	Name for the Inbound NAT (INAT) entry. Leading character must be a number or letter. Other characters allowed, after the first character, are @ _ - . (period) : (colon) # and space ( ). Minimum length = 1
publicip	<String>	Read-write	Public IP address of packets received on the Citrix ADC. Can be aNetScaler-

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
			owned VIP or VIP6 address. Minimum length = 1
privateip	<String>	Read-write	IP address of the server to which the packet is sent by the Citrix ADC. Can be an IPv4 or IPv6 address. Minimum length = 1
mode	<String>	Read-write	Stateless translation. Possible values = STATELESS
tcpproxy	<String>	Read-write	Enable TCP proxy, which enables the Citrix ADC to optimize the RNAT TCP traffic by using Layer 4 features. Default value: DISABLED Possible values = ENABLED, DISABLED
ftp	<String>	Read-write	Enable the FTP protocol on the server for transferring files between the client and the server. Default value: DISABLED

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
			Possible values = ENABLED, DISABLED
tftp	<String>	Read-write	To enable/disable TFTP (Default DISABLED). Default value: DISABLED Possible values = ENABLED, DISABLED
usip	<String>	Read-write	Enable the Citrix ADC to retain the source IP address of packets before sending the packets to the server. Possible values = ON, OFF
usnip	<String>	Read-write	Enable the Citrix ADC to use a SNIP address as the source IP address of packets before sending the packets to the server. Possible values = ON, OFF
proxyip	<String>	Read-write	Unique IP address used as the source IP address in packets sent to the server. Must be a MIP or SNIP address.

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
useproxyport	<String>	Read-write	Enable the Citrix ADC to proxy the source port of packets before sending the packets to the server. Default value: ENABLED Possible values = ENABLED, DISABLED
td	<Double>	Read-write	Integer value that uniquely identifies the traffic domain in which you want to configure the entity. If you do not specify an ID, the entity becomes part of the default traffic domain, which has an ID of 0. Minimum value = 0 Maximum value = 4094
flags	<Double>	Read-only	Flags for different modes.
__count	<Double>	Read-only	count parameter

Operations

(click to see Properties)

ADD| DELETE| UPDATE| UNSET| GET (ALL)| GET| COUNT

Some options that you can use for each operations:

- **Getting warnings in response:**NITRO allows you to get warnings in an operation by specifying the "warning" query parameter as "yes". For example, to get warnings while connecting to the Citrix ADC appliance, the URL is as follows:

  http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/login?warning=yes

  If any, the warnings are displayed in the response payload with the HTTP code "209 X-NITRO-WARNING".

- **Authenticated access for individual NITRO operations:**NITRO allows you to logon to the Citrix ADC appliance to perform individual operations. You can use this option instead of creating a NITRO session (using the login object) and then using that session to perform all operations,

  To do this, you must specify the username and password in the request header of the NITRO request as follows:

  X-NITRO-USER:<username>

  X-NITRO-PASS:<password>

  Note: In such cases, make sure that the request header DOES not include the following:

  Cookie:NITRO_AUTH_TOKEN=<tokenvalue>

`*Note:`*

Mandatory parameters are marked in redand placeholder content is marked in <green>.

add

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/inat

HTTP Method:POST

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Content-Type:application/json

Request Payload:

```
{"inat":{
```

```
<b>"name":<String_value>,

</b><b>"publicip":<String_value>,

</b><b>"privateip":<String_value>,

</b>"mode":<String_value>,

"tcpproxy":<String_value>,

"ftp":<String_value>,

"tftp":<String_value>,

"usip":<String_value>,

"usnip":<String_value>,

"proxyip":<String_value>,

"useproxyport":<String_value>,

"td":<Double_value>

}}
```

Response:

HTTP Status Code on Success: 201 Created HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error

delete

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/inat/name_value<String>

HTTP Method:DELETE

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue>

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error

update

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/inat

HTTP Method:PUT

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Content-Type:application/json

Request Payload:

```
{"inat":{

<b>"name":<String_value>,

</b>"privateip":<String_value>,

"tcpproxy":<String_value>,

"ftp":<String_value>,

"tftp":<String_value>,

"usip":<String_value>,

"usnip":<String_value>,

"proxyip":<String_value>,

"useproxyport":<String_value>,

"mode":<String_value>

}}
```

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error

unset

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/inat?**action=unset**

HTTP Method:POST

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Content-Type:application/json

Request Payload:

```
{"inat":{

<b>"name":<String_value>,

</b>"tcpproxy":true,

"ftp":true,

"tftp":true,

"usip":true,

"usnip":true,

"proxyip":true,

"useproxyport":true,

"mode":true

}}
```

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error

get (all)

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/inat

Query-parameters:

attrs

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/inat?**attrs=property-name1,property-name2**

Use this query parameter to specify the resource details that you want to retrieve.

filter

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/inat?**filter=property-name1:property-val1,property-name2:property-val2**

Use this query-parameter to get the filtered set of inat resources configured on Citrix ADC. Filtering can be done on any of the properties of the resource.

view

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/inat?**view=summary**

Note: By default, the retrieved results are displayed in detail view (?view=detail).

pagination

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/inat?**pagesize=#no;pageno=#no**

Use this query-parameter to get the inat resources in chunks.

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "inat": [ {

"name":<String_value>,

"publicip":<String_value>,

"privateip":<String_value>,

"proxyip":<String_value>,

"tcpproxy":<String_value>,

"ftp":<String_value>,

"tftp":<String_value>,

"usip":<String_value>,

"usnip":<String_value>,

"useproxyport":<String_value>,

"flags":<Double_value>,

"mode":<String_value>,

"td":<Double_value>

}]}
```

get

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/inat/name_value<String>

Query-parameters:

attrs

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/inat/name_value<String>?**attrs=property-name1,property-name2**

Use this query parameter to specify the resource details that you want to retrieve.

view

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/inat/name_value<String>?**view=summary**

Note: By default, the retrieved results are displayed in detail view (?view=detail).

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "inat": [ {

"name":<String_value>,

"publicip":<String_value>,

"privateip":<String_value>,

"proxyip":<String_value>,
```

```
"tcpproxy":<String_value>,

"ftp":<String_value>,

"tftp":<String_value>,

"usip":<String_value>,

"usnip":<String_value>,

"useproxyport":<String_value>,

"flags":<Double_value>,

"mode":<String_value>,

"td":<Double_value>

}]}
```

count

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/inat?**count=yes**

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "inat": [ { "__count": "#no"} ] }
```

inatparam

Configuration for INAT parameter resource.

Properties

(click to see Operations)

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
nat46v6prefix	<String>	Read-write	The prefix used for translating packets received from private IPv6 servers into IPv4 packets. This prefix has a length of 96 bits (128-32 = 96). The IPv6 servers embed the destination IP address of the IPv4 servers or hosts in the last 32 bits of the destination IP address field of the IPv6 packets. The first 96 bits of the destination IP address field are set as the IPv6 NAT prefix. IPv6 packets addressed to this prefix have to be routed to the Citrix ADC to ensure that the IPv6-IPv4

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
			translation is done by the appliance.
td	<Double>	Read-write	Integer value that uniquely identifies the traffic domain in which you want to configure the entity. If you do not specify an ID, the entity becomes part of the default traffic domain, which has an ID of 0. Default value: 0 Minimum value = 0 Maximum value = 4094
nat46ignoretos	<String>	Read-write	Ignore TOS. Default value: NO Possible values = YES, NO
nat46zerochecksum	<String>	Read-write	Calculate checksum for UDP packets with zero checksum. Default value: ENABLED Possible values = ENABLED, DISABLED
nat46v6mtu	<Double>	Read-write	MTU setting for the IPv6 side. If the incoming IPv4 packet greater than

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
			this, either fragment or send icmp need fragmentation error. Default value: 1280 Minimum value = 1280 Maximum value = 9216
nat46fragheader	<String>	Read-write	When disabled, translator will not insert IPv6 fragmentation header for non fragmented IPv4 packets. Default value: ENABLED Possible values = ENABLED, DISABLED
__count	<Double>	Read-only	count parameter

Operations

(click to see Properties)

UPDATE| UNSET| GET (ALL)| GET| COUNT

Some options that you can use for each operations:

- **Getting warnings in response:**NITRO allows you to get warnings in an operation by specifying the "warning" query parameter as "yes". For example, to get warnings while connecting to the Citrix ADC appliance, the URL is as follows:

  http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/login?warning=yes

If any, the warnings are displayed in the response payload with the HTTP code "209 X-NITRO-WARNING".

- **Authenticated access for individual NITRO operations:**NITRO allows you to logon to the Citrix ADC appliance to perform individual operations. You can use this option instead of creating a NITRO session (using the login object) and then using that session to perform all operations,

  To do this, you must specify the username and password in the request header of the NITRO request as follows:

  X-NITRO-USER:<username>

  X-NITRO-PASS:<password>

  Note: In such cases, make sure that the request header DOES not include the following:

  Cookie:NITRO_AUTH_TOKEN=<tokenvalue>

`*Note:`*

Mandatory parameters are marked in redand placeholder content is marked in <green>.

update

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/inatparam

HTTP Method:PUT

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Content-Type:application/json

Request Payload:

```
{"inatparam":{

"nat46v6prefix":<String_value>,

"td":<Double_value>,

"nat46ignoretos":<String_value>,
```

```
"nat46zerochecksum":<String_value>,

"nat46v6mtu":<Double_value>,

"nat46fragheader":<String_value>

}}
```

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error

unset

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/inatparam?**action=unset**

HTTP Method:POST

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Content-Type:application/json

Request Payload:

```
{"inatparam":{

"nat46v6prefix":true,

"td":<Double_value>

}}
```

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error

get (all)

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/inatparam

Query-parameters:

attrs

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/inatparam?**attrs=property-name1,property-name2**

Use this query parameter to specify the resource details that you want to retrieve.

filter

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/inatparam?**filter=property-name1:property-val1,property-name2:property-val2**

Use this query-parameter to get the filtered set of inatparam resources configured on Citrix ADC. Filtering can be done on any of the properties of the resource.

view

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/inatparam?**view=summary**

Note: By default, the retrieved results are displayed in detail view (?view=detail).

pagination

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/inatparam?**pagesize=#no;pageno=#no**

Use this query-parameter to get the inatparam resources in chunks.

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "inatparam": [ {

"td":<Double_value>,

"nat46v6prefix":<String_value>,

"nat46ignoretos":<String_value>,

"nat46zerochecksum":<String_value>,

"nat46v6mtu":<Double_value>,

"nat46fragheader":<String_value>

}]}
```

get

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/inatparam/td_value<Double>

Query-parameters:

attrs

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/inatparam/td_value<Double>?**attrs=property-name1,property-name2**

Use this query parameter to specify the resource details that you want to retrieve.

view

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/inatparam/td_value<Double>?**view=summary**

Note: By default, the retrieved results are displayed in detail view (?view=detail).

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "inatparam": [ {

"td":<Double_value>,

"nat46v6prefix":<String_value>,

"nat46ignoretos":<String_value>,

"nat46zerochecksum":<String_value>,

"nat46v6mtu":<Double_value>,

"nat46fragheader":<String_value>

}]}
```

count

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/inatparam?**count=yes**

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "inatparam": [ { "__count": "#no"} ] }
```

Interface

Configuration for interface resource.

Properties

(click to see Operations)

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
id	<String>	Read-write	Interface number, in C/U format, where C can take one of the following values: * 0 - Indicates a management interface. * 1 - Indicates a 1 Gbps port. * 10 - Indicates a 10 Gbps port. * LA - Indicates a link aggregation port.

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
			* LO - Indicates a loop back port. U is a unique integer for representing an interface in a particular port group.
speed	<String>	Read-write	Ethernet speed of the interface, in Mbps. Notes: * If you set the speed as AUTO, the Citrix ADC attempts to auto-negotiate or auto-sense the link speed of the interface when it is UP. You must enable auto negotiation on the interface. * If you set a speed other than AUTO, you must specify the same speed for the peer network device. Mismatched speed and duplex settings between the peer devices of a link lead to link

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
			errors, packet loss, and other errors. Some interfaces do not support certain speeds. If you specify an unsupported speed, an error message appears. Default value: AUTO Possible values = AUTO, 10, 100, 1000, 10000, 25000, 40000, 50000, 100000
duplex	<String>	Read-write	The duplex mode for the interface. Notes:* If you set the duplex mode to AUTO, the Citrix ADC attempts to auto-negotiate the duplex mode of the interface when it is UP. You must enable auto negotiation on the interface. If you set a duplex mode other than AUTO, you must specify the same duplex mode for the peer network device. Mismatched speed and duplex settings between the peer devices of a link

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
			lead to link errors, packet loss, and other errors. Default value: AUTO Possible values = AUTO, HALF, FULL
flowctl	<String>	Read-write	802.3x flow control setting for the interface. The 802.3x specification does not define flow control for 10 Mbps and 100 Mbps speeds, but if a Gigabit Ethernet interface operates at those speeds, the flow control settings can be applied. The flow control setting that is finally applied to an interface depends on auto-negotiation. With the ON option, the peer negotiates the flow control, but the appliance then forces two-way flow control for the interface. Default value: OFF Possible values = OFF, RX, TX, RXTX, ON

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
autoneg	<String>	Read-write	Auto-negotiation state of the interface. With the ENABLED setting, the Citrix ADC auto-negotiates the speed and duplex settings with the peer network device on the link. The Citrix ADC appliance auto-negotiates the settings of only those parameters (speed or duplex mode) for which the value is set as AUTO. Default value: NSA_DVC_AUTONEG_ON Possible values = DISABLED, ENABLED
hamonitor	<String>	Read-write	In a High Availability (HA) configuration, monitor the interface for failure events. In an HA configuration, an interface that has HA MON enabled and is not bound to any Failover Interface Set (FIS), is a critical interface. Failure or disabling of any critical interface triggers HA failover.

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
			Default value: ON Possible values = ON, OFF
haheartbeat	<String>	Read-write	In a High Availability (HA) or Cluster configuration, configure the interface for sending heartbeats. In an HA or Cluster configuration, an interface that has HA Heartbeat disabled should not send the heartbeats. Default value: ON Possible values = OFF, ON
mtu	<Double>	Read-write	The maximum transmission unit (MTU) is the largest packet size, measured in bytes excluding 14 bytes ethernet header and 4 bytes crc, that can be transmitted and received by this interface. Default value of MTU is 1500 on all the interface of Citrix ADC any value configured more than 1500 on the interface will make the interface as jumbo enabled. In case of

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
			cluster backplane interface MTU value will be changed to 1514 by default, user has to change the backplane interface value to maximum mtu configured on any of the interface in cluster system plus 14 bytes more for backplane interface if Jumbo is enabled on any of the interface in a cluster system. Changing the backplane will bring back the MTU of backplane interface to default value of 1500. If a channel is configured as backplane then the same holds true for channel as well as member interfaces. Default value: 1500 Minimum value = 1500 Maximum value = 9216
tagall	<String>	Read-write	Add a four-byte 802.1q tag to every packet sent on this interface. The ON setting applies the tag for this

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
			interface's native VLAN. OFF applies the tag for all VLANs other than the native VLAN. Default value: OFF Possible values = ON, OFF
trunk	<String>	Read-write	This argument is deprecated by tagall. Default value: OFF Possible values = ON, OFF
trunkmode	<String>	Read-write	Accept and send 802.1q VLAN tagged packets, based on Allowed Vlan List of this interface. Default value: OFF Possible values = ON, OFF
trunkallowedvlan	<String[]>	Read-write	VLAN ID or range of VLAN IDs will be allowed on this trunk interface. In the command line interface, separate the range with a hyphen. For example: 40-90. Minimum length = 1 Maximum length = 4094

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
lacpmode	<String>	Read-write	Bind the interface to a LA channel created by the Link Aggregation control protocol (LACP). Available settings function as follows: * Active - The LA channel port of the Citrix ADC generates LACPDU messages on a regular basis, regardless of any need expressed by its peer device to receive them. * Passive - The LA channel port of the Citrix ADC does not transmit LACPDU messages unless the peer device port is in the active mode. That is, the port does not speak unless spoken to. * Disabled - Unbinds the interface from the LA channel. If this is the only interface in the LA channel, the

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
			LA channel is removed. Default value: DISABLED Possible values = DISABLED, ACTIVE, PASSIVE
lacpkey	<Double>	Read-write	Integer identifying the LACP LA channel to which the interface is to be bound. For an LA channel of the Citrix ADC, this digit specifies the variable x of an LA channel in LA/x notation, where x can range from 1 to 8. For example, if you specify 3 as the LACP key for an LA channel, the interface is bound to the LA channel LA/3. For an LA channel of a cluster configuration, this digit specifies the variable y of a cluster LA channel in CLA/(y-4) notation, where y

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
			can range from 5 to 8. For example, if you specify 6 as the LACP key for a cluster LA channel, the interface is bound to the cluster LA channel CLA/2. Minimum value = 1 Maximum value = 8
lagtype	<String>	Read-write	Type of entity (Citrix ADC or cluster configuration) for which to create the channel. Default value: NODE Possible values = NODE, CLUSTER
lacppriority	<Double>	Read-write	LACP port priority, expressed as an integer. The lower the number, the higher the priority. The Citrix ADC limits the number of interfaces in an LA channel to sixteen. Default value: 32768 Minimum value = 1 Maximum value = 65535

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
lacptimeout	<String>	Read-write	Interval at which the Citrix ADC sends LACPDU messages to the peer device on the LA channel. Available settings function as follows: LONG - 30 seconds. SHORT - 1 second. Default value: NSA_LACP_TIMEOUT_LONG Possible values = LONG, SHORT
ifalias	<String>	Read-write	Alias name for the interface. Used only to enhance readability. To perform any operations, you have to specify the interface ID. Default value: " " Maximum length = 31
throughput	<Double>	Read-write	Low threshold value for the throughput of the interface, in Mbps. In an HA configuration, failover is triggered if the interface has HA MON enabled and

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
			the throughput is below the specified the threshold. Minimum value = 0 Maximum value = 160000
linkredundancy	<String>	Read-write	Link Redundancy for Cluster LAG. Default value: OFF Possible values = ON, OFF
bandwidthhigh	<Double>	Read-write	High threshold value for the bandwidth usage of the interface, in Mbps. The Citrix ADC generates an SNMP trap message when the bandwidth usage of the interface is greater than or equal to the specified high threshold value. Minimum value = 0 Maximum value = 160000
bandwidthnormal	<Double>	Read-write	Normal threshold value for the bandwidth usage of the interface, in Mbps. When the bandwidth usage of the interface becomes less than or equal to the

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
			specified normal threshold after exceeding the high threshold, the Citrix ADC generates an SNMP trap message to indicate that the bandwidth usage has returned to normal. Minimum value = 0 Maximum value = 160000
lldpmode	<String>	Read-write	Link Layer Discovery Protocol (LLDP) mode for an interface. The resultant LLDP mode of an interface depends on the LLDP mode configured at the global and the interface levels. Possible values = NONE, TRANSMITTER, RECEIVER, TRANSCEIVER
lrsetpriority	<Double>	Read-write	LRSET port priority, expressed as an integer ranging from 1 to 1024. The highest priority is 1. The Citrix ADC limits the number of interfaces in an LRSET to 8. Within a

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
			LRSET the highest LR Priority Interface is considered as the first candidate for the Active interface, if the interface is UP. Default value: 1024 Minimum value = 1 Maximum value = 1024
devicename	<String>	Read-only	Name of the interface.
unit	<Double>	Read-only	Unit number for this interface, signifying the sequence number in which this interface is discovered on this Citrix ADC.
description	<String>	Read-only	Display the type of interface, the speeds at which this interface can operate, and, if applicable, the type of SFP,.
flags	<Double>	Read-only	Flags for this interface. Used for communicating the device states.
actualmtu	<Double>	Read-only	MTU for this interface (the largest frame that can transit this interface).

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
vlan	<Double>	Read-only	Native VLAN for this interface.
mac	<String>	Read-only	MAC address for this interface.
uptime	<Double>	Read-only	Duration for which the interface has been UP (Example: 3 hours 1 minute 1 second). This value is reset when the interface state changes to DOWN..
downtime	<Double>	Read-only	Duration for which the interface has been DOWN. This value is reset when the interface state changes to UP. (Example: 3 hours 1 minute 1 second).
reqmedia	<String>	Read-only	Requested media setting for this interface. Possible values = AUTO, UTP, FIBER
reqspeed	<String>	Read-only	Requested speed setting for this interface. Possible values = AUTO, 10, 100, 1000, 10000, 25000, 40000, 50000, 100000

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
reqduplex	<String>	Read-only	Requested duplex setting for this interface. Possible values = AUTO, HALF, FULL
reqflowcontrol	<String>	Read-only	Requested flow control setting for this interface. Possible values = OFF, RX, TX, RXTX, ON
actmedia	<String>	Read-only	Actual media setting for this interface. Possible values = AUTO, UTP, FIBER
actspeed	<String>	Read-only	Actual speed setting for this interface. Possible values = AUTO, 10, 100, 1000, 10000, 25000, 40000, 50000, 100000
actduplex	<String>	Read-only	Actual duplex setting for this interface. Possible values = AUTO, HALF, FULL
actflowctl	<String>	Read-only	Actual flow control setting for this interface. Possible values = OFF, RX, TX, RXTX, ON

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
mode	<String>	Read-only	Interface Link Aggregation mode (Auto/Manual) setting. Possible values = MANUAL, AUTO
state	<String>	Read-only	Link state of the interface (UP/DOWN). Default value: ENABLED Possible values = ENABLED, DISABLED
autonegresult	<Double>	Read-only	Actual auto-negotiation setting for this interface.
tagged	<Double>	Read-only	VLAN tags setting on this channel.
taggedany	<Double>	Read-only	Interface setting to accept/drop all tagged packets.
taggedautolearn	<Double>	Read-only	Dynamic VLAN membership autolearning enabled or disabled on this interface.
hangdetect	<Double>	Read-only	Hang detection enabled or disabled for this interface.

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
hangreset	<Double>	Read-only	Hang reset enabled or disabled for this interface.
linkstate	<Double>	Read-only	The current state of the link associated with the interface. For logical interfaces (LA), the state of the link is dependent on the state of the slave interfaces. For the link to be UP at least one of the slave interfaces needs to be UP.
intfstate	<Double>	Read-only	Current state of the specified interface. The interface state set to UP only if the link state is UP and administrative state is ENABLED.
rxpackets	<Double>	Read-only	Number of packets received by an interface since the Citrix ADC was started or the interface statistics were cleared.
rxbytes	<Double>	Read-only	Number of bytes received by an interface since the Citrix ADC was started or the interface statistics were cleared.

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
rxerrors	<Double>	Read-only	Number of inbound packets dropped by the hardware on a specified interface since the Citrix ADC was started or the interface statistics were cleared. Packets can be dropped because of CRC, length (undersize or oversize), or alignment errors.
rxdrops	<Double>	Read-only	Number of inbound packets dropped by the specified interface. Commonly dropped packets are multicast frames, spanning tree BPDUs, packets destined to a MAC not owned by the Citrix ADC when L2 mode is disabled, or packets tagged for a VLAN that is not bound to the interface. In most healthy networks, this statistic increments at a steady rate regardless of traffic load. A sharp spike in dropped packets generally indicates an issue with connected L2

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
			switches, such as a forwarding database overflow resulting in packets being broadcast on all ports.
txpackets	<Double>	Read-only	Number of packets transmitted by an interface since the Citrix ADC was started or the interface statistics were cleared.
txbytes	<Double>	Read-only	Number of bytes transmitted by an interface since the Citrix ADC was started or the interface statistics were cleared.
txerrors	<Double>	Read-only	Number of outbound packets dropped by the hardware on a specified interface since the Citrix ADC was started or the interface statistics were cleared. Packets can be dropped because of length (undersize or oversize) errors or a lack of resources. This statistic is available only for:

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
			(1) Loop back interface (LO) of all platforms. (2) All data ports on the Citrix ADC 12000 platform. (3) Management ports on the Citrix ADC MPX 15000 and 17000 platforms. .
txdrops	<Double>	Read-only	Number of packets dropped in transmission by the specified interface for one of the following reasons. (1) VLAN mismatch. (2) Oversized packets. (3) Interface congestion. (4) Loopback packets sent on non loop back interface. .
indisc	<Double>	Read-only	Number of error-free inbound packets discarded by the specified interface because of a lack of

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
			resources (for example, insufficient receive buffers).
outdisc	<Double>	Read-only	Number of error-free outbound packets discarded by the specified interface because of a lack of resources. This statistic is not available on: (1) 10G ports of Citrix ADC MPX 12500/12500/15500 -10G platforms. (2) 10G data ports on Citrix ADC MPX 17500/19500/21500 platforms. .
fctls	<Double>	Read-only	Number of times flow control is performed on the specified interface because of received pause frames.
hangs	<Double>	Read-only	Number of times the specified interface detected hangs in the transmit and receive paths since the Citrix ADC was started or the interface statistics were cleared.

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
stsstalls	<Double>	Read-only	Number of times the status updates for a specified interface were stalled since the Citrix ADC was started or the interface statistics were cleared. A status stall is detected when the status of the interface is not updated by the NIC hardware within 0.8 seconds of the last update.
txstalls	<Double>	Read-only	Number of times the interface stalled, when transmitting packets, since the Citrix ADC was started or the interface statistics were cleared. Transmit (Tx) stalls are detected when a packet posted for transmission is not transmitted in 4 seconds.
rxstalls	<Double>	Read-only	Number of times the interface stalled, when receiving packets, since the Citrix ADC was started or the interface statistics were cleared. Receive (Rx) stalls

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
			are detected when the following conditions are met: (1)The link is up for more than 10 minutes. (2)Packets are transmitted, but no packets are received for 16 seconds. .
bdgmacmoved	<Double>	Read-only	Number of MAC moves between ports. A high rate of MAC moves typically indicates a bridge loop between two interfaces.
bdgmuted	<Double>	Read-only	Number of times the specified interface stopped transmitting and receiving packets because of MAC moves between ports.
vmac	<String>	Read-only	Virtual MAC of this interface.
vmac6	<String>	Read-only	Virtual MAC for IPv6 of this interface.
reqthroughput	<Double>	Read-only	Minimum required throughput for an interface. Failover is triggered if the

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
			operating throughput of a Link Aggregation (LA) channel for which HAMON is ON falls below this value. The possible values are: 1. 1000Mbps for 1G interfaces. 2. 10000Mbps for 10G interfaces. 3. 160000Mbps for Link Aggregation channels. Minimum value = 0 Maximum value = 160000
actthroughput	<Double>	Read-only	Actual throughput for the interface.
backplane	<String>	Read-only	The cluster backplane status of the interface. If the status is enabled, the interface is part of the cluster backplane. By default, the backplane status is disabled. Possible values = ENABLED, DISABLED

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
ifnum	<String[]>	Read-only	Contains the LA Master, if the interface is part of LA channel.
cleartime	<Double>	Read-only	Time since the interface stats are cleared last time.
slavestate	<Double>	Read-only	State of the member interfaces.
slavemedia	<Double>	Read-only	Media type of the member interfaces.
slavespeed	<Double>	Read-only	Speed of the member interfaces.
slaveduplex	<Double>	Read-only	Duplex of the member interfaces.
slaveflowctl	<Double>	Read-only	Flowcontrol of the member interfaces.
slavetime	<Double>	Read-only	UP time of the member interfaces.
intftype	<String>	Read-only	Interface Type, this field will have the interface type either it is virtual, physical or loopback. . Possible values = BROADCOM 5700/5701, TIGON1/TIGON2, INTEL 82546, INTEL 8255X(PRO), Link Aggregate,

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
			Loopback, Intel 82541/47, Broadcom 5704, Chelsio 1G, Intel 8247X, Intel 82576 VF, Xen Virtual, Intel 10G, Intel 82599 VF, Mellanox 100G, Mellanox 50G, Mellanox 25G, Intel FVL 40G, Intel FVL 25G, Intel FVL 10G, Intel FVL 40G VF, Mellanox VF, Hyper v, Cluster LAG, Intel 8247X SFP, XEN Interface, Chelsio 10G, KVM Virtio, VMXNET3
svmcmd	<Double>	Read-only	New attribute added to identify the source of cmd, when SVM fires the nitro cmd, it will set the value of SVMCMD to be 1. . Default value: 0
lacpactormode	<String>	Read-only	* Active - The LA channel port of the Citrix ADC generates LACPDU messages on a regular basis, regardless of any need expressed by

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
			its peer device to receive them. * Passive - The LA channel port of the Citrix ADC does not transmit LACPDU messages unless the peer device port is in the active mode. That is, the port does not speak unless spoken to. * Disabled - Unbinds the interface from the LA channel. If this is the only interface in the LA channel, the LA channel is removed. Possible values = DISABLED, ACTIVE, PASSIVE
lacpactortimeout	<String>	Read-only	Interval at which the Citrix ADC sends LACPDU messages to the peer device on the LA channel. Available settings function as follows: LONG - 30 seconds.

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
			SHORT - 1 second. Possible values = LONG, SHORT
lacpactorpriority	<Double>	Read-only	LACP Actor Priority. A LACP port priority is configured on each port using LACP. LACP uses the port priority with the port number to form the port identifier. The port priority determines which ports should be put in standby mode when there is a hardware limitation that prevents all compatible ports from aggregating.
lacpactorportno	<Double>	Read-only	LACP Actor port number. LACP uses the port priority with the port number to form the port identifier.
lacppartnerstate	<String>	Read-only	LACP Partner State. Whether the port is in Active or Passive negotiating state. Possible values = MANUAL, AUTO
lacppartnertimeout	<String>	Read-only	The timeout value for the information revieved in LACPDUs. It can

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
			have values as SHORT or LONG. The SHORT timeout is 3s and the LONG timeout is 90s. Possible values = LONG, SHORT
lacppartneraggregation	<String>	Read-only	The Aggregation flag indicates that the participant will allow the link to be used as part of an aggregate. Otherwise the link is to be used as an individual link, i.e. not aggregated with any other. Possible values = NS_EMPTY_STR, AGG
lacppartnerinsync	<String>	Read-only	The Synchronization flag indicates that the transmitting participant.s mux component is in sync with the system id and key information transmitted. Possible values = NS_EMPTY_STR, SYNC
lacppartnercollecting	<String>	Read-only	The Collecting flag indicates that the participant.s collector, i.e. the reception component of the mux, is

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
			definitely on. If set the flag communicates collecting. Possible values = NS_EMPTY_STR, COLLECTING
lacppartnerdistributing	<String>	Read-only	The Distributing flag indicates that the participant.s distributor is not definitely off. If reset the flag indicates not distributing. Possible values = NS_EMPTY_STR, DISTRIBUTING
lacppartnerdefaulted	<String>	Read-only	If the timer expires in the Expired state, the Receive Machine enters the Defaulted state. Possible values = NS_EMPTY_STR, DEFAULTED
lacppartnerexpired	<String>	Read-only	If the LACPDUs are received for timeout period, the Receive Machine enters the Expired state and the timer is restarted with the timeout value of SHORT timeout.

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
			Possible values = NS_EMPTY_STR, EXPIRED
lacppartnerpriority	<Double>	Read-only	LACP Partner Priority. A LACP port priority is configured on each port using LACP. LACP uses the port priority with the port number to form the port identifier. The port priority determines which ports should be put in standby mode when there is a hardware limitation that prevents all compatible ports from aggregating.
lacppartnersystemmac	<String>	Read-only	LACP Partner System MAC.
lacppartnersystempriority	<Double>	Read-only	LACP Partner System Priority. The LACP partner's system priority. The values for the priority range from 0 to 65535. The lower the value, the higher the system priority. The switch with the lower system priority value determines which links between LACP partner are active

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
			and which are in the standby for each LACP Channel.
lacppartnerportno	<Double>	Read-only	LACP Partner Port number. LACP uses the port priority with the port number to form the port identifier.
lacppartnerkey	<Double>	Read-only	LACP Partner Key. The LACP key used by the partner port.
lacpactoraggregation	<String>	Read-only	The Aggregation flag indicates that the participant will allow the link to be used as part of an aggregate. Otherwise the link is to be used as an individual link, i.e. not aggregated with any other. Possible values = NS_EMPTY_STR, AGG
lacpactorinsync	<String>	Read-only	The Synchronization flag indicates that the transmitting participant.s mux component is in sync with the system id and key information transmitted. Possible values = NS_EMPTY_STR, SYNC

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
lacpactorcollecting	<String>	Read-only	The Collecting flag indicates that the participant.s collector, i.e. the reception component of the mux, is definitely on. If set the flag communicates collecting. Possible values = NS_EMPTY_STR, COLLECTING
lacpactordistributing	<String>	Read-only	The Distributing flag indicates that the participant.s distributor is not definitely off. If reset the flag indicates not distributing. Possible values = NS_EMPTY_STR, DISTRIBUTING
lacpportmuxstate	<String>	Read-only	LACP Port MUX state. The state of the MUX control machine. The Mux Control Machine attaches the physical port to an aggregate port, using the Selection Logic to choose an appropriate port, and turns the distributor and collector for the physical port on or

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
			off as required by protocol information. Possible values = DETACHED, WAITING, ATTACHED, COLLECTING, DISTRIBUTING
lacpportrxstat	<String>	Read-only	LACP Port RX state. The state of the Receive machine. The Receive Machine maintains partner information, recording protocol information from LACPDUs sent by remote partner(s). Received information is subject to a timeout, and if sufficient time elapses the receive machine will revert to using default partner information. Possible values = INIT, PORT_DISABLED, LACP_DISABLED, EXPIRED, DEFAULTED, CURRENT
lacpportselectstate	<String>	Read-only	LACP Port SELECT state. The state of the SELECT state machine, It could be

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
			SELECTED or UNSELECTED. Possible values = UNSELECTED, SELECTED, STANDBY
lractiveintf	<Boolean>	Read-only	LR set member interface state(active/inactive).
__count	<Double>	Read-only	count parameter

Operations

(click to see Properties)

CLEAR| UPDATE| UNSET| ENABLE| DISABLE| RESET| GET (ALL)| GET| COUNT

Some options that you can use for each operations:

- **Getting warnings in response:**NITRO allows you to get warnings in an operation by specifying the "warning" query parameter as "yes". For example, to get warnings while connecting to the Citrix ADC appliance, the URL is as follows:

  http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/login?warning=yes

  If any, the warnings are displayed in the response payload with the HTTP code "209 X-NITRO-WARNING".

- **Authenticated access for individual NITRO operations:**NITRO allows you to logon to the Citrix ADC appliance to perform individual operations. You can use this option instead of creating a NITRO session (using the login object) and then using that session to perform all operations,

  To do this, you must specify the username and password in the request header of the NITRO request as follows:

  X-NITRO-USER:<username>

  X-NITRO-PASS:<password>

Note: In such cases, make sure that the request header DOES not include the following:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue>

`*Note:` *

Mandatory parameters are marked in redand placeholder content is marked in <green>.

clear

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/Interface?**action=clear**

HTTP Method:POST

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Content-Type:application/json

Request Payload:

```
{"Interface":{

<b>"id":<String_value>

</b>}}
```

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error

update

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/Interface

HTTP Method:PUT

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Content-Type:application/json

Request Payload:

```
{"Interface":{

<b>"id":<String_value>,

</b>"speed":<String_value>,

"duplex":<String_value>,

"flowctl":<String_value>,

"autoneg":<String_value>,

"hamonitor":<String_value>,

"haheartbeat":<String_value>,

"mtu":<Double_value>,

"tagall":<String_value>,

"trunk":<String_value>,

"trunkmode":<String_value>,

"trunkallowedvlan":<String[]_value>,

"lacpmode":<String_value>,

"lacpkey":<Double_value>,

"lagtype":<String_value>,

"lacppriority":<Double_value>,

"lacptimeout":<String_value>,

"ifalias":<String_value>,

"throughput":<Double_value>,

"linkredundancy":<String_value>,
```

```
"bandwidthhigh":<Double_value>,

"bandwidthnormal":<Double_value>,

"lldpmode":<String_value>,

"lrsetpriority":<Double_value>

}}
```

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error

unset

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/Interface?**action=unset**

HTTP Method:POST

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Content-Type:application/json

Request Payload:

```
{"Interface":{

<b>"id":<String_value>,

</b>"speed":true,

"duplex":true,

"flowctl":true,

"autoneg":true,
```

```
"hamonitor":true,

"haheartbeat":true,

"mtu":true,

"tagall":true,

"trunk":true,

"trunkmode":true,

"trunkallowedvlan":true,

"lacpmode":true,

"lacppriority":true,

"lacptimeout":true,

"ifalias":true,

"throughput":true,

"linkredundancy":true,

"bandwidthhigh":true,

"bandwidthnormal":true,

"lldpmode":true,

"lrsetpriority":true

}}
```


Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error

enable

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/Interface?**action=enable**

HTTP Method:POST

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Content-Type:application/json

Request Payload:

```
{"Interface":{

<b>"id":<String_value>

</b>}}
```

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error

disable

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/Interface?**action=disable**

HTTP Method:POST

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Content-Type:application/json

Request Payload:

```
{"Interface":{

<b>"id":<String_value>
```

```
</b>}}
```

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error

reset

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/Interface?**action=reset**

HTTP Method:POST

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Content-Type:application/json

Request Payload:

```
{"Interface":{

<b>"id":<String_value>

</b>}}
```

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error

get (all)

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/Interface

Query-parameters:

attrs

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/Interface?**attrs=property-name1,property-name2**

Use this query parameter to specify the resource details that you want to retrieve.

filter

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/Interface?**filter=property-name1:property-val1,property-name2:property-val2**

Use this query-parameter to get the filtered set of Interface resources configured on Citrix ADC. Filtering can be done on any of the properties of the resource.

view

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/Interface?**view=summary**

Note: By default, the retrieved results are displayed in detail view (?view=detail).

pagination

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/Interface?**pagesize=#no;pageno=#no**

Use this query-parameter to get the Interface resources in chunks.

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "Interface": [ {
```

```
"id":<String_value>,

"devicename":<String_value>,

"unit":<Double_value>,

"description":<String_value>,

"flags":<Double_value>,

"mtu":<Double_value>,

"actualmtu":<Double_value>,

"vlan":<Double_value>,

"mac":<String_value>,

"uptime":<Double_value>,

"downtime":<Double_value>,

"reqmedia":<String_value>,

"reqspeed":<String_value>,

"reqduplex":<String_value>,

"reqflowcontrol":<String_value>,

"actmedia":<String_value>,

"actspeed":<String_value>,

"actduplex":<String_value>,

"actflowctl":<String_value>,

"mode":<String_value>,

"hamonitor":<String_value>,

"haheartbeat":<String_value>,
```

```
"state":<String_value>,

"autoneg":<String_value>,

"autonegresult":<Double_value>,

"tagged":<Double_value>,

"tagall":<String_value>,

"trunk":<String_value>,

"trunkmode":<String_value>,

"trunkallowedvlan":<String[]_value>,

"taggedany":<Double_value>,

"taggedautolearn":<Double_value>,

"hangdetect":<Double_value>,

"hangreset":<Double_value>,

"linkstate":<Double_value>,

"intfstate":<Double_value>,

"rxpackets":<Double_value>,

"rxbytes":<Double_value>,

"rxerrors":<Double_value>,

"rxdrops":<Double_value>,

"txpackets":<Double_value>,

"txbytes":<Double_value>,

"txerrors":<Double_value>,

"txdrops":<Double_value>,
```

```
"indisc":<Double_value>,

"outdisc":<Double_value>,

"fctls":<Double_value>,

"hangs":<Double_value>,

"stsstalls":<Double_value>,

"txstalls":<Double_value>,

"rxstalls":<Double_value>,

"bdgmacmoved":<Double_value>,

"bdgmuted":<Double_value>,

"vmac":<String_value>,

"vmac6":<String_value>,

"lacpmode":<String_value>,

"lacpkey":<Double_value>,

"lacppriority":<Double_value>,

"lacptimeout":<String_value>,

"lagtype":<String_value>,

"ifalias":<String_value>,

"reqthroughput":<Double_value>,

"linkredundancy":<String_value>,

"actthroughput":<Double_value>,

"bandwidthhigh":<Double_value>,

"bandwidthnormal":<Double_value>,
```

```
"backplane":<String_value>,

"ifnum":<String[]_value>,

"cleartime":<Double_value>,

"slavestate":<Double_value>,

"slavemedia":<Double_value>,

"slavespeed":<Double_value>,

"slaveduplex":<Double_value>,

"slaveflowctl":<Double_value>,

"slavetime":<Double_value>,

"intftype":<String_value>,

"svmcmd":<Double_value>,

"lacpactormode":<String_value>,

"lacpactortimeout":<String_value>,

"lacpactorpriority":<Double_value>,

"lacpactorportno":<Double_value>,

"lacppartnerstate":<String_value>,

"lacppartnertimeout":<String_value>,

"lacppartneraggregation":<String_value>,

"lacppartnerinsync":<String_value>,

"lacppartnercollecting":<String_value>,

"lacppartnerdistributing":<String_value>,

"lacppartnerdefaulted":<String_value>,
```

```
"lacppartnerexpired":<String_value>,

"lacppartnerpriority":<Double_value>,

"lacppartnersystemmac":<String_value>,

"lacppartnersystempriority":<Double_value>,

"lacppartnerportno":<Double_value>,

"lacppartnerkey":<Double_value>,

"lacpactoraggregation":<String_value>,

"lacpactorinsync":<String_value>,

"lacpactorcollecting":<String_value>,

"lacpactordistributing":<String_value>,

"lacpportmuxstate":<String_value>,

"lacpportrxstat":<String_value>,

"lacpportselectstate":<String_value>,

"lldpmode":<String_value>,

"lrsetpriority":<Double_value>,

"lractiveintf":<Boolean_value>

}]}
```

get

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/Interface/id_value<String>

Query-parameters:

attrs

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/Interface/id_value<String>?**attrs=property-name1,property-name2**

Use this query parameter to specify the resource details that you want to retrieve.

view

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/Interface/id_value<String>?**view=summary**

Note: By default, the retrieved results are displayed in detail view (?view=detail).

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "Interface": [ {

"id":<String_value>,

"devicename":<String_value>,

"unit":<Double_value>,

"description":<String_value>,

"flags":<Double_value>,

"mtu":<Double_value>,

"actualmtu":<Double_value>,
```

```
"vlan":<Double_value>,

"mac":<String_value>,

"uptime":<Double_value>,

"downtime":<Double_value>,

"reqmedia":<String_value>,

"reqspeed":<String_value>,

"reqduplex":<String_value>,

"reqflowcontrol":<String_value>,

"actmedia":<String_value>,

"actspeed":<String_value>,

"actduplex":<String_value>,

"actflowctl":<String_value>,

"mode":<String_value>,

"hamonitor":<String_value>,

"haheartbeat":<String_value>,

"state":<String_value>,

"autoneg":<String_value>,

"autonegresult":<Double_value>,

"tagged":<Double_value>,

"tagall":<String_value>,

"trunk":<String_value>,

"trunkmode":<String_value>,
```

```
"trunkallowedvlan":<String[]_value>,

"taggedany":<Double_value>,

"taggedautolearn":<Double_value>,

"hangdetect":<Double_value>,

"hangreset":<Double_value>,

"linkstate":<Double_value>,

"intfstate":<Double_value>,

"rxpackets":<Double_value>,

"rxbytes":<Double_value>,

"rxerrors":<Double_value>,

"rxdrops":<Double_value>,

"txpackets":<Double_value>,

"txbytes":<Double_value>,

"txerrors":<Double_value>,

"txdrops":<Double_value>,

"indisc":<Double_value>,

"outdisc":<Double_value>,

"fctls":<Double_value>,

"hangs":<Double_value>,

"stsstalls":<Double_value>,

"txstalls":<Double_value>,

"rxstalls":<Double_value>,
```

```
"bdgmacmoved":<Double_value>,

"bdgmuted":<Double_value>,

"vmac":<String_value>,

"vmac6":<String_value>,

"lacpmode":<String_value>,

"lacpkey":<Double_value>,

"lacppriority":<Double_value>,

"lacptimeout":<String_value>,

"lagtype":<String_value>,

"ifalias":<String_value>,

"reqthroughput":<Double_value>,

"linkredundancy":<String_value>,

"actthroughput":<Double_value>,

"bandwidthhigh":<Double_value>,

"bandwidthnormal":<Double_value>,

"backplane":<String_value>,

"ifnum":<String[]_value>,

"cleartime":<Double_value>,

"slavestate":<Double_value>,

"slavemedia":<Double_value>,

"slavespeed":<Double_value>,

"slaveduplex":<Double_value>,
```

```
"slaveflowctl":<Double_value>,

"slavetime":<Double_value>,

"intftype":<String_value>,

"svmcmd":<Double_value>,

"lacpactormode":<String_value>,

"lacpactortimeout":<String_value>,

"lacpactorpriority":<Double_value>,

"lacpactorportno":<Double_value>,

"lacppartnerstate":<String_value>,

"lacppartnertimeout":<String_value>,

"lacppartneraggregation":<String_value>,

"lacppartnerinsync":<String_value>,

"lacppartnercollecting":<String_value>,

"lacppartnerdistributing":<String_value>,

"lacppartnerdefaulted":<String_value>,

"lacppartnerexpired":<String_value>,

"lacppartnerpriority":<Double_value>,

"lacppartnersystemmac":<String_value>,

"lacppartnersystempriority":<Double_value>,

"lacppartnerportno":<Double_value>,

"lacppartnerkey":<Double_value>,

"lacpactoraggregation":<String_value>,
```

```
"lacpactorinsync":<String_value>,

"lacpactorcollecting":<String_value>,

"lacpactordistributing":<String_value>,

"lacpportmuxstate":<String_value>,

"lacpportrxstat":<String_value>,

"lacpportselectstate":<String_value>,

"lldpmode":<String_value>,

"lrsetpriority":<Double_value>,

"lractiveintf":<Boolean_value>

}]}
```

count

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/Interface?**count=yes**

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "Interface": [ { "__count": "#no"} ] }
```

interfacepair

Configuration for "Interface Pair" resource.

Properties

(click to see Operations)

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
id	<Double>	Read-write	The Interface pair id. Minimum value = 1 Maximum value = 255
ifnum	<String[]>	Read-write	The constituent interfaces in the interface pair. Minimum length = 1
ifaces	<String>	Read-only	Names of all member interfaces of this Interface Pair.
__count	<Double>	Read-only	count parameter

Operations

(click to see Properties)

ADD| DELETE| GET (ALL)| GET| COUNT

Some options that you can use for each operations:

- **Getting warnings in response:**NITRO allows you to get warnings in an operation by specifying the "warning" query parameter as "yes". For example, to get warnings while connecting to the Citrix ADC appliance, the URL is as follows:

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/login?warning=yes

If any, the warnings are displayed in the response payload with the HTTP code "209 X-NITRO-WARNING".

- **Authenticated access for individual NITRO operations:**NITRO allows you to logon to the Citrix ADC appliance to perform individual operations. You can use this option instead of creating a NITRO session (using the login object) and then using that session to perform all operations,

  To do this, you must specify the username and password in the request header of the NITRO request as follows:

  X-NITRO-USER:<username>

  X-NITRO-PASS:<password>

  Note: In such cases, make sure that the request header DOES not include the following:

  Cookie:NITRO_AUTH_TOKEN=<tokenvalue>

`*Note:`*

Mandatory parameters are marked in redand placeholder content is marked in <green>.

add

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/interfacepair

HTTP Method:POST

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Content-Type:application/json

Request Payload:

```
{"interfacepair":{

<b>"id":<Double_value>,

</b><b>"ifnum":<String[]_value>

</b>}}
```

Response:

HTTP Status Code on Success: 201 Created HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error

delete

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/interfacepair/id_value<Double>

HTTP Method:DELETE

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue>

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error

get (all)

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/interfacepair

Query-parameters:

attrs

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/interfacepair?**attrs=property-name1,property-name2**

Use this query parameter to specify the resource details that you want to retrieve.

filter

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/interfacepair?**filter=property-name1:property-val1,property-name2:property-val2**

Use this query-parameter to get the filtered set of interfacepair resources configured on Citrix ADC. Filtering can be done on any of the properties of the resource.

view

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/interfacepair?**view=summary**

Note: By default, the retrieved results are displayed in detail view (?view=detail).

pagination

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/interfacepair?**pagesize=#no;pageno=#no**

Use this query-parameter to get the interfacepair resources in chunks.

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "interfacepair": [ {

"id":<Double_value>,

"ifnum":<String[]_value>,

"ifaces":<String_value>

}]}
```

get

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/interfacepair/id_value<Double>

Query-parameters:

attrs

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/interfacepair/id_value<Double>?**attrs=property-name1,property-name2**

Use this query parameter to specify the resource details that you want to retrieve.

view

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/interfacepair/id_value<Double>?**view=summary**

Note: By default, the retrieved results are displayed in detail view (?view=detail).

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "interfacepair": [ {

"id":<Double_value>,

"ifnum":<String[]_value>,

"ifaces":<String_value>

}]}
```

count

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/interfacepair?**count=yes**

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "interfacepair": [ { "__count": "#no"} ] }
```

ip6tunnel

Configuration for ip6 Tunnel resource.

Properties

(click to see Operations)

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
name	<String>	Read-write	Name for the IPv6 Tunnel. Cannot be changed after the service group is created. Must begin with a number or letter, and can consist of letters, numbers, and the @

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
			_ - . (period) : (colon) # and space ( ) characters. Minimum length = 1 Maximum length = 31
remote	<String>	Read-write	An IPv6 address of the remote Citrix ADC used to set up the tunnel. Minimum length = 1
local	<String>	Read-write	An IPv6 address of the local Citrix ADC used to set up the tunnel.
ownergroup	<String>	Read-write	The owner node group in a Cluster for the tunnel. Default value: DEFAULT_NG Minimum length = 1
remoteip	<String>	Read-only	The remote IP address or subnet of the tunnel.
type	<Double>	Read-only	The type of this tunnel.
encapip	<String>	Read-only	The effective local IP address of the tunnel. Used as the source of the encapsulated packets.

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
__count	<Double>	Read-only	count parameter

Operations

(click to see Properties)

ADD| DELETE| GET (ALL)| GET| COUNT

Some options that you can use for each operations:

- **Getting warnings in response:**NITRO allows you to get warnings in an operation by specifying the "warning" query parameter as "yes". For example, to get warnings while connecting to the Citrix ADC appliance, the URL is as follows:

  http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/login?warning=yes

  If any, the warnings are displayed in the response payload with the HTTP code "209 X-NITRO-WARNING".
- **Authenticated access for individual NITRO operations:**NITRO allows you to logon to the Citrix ADC appliance to perform individual operations. You can use this option instead of creating a NITRO session (using the login object) and then using that session to perform all operations,

  To do this, you must specify the username and password in the request header of the NITRO request as follows:

  X-NITRO-USER:<username>

  X-NITRO-PASS:<password>

  Note: In such cases, make sure that the request header DOES not include the following:

  Cookie:NITRO_AUTH_TOKEN=<tokenvalue>

`*Note:`*

Mandatory parameters are marked in redand placeholder content is marked in <green>.

add

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/ip6tunnel

HTTP Method:POST

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Content-Type:application/json

Request Payload:

```
{"ip6tunnel":{

<b>"name":<String_value>,

</b><b>"remote":<String_value>,

</b><b>"local":<String_value>,

</b>"ownergroup":<String_value>

}}
```

Response:

HTTP Status Code on Success: 201 Created HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error

delete

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/ip6tunnel/name_value<String>

HTTP Method:DELETE

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue>

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error

get (all)

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/ip6tunnel

Query-parameters:

args

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/ip6tunnel?**args=name:<String_value>,remote:<String_value>**

Use this query-parameter to get ip6tunnel resources based on additional properties.

attrs

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/ip6tunnel?**attrs=property-name1,property-name2**

Use this query parameter to specify the resource details that you want to retrieve.

filter

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/ip6tunnel?**filter=property-name1:property-val1,property-name2:property-val2**

Use this query-parameter to get the filtered set of ip6tunnel resources configured on Citrix ADC. Filtering can be done on any of the properties of the resource.

view

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/ip6tunnel?**view=summary**

Note: By default, the retrieved results are displayed in detail view (?view=detail).

pagination

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/ip6tunnel?**pagesize=#no;pageno=#no**

Use this query-parameter to get the ip6tunnel resources in chunks.

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "ip6tunnel": [ {

name:<String_value>,remote:<String_value>"remoteip":<String_value>,

"local":<String_value>,

"type":<Double_value>,

"encapip":<String_value>,

"ownergroup":<String_value>

}]}
```

get

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/ip6tunnel/name_value<String>

Query-parameters:

attrs

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/ip6tunnel/name_value<String>?**attrs=property-name1,property-name2**

Use this query parameter to specify the resource details that you want to retrieve.

view

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/ip6tunnel/name_value<String>?**view=summary**

Note: By default, the retrieved results are displayed in detail view (?view=detail).

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "ip6tunnel": [ {

name:<String_value>,remote:<String_value>"remoteip":<String_value>,

"local":<String_value>,

"type":<Double_value>,

"encapip":<String_value>,

"ownergroup":<String_value>

}]}
```

count

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/ip6tunnel?**count=yes**

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "ip6tunnel": [ { "__count": "#no"} ] }
```

ip6tunnelparam

Configuration for ip6 tunnel parameter resource.

Properties

(click to see Operations)

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
srcip	<String>	Read-write	Common source IPv6 address for all IPv6 tunnels. Must be a SNIP6 or VIP6 address. Minimum length = 1
dropfrag	<String>	Read-write	Drop any packet that requires fragmentation. Default value: NO Possible values = YES, NO
dropfragcputhreshold	<Double>	Read-write	Threshold value, as a percentage of CPU usage, at which to drop packets that require

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
			fragmentation. Applies only if dropFragparameter is set to NO. Minimum value = 1 Maximum value = 100
srciproundrobin	<String>	Read-write	Use a different source IPv6 address for each new session through a particular IPv6 tunnel, as determined by round robin selection of one of the SNIP6 addresses. This setting is ignored if a common global source IPv6 address has been specified for all the IPv6 tunnels. This setting does not apply to a tunnel for which a source IPv6 address has been specified. Default value: NO Possible values = YES, NO
useclientsourceipv6	<String>	Read-write	Use client source IPv6 address as source IPv6 address for outer tunnel IPv6 header. Default value: NO

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
			Possible values = YES, NO

Operations

(click to see Properties)

UPDATE| UNSET| GET (ALL)

Some options that you can use for each operations:

- **Getting warnings in response:**NITRO allows you to get warnings in an operation by specifying the "warning" query parameter as "yes". For example, to get warnings while connecting to the Citrix ADC appliance, the URL is as follows:

  http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/login?warning=yes

  If any, the warnings are displayed in the response payload with the HTTP code "209 X-NITRO-WARNING".

- **Authenticated access for individual NITRO operations:**NITRO allows you to logon to the Citrix ADC appliance to perform individual operations. You can use this option instead of creating a NITRO session (using the login object) and then using that session to perform all operations,

  To do this, you must specify the username and password in the request header of the NITRO request as follows:

  X-NITRO-USER:<username>

  X-NITRO-PASS:<password>

  Note: In such cases, make sure that the request header DOES not include the following:

  Cookie:NITRO_AUTH_TOKEN=<tokenvalue>

`*Note:`*

Mandatory parameters are marked in redand placeholder content is marked in <green>.

update

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/ip6tunnelparam

HTTP Method:PUT

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Content-Type:application/json

Request Payload:

```
{"ip6tunnelparam":{

"srcip":<String_value>,

"dropfrag":<String_value>,

"dropfragcputhreshold":<Double_value>,

"srciproundrobin":<String_value>,

"useclientsourceipv6":<String_value>

}}
```

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error

unset

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/ip6tunnelparam?**action=unset**

HTTP Method:POST

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Content-Type:application/json

Request Payload:

```
{"ip6tunnelparam":{

"srcip":true,

"dropfrag":true,

"dropfragcputhreshold":true,

"srciproundrobin":true,

"useclientsourceipv6":true

}}
```

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error

get (all)

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/ip6tunnelparam

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "ip6tunnelparam": [ {

"srcip":<String_value>,

"dropfrag":<String_value>,

"dropfragcputhreshold":<Double_value>,

"srciproundrobin":<String_value>,

"useclientsourceipv6":<String_value>

}]}
```

ipset

Configuration for network ipset resource.

Properties

(click to see Operations)

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
name	<String>	Read-write	Name for the IP set. Must begin with a letter, number, or the underscore character (_), and can consist of letters, numbers, and the hyphen (-), period (.) pound (#), space ( ), at sign (@), equals (=), colon (:), and underscore characters. Cannot be changed after the IP set is created. Choose a name that

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
			helps identify the IP set. Minimum length = 1
td	<Double>	Read-write	Integer value that uniquely identifies the traffic domain in which you want to configure the entity. If you do not specify an ID, the entity becomes part of the default traffic domain, which has an ID of 0. Minimum value = 0 Maximum value = 4094
__count	<Double>	Read-only	count parameter

Operations

(click to see Properties)

ADD| DELETE| GET (ALL)| GET| COUNT

Some options that you can use for each operations:

- **Getting warnings in response:**NITRO allows you to get warnings in an operation by specifying the "warning" query parameter as "yes". For example, to get warnings while connecting to the Citrix ADC appliance, the URL is as follows:

  http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/login?warning=yes

  If any, the warnings are displayed in the response payload with the HTTP code "209 X-NITRO-WARNING".
- **Authenticated access for individual NITRO operations:**NITRO allows you to logon to the Citrix ADC appliance to perform individual operations. You can use this option instead of

creating a NITRO session (using the login object) and then using that session to perform all operations,

To do this, you must specify the username and password in the request header of the NITRO request as follows:

X-NITRO-USER:<username>

X-NITRO-PASS:<password>

Note: In such cases, make sure that the request header DOES not include the following:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue>

`*Note:`*

Mandatory parameters are marked in redand placeholder content is marked in <green>.

add

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/ipset

HTTP Method:POST

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Content-Type:application/json

Request Payload:

```
{"ipset":{

<b>"name":<String_value>,

</b>"td":<Double_value>

}}
```

Response:

HTTP Status Code on Success: 201 Created HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error

delete

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/ipset/name_value<String>

HTTP Method:DELETE

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue>

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error

get (all)

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/ipset

Query-parameters:

attrs

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/ipset?**attrs=property-name1,property-name2**

Use this query parameter to specify the resource details that you want to retrieve.

filter

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/ipset?**filter=property-name1:property-val1,property-name2:property-val2**

Use this query-parameter to get the filtered set of ipset resources configured on Citrix ADC. Filtering can be done on any of the properties of the resource.

view

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/ipset?**view=summary**

Note: By default, the retrieved results are displayed in detail view (?view=detail).

pagination

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/ipset?**pagesize=#no;pageno=#no**

Use this query-parameter to get the ipset resources in chunks.

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "ipset": [ {

"name":<String_value>,

"td":<Double_value>

}]}
```

get

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/ipset/name_value<String>

Query-parameters:

attrs

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/ipset/name_value<String>?**attrs=property-name1,property-name2**

Use this query parameter to specify the resource details that you want to retrieve.

view

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/ipset/name_value<String>?**view=summary**

Note: By default, the retrieved results are displayed in detail view (?view=detail).

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "ipset": [ {

"name":<String_value>,

"td":<Double_value>

}]}
```

count

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/ipset?**count=yes**

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "ipset": [ { "__count": "#no"} ] }
```

ipset_binding

Binding object which returns the resources bound to ipset.

Properties

(click to see Operations)

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
name	<String>	Read-write	Name of the IP set whose details you want to display. Minimum length = 1
ipset_nsip_binding	<ipset_nsip_binding[]>	Read-only	nsip that can be bound to ipset.
ipset_nsip6_binding	<ipset_nsip6_binding[]>	Read-only	nsip6 that can be bound to ipset.

Operations

(click to see Properties)

GET| GET (ALL)

Some options that you can use for each operations:

- **Getting warnings in response:**NITRO allows you to get warnings in an operation by specifying the "warning" query parameter as "yes". For example, to get warnings while connecting to the Citrix ADC appliance, the URL is as follows:

  http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/login?warning=yes

  If any, the warnings are displayed in the response payload with the HTTP code "209 X-NITRO-WARNING".

- **Authenticated access for individual NITRO operations:**NITRO allows you to logon to the Citrix ADC appliance to perform individual operations. You can use this option instead of creating a NITRO session (using the login object) and then using that session to perform all operations,

  To do this, you must specify the username and password in the request header of the NITRO request as follows:

  X-NITRO-USER:<username>

  X-NITRO-PASS:<password>

  Note: In such cases, make sure that the request header DOES not include the following:

  Cookie:NITRO_AUTH_TOKEN=<tokenvalue>

`*Note:`*

Mandatory parameters are marked in redand placeholder content is marked in <green>.

get

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/ipset_binding/name_value<String>

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "ipset_binding": [ {

"name":<String_value>,

"ipset_nsip_binding":<ipset_nsip_binding[]_value>,

"ipset_nsip6_binding":<ipset_nsip6_binding[]_value>

}]}
```

get (all)

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/ipset_binding

Query-parameters:

bulkbindings

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/ipset_binding?**bulkbindings=yes**

NITRO allows you to fetch bindings in bulk.

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "ipset_binding": [ {

"name":<String_value>,

"ipset_nsip_binding":<ipset_nsip_binding[]_value>,

"ipset_nsip6_binding":<ipset_nsip6_binding[]_value>

}]}
```

ipset_nsip6_binding

Binding object showing the nsip6 that can be bound to ipset.

Properties

(click to see Operations)

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
name	<String>	Read-write	Name of the IP set to which to bind IP addresses. Minimum length = 1
ipaddress	<String>	Read-write	One or more IP addresses bound to the IP set. Minimum length = 1
__count	<Double>	Read-write	count parameter

Operations

(click to see Properties)

ADD:| DELETE:| GET| GET (ALL)| COUNT

Some options that you can use for each operations:

- **Getting warnings in response:**NITRO allows you to get warnings in an operation by specifying the "warning" query parameter as "yes". For example, to get warnings while connecting to the Citrix ADC appliance, the URL is as follows:

  http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/login?warning=yes

  If any, the warnings are displayed in the response payload with the HTTP code "209 X-NITRO-WARNING".

- **Authenticated access for individual NITRO operations:**NITRO allows you to logon to the Citrix ADC appliance to perform individual operations. You can use this option instead of creating a NITRO session (using the login object) and then using that session to perform all operations,

  To do this, you must specify the username and password in the request header of the NITRO request as follows:

  X-NITRO-USER:<username>

  X-NITRO-PASS:<password>

  Note: In such cases, make sure that the request header DOES not include the following:

  Cookie:NITRO_AUTH_TOKEN=<tokenvalue>

`*Note:`*

Mandatory parameters are marked in redand placeholder content is marked in <green>.

add

URL:http://<Citrix-ADC-IP-address(NSIP)/nitro/v1/config/ipset_nsip6_binding

HTTP Method:PUT

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Content-Type:application/json

Request Payload:

```
{
```

```
"ipset_nsip6_binding":{

<b>"name":<String_value>,

</b><b>"ipaddress":<String_value></b>

}}
```

Response:

HTTP Status Code on Success: 201 Created HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error

delete

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/ipset_nsip6_binding/name_value<String>

Query-parameters:

args

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/ipset_nsip6_binding/name_value<String>?**args=ipaddress:<String_value>**

HTTP Method:DELETE

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue>

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error

get

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/ipset_nsip6_binding/name_value<String>

Query-parameters:

filter

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/ipset_nsip6_binding/name_value<String>?**filter=property-name1:property-value1,property-name2:property-value2**

Use this query-parameter to get the filtered set of ipset_nsip6_binding resources configured on Citrix ADC. Filtering can be done on any of the properties of the resource.

pagination

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/ipset_nsip6_binding/name_value<String>?**pagesize=#no;pageno=#no**

Use this query-parameter to get the ipset_nsip6_binding resources in chunks.

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "ipset_nsip6_binding": [ {

"name":<String_value>,

"ipaddress":<String_value>

}]}
```

get (all)

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/ipset_nsip6_binding

Query-parameters:

bulkbindings

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/ipset_nsip6_binding?**bulkbindings=yes**

NITRO allows you to fetch bindings in bulk.

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "ipset_nsip6_binding": [ {

"name":<String_value>,

"ipaddress":<String_value>

}]}
```

count

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/ipset_nsip6_binding/name_value<String>?**count=yes**

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{"ipset_nsip6_binding": [ { "__count": "#no"} ] }
```

ipset_nsip_binding

Binding object showing the nsip that can be bound to ipset.

Properties

(click to see Operations)

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
name	<String>	Read-write	Name of the IP set to which to bind IP addresses. Minimum length = 1
ipaddress	<String>	Read-write	One or more IP addresses bound to the IP set. Minimum length = 1
__count	<Double>	Read-write	count parameter

Operations

(click to see Properties)

ADD:| DELETE:| GET| GET (ALL)| COUNT

Some options that you can use for each operations:

- **Getting warnings in response:**NITRO allows you to get warnings in an operation by specifying the "warning" query parameter as "yes". For example, to get warnings while connecting to the Citrix ADC appliance, the URL is as follows:

  http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/login?warning=yes

  If any, the warnings are displayed in the response payload with the HTTP code "209 X-NITRO-WARNING".

- **Authenticated access for individual NITRO operations:**NITRO allows you to logon to the Citrix ADC appliance to perform individual operations. You can use this option instead of creating a NITRO session (using the login object) and then using that session to perform all operations,

  To do this, you must specify the username and password in the request header of the NITRO request as follows:

  X-NITRO-USER:<username>

  X-NITRO-PASS:<password>

  Note: In such cases, make sure that the request header DOES not include the following:

  Cookie:NITRO_AUTH_TOKEN=<tokenvalue>

`*Note:`*

Mandatory parameters are marked in redand placeholder content is marked in <green>.

add

URL:http://<Citrix-ADC-IP-address(NSIP)/nitro/v1/config/ipset_nsip_binding

HTTP Method:PUT

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Content-Type:application/json

Request Payload:

```
{

"ipset_nsip_binding":{

<b>"name":<String_value>,

</b><b>"ipaddress":<String_value></b>

}}
```

Response:

HTTP Status Code on Success: 201 Created HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error

delete

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/ipset_nsip_binding/name_value<String>

Query-parameters:

args

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/ipset_nsip_binding/name_value<String>?**args=ipaddress:<String_value>**

HTTP Method:DELETE

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue>

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error

get

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/ipset_nsip_binding/name_value<String>

Query-parameters:

filter

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/ipset_nsip_binding/name_value<String>?**filter=property-name1:property-value1,property-name2:property-value2**

Use this query-parameter to get the filtered set of ipset_nsip_binding resources configured on Citrix ADC. Filtering can be done on any of the properties of the resource.

pagination

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/ipset_nsip_binding/name_value<String>?**pagesize=#no;pageno=#no**

Use this query-parameter to get the ipset_nsip_binding resources in chunks.

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "ipset_nsip_binding": [ {

"name":<String_value>,

"ipaddress":<String_value>
```

```
}]}
```

get (all)

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/ipset_nsip_binding

Query-parameters:

bulkbindings

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/ipset_nsip_binding?**bulkbindings=yes**

NITRO allows you to fetch bindings in bulk.

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "ipset_nsip_binding": [ {

"name":<String_value>,

"ipaddress":<String_value>

}]}
```

count

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/ipset_nsip_binding/name_value<String>?**count=yes**

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{"ipset_nsip_binding": [ { "__count": "#no"} ] }
```

iptunnel

Configuration for ip Tunnel resource.

Properties

(click to see Operations)

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
name	<String>	Read-write	Name for the IP tunnel. Leading character must be a number or letter. Other characters allowed, after the first character, are @

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
			_ - . (period) : (colon) # and space ( ). Minimum length = 1
remote	<String>	Read-write	Public IPv4 address, of the remote device, used to set up the tunnel. For this parameter, you can alternatively specify a network address. Minimum length = 1
remotesubnetmask	<String>	Read-write	Subnet mask of the remote IP address of the tunnel.
local	<String>	Read-write	Type of Citrix ADC owned public IPv4 address, configured on the local Citrix ADC and used to set up the tunnel.
protocol	<String>	Read-write	Name of the protocol to be used on this tunnel. Default value: IPIP Possible values = IPIP, GRE, IPSEC, UDP
grepayload	<String>	Read-write	The payload GRE will carry. Default value: ETHERNETwithDOT1Q

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
			Possible values = ETHERNETwithDOT1Q, ETHERNET, IP
ipsecprofilename	<String>	Read-write	Name of IPSec profile to be associated. Default value: "ns_ipsec_default_profile"
vlan	<Double>	Read-write	The vlan for mulicast packets. Minimum value = 1 Maximum value = 4094
ownergroup	<String>	Read-write	The owner node group in a Cluster for the iptunnel. Default value: DEFAULT_NG Minimum length = 1
sysname	<String>	Read-only	The name of the ip tunnel.
type	<Double>	Read-only	The type of this tunnel.
encapip	<String>	Read-only	The effective local IP address of the tunnel. Used as the source of the encapsulated packets.

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
channel	<Double>	Read-only	The tunnel that is bound to a netbridge.
tunneltype	<String[]>	Read-only	Indicates that a tunnel is User-Configured, Internal or DELETE-IN-PROGRESS. Possible values = Configured, Delete-In-Progress
ipsectunnelstatus	<String>	Read-only	Whether the ipsec on this tunnel is up or down. Possible values = DOWN, UP, PARTIAL-UP, UNKNOWN
refcnt	<Double>	Read-only	Number of PBRs to bound to this iptunnel.
__count	<Double>	Read-only	count parameter

Operations

(click to see Properties)

ADD| DELETE| GET (ALL)| GET| COUNT

Some options that you can use for each operations:

- **Getting warnings in response:**NITRO allows you to get warnings in an operation by specifying the "warning" query parameter as "yes". For example, to get warnings while connecting to the Citrix ADC appliance, the URL is as follows:

  http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/login?warning=yes

If any, the warnings are displayed in the response payload with the HTTP code "209 X-NITRO-WARNING".

- **Authenticated access for individual NITRO operations:**NITRO allows you to logon to the Citrix ADC appliance to perform individual operations. You can use this option instead of creating a NITRO session (using the login object) and then using that session to perform all operations,

  To do this, you must specify the username and password in the request header of the NITRO request as follows:

  X-NITRO-USER:<username>

  X-NITRO-PASS:<password>

  Note: In such cases, make sure that the request header DOES not include the following:

  Cookie:NITRO_AUTH_TOKEN=<tokenvalue>

`*Note:`*

Mandatory parameters are marked in redand placeholder content is marked in <green>.

add

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/iptunnel

HTTP Method:POST

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Content-Type:application/json

Request Payload:

```
{"iptunnel":{

<b>"name":<String_value>,

</b><b>"remote":<String_value>,

</b><b>"remotesubnetmask":<String_value>,
```

```
</b><b>"local":<String_value>,

</b>"protocol":<String_value>,

"grepayload":<String_value>,

"ipsecprofilename":<String_value>,

"vlan":<Double_value>,

"ownergroup":<String_value>

}}
```

Response:

HTTP Status Code on Success: 201 Created HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error

delete

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/iptunnel/name_value<String>

HTTP Method:DELETE

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue>

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error

get (all)

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/iptunnel

Query-parameters:

args

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/iptunnel?**args=remote:<String_value>,remotesubnetmask:<String_value>,name:<String_value>**

Use this query-parameter to get iptunnel resources based on additional properties.

attrs

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/iptunnel?**attrs=property-name1,property-name2**

Use this query parameter to specify the resource details that you want to retrieve.

filter

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/iptunnel?**filter=property-name1:property-val1,property-name2:property-val2**

Use this query-parameter to get the filtered set of iptunnel resources configured on Citrix ADC. Filtering can be done on any of the properties of the resource.

view

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/iptunnel?**view=summary**

Note: By default, the retrieved results are displayed in detail view (?view=detail).

pagination

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/iptunnel?**pagesize=#no;pageno=#no**

Use this query-parameter to get the iptunnel resources in chunks.

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "iptunnel": [ {

remote:<String_value>,remotesubnetmask:<String_value>,name:<String_value>"
sysname":<String_value>,

"local":<String_value>,

"protocol":<String_value>,

"grepayload":<String_value>,

"type":<Double_value>,

"encapip":<String_value>,

"channel":<Double_value>,

"ipsecprofilename":<String_value>,

"vlan":<Double_value>,

"tunneltype":<String[]_value>,

"ipsectunnelstatus":<String_value>,

"ownergroup":<String_value>,

"refcnt":<Double_value>

}]}
```

get

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/iptunnel/name_value<String>

Query-parameters:

attrs

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/iptunnel/name_value<String>?**attrs=property-name1,property-name2**

Use this query parameter to specify the resource details that you want to retrieve.

view

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/iptunnel/name_value<String>?**view=summary**

Note: By default, the retrieved results are displayed in detail view (?view=detail).

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "iptunnel": [ {

remote:<String_value>,remotesubnetmask:<String_value>,name:<String_value>"
sysname":<String_value>,

"local":<String_value>,

"protocol":<String_value>,

"grepayload":<String_value>,

"type":<Double_value>,

"encapip":<String_value>,
```

```
"channel":<Double_value>,

"ipsecprofilename":<String_value>,

"vlan":<Double_value>,

"tunneltype":<String[]_value>,

"ipsectunnelstatus":<String_value>,

"ownergroup":<String_value>,

"refcnt":<Double_value>

}]}
```

count

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/iptunnel?**count=yes**

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "iptunnel": [ { "__count": "#no"} ] }
```

iptunnelparam

Configuration for ip tunnel parameter resource.

Properties

(click to see Operations)

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
srcip	<String>	Read-write	Common source-IP address for all tunnels. For a specific tunnel, this global setting is overridden if you have specified another source IP address. Must be a MIP or SNIP address. Minimum length = 1
dropfrag	<String>	Read-write	Drop any IP packet that requires fragmentation before it is sent through the tunnel. Default value: NO Possible values = YES, NO
dropfragcputhreshold	<Double>	Read-write	Threshold value, as a percentage of CPU usage, at which to drop packets that require fragmentation to use the IP tunnel. Applies only if dropFragparameter is set to NO. The default value, 0,

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
			specifies that this parameter is not set. Minimum value = 1 Maximum value = 100
srciproundrobin	<String>	Read-write	Use a different source IP address for each new session through a particular IP tunnel, as determined by round robin selection of one of the SNIP addresses. This setting is ignored if a common global source IP address has been specified for all the IP tunnels. This setting does not apply to a tunnel for which a source IP address has been specified. Default value: NO Possible values = YES, NO
enablestrictrx	<String>	Read-write	Strict PBR check for IPSec packets received through tunnel. Default value: NO Possible values = YES, NO

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
enablestricttx	<String>	Read-write	Strict PBR check for packets to be sent IPSec protected. Default value: NO Possible values = YES, NO
mac	<String>	Read-write	The shared MAC used for shared IP between cluster nodes/HA peers.
useclientsourceip	<String>	Read-write	Use client source IP as source IP for outer tunnel IP header. Default value: NO Possible values = YES, NO

Operations

(click to see Properties)

UPDATE| UNSET| GET (ALL)

Some options that you can use for each operations:

- **Getting warnings in response:**NITRO allows you to get warnings in an operation by specifying the "warning" query parameter as "yes". For example, to get warnings while connecting to the Citrix ADC appliance, the URL is as follows:

  http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/login?warning=yes

  If any, the warnings are displayed in the response payload with the HTTP code "209 X-NITRO-WARNING".

- **Authenticated access for individual NITRO operations:**NITRO allows you to logon to the Citrix ADC appliance to perform individual operations. You can use this option instead of

creating a NITRO session (using the login object) and then using that session to perform all operations,

To do this, you must specify the username and password in the request header of the NITRO request as follows:

X-NITRO-USER:<username>

X-NITRO-PASS:<password>

Note: In such cases, make sure that the request header DOES not include the following:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue>

`*Note:`*

Mandatory parameters are marked in redand placeholder content is marked in <green>.

update

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/iptunnelparam

HTTP Method:PUT

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Content-Type:application/json

Request Payload:

```
{"iptunnelparam":{

"srcip":<String_value>,

"dropfrag":<String_value>,

"dropfragcputhreshold":<Double_value>,

"srciproundrobin":<String_value>,

"enablestrictrx":<String_value>,

"enablestricttx":<String_value>,
```

```
"mac":<String_value>,

"useclientsourceip":<String_value>

}}
```

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error

unset

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/iptunnelparam?**action=unset**

HTTP Method:POST

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Content-Type:application/json

Request Payload:

```
{"iptunnelparam":{

"srcip":true,

"dropfrag":true,

"dropfragcputhreshold":true,

"srciproundrobin":true,

"enablestrictrx":true,

"enablestricttx":true,

"mac":true,
```

```
"useclientsourceip":true

}}
```

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error

get (all)

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/iptunnelparam

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "iptunnelparam": [ {

"srcip":<String_value>,

"dropfrag":<String_value>,

"dropfragcputhreshold":<Double_value>,

"srciproundrobin":<String_value>,
```

```
"enablestrictrx":<String_value>,

"enablestricttx":<String_value>,

"mac":<String_value>,

"useclientsourceip":<String_value>

}]}
```

ipv6

Configuration for ip v6 resource.

Properties

(click to see Operations)

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
ralearning	<String>	Read-write	Enable the Citrix ADC to learn about various routes from Router Advertisement (RA) and Router Solicitation (RS) messages sent by the routers. Default value: DISABLED Possible values = ENABLED, DISABLED
routerredirection	<String>	Read-write	Enable the Citrix ADC to do Router Redirection.

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
			Default value: DISABLED Possible values = ENABLED, DISABLED
ndbasereachtime	<Double>	Read-write	Base reachable time of the Neighbor Discovery (ND6) protocol. The time, in milliseconds, that the Citrix ADC assumes an adjacent device is reachable after receiving a reachability confirmation. Default value: 30000 Minimum value = 1
ndretransmissiontime	<Double>	Read-write	Retransmission time of the Neighbor Discovery (ND6) protocol. The time, in milliseconds, between retransmitted Neighbor Solicitation (NS) messages, to an adjacent device. Default value: 1000 Minimum value = 1
natprefix	<String>	Read-write	Prefix used for translating packets from private IPv6 servers to IPv4

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
			packets. This prefix has a length of 96 bits (128-32 = 96). The IPv6 servers embed the destination IP address of the IPv4 servers or hosts in the last 32 bits of the destination IP address field of the IPv6 packets. The first 96 bits of the destination IP address field are set as the IPv6 NAT prefix. IPv6 packets addressed to this prefix have to be routed to the Citrix ADC to ensure that the IPv6-IPv4 translation is done by the appliance.
td	<Double>	Read-write	Integer value that uniquely identifies the traffic domain in which you want to configure the entity. If you do not specify an ID, the entity becomes part of the default traffic domain, which has an ID of 0. Default value: 0 Minimum value = 0

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
			Maximum value = 4094
dodad	<String>	Read-write	Enable the Citrix ADC to do Duplicate Address Detection (DAD) for all the Citrix ADC owned IPv6 addresses regardless of whether they are obtained through stateless auto configuration, DHCPv6, or manual configuration. Default value: DISABLED Possible values = ENABLED, DISABLED
usipnatprefix	<String>	Read-write	IPV6 NATPREFIX used in NAT46 scenario when USIP is turned on.
basereachtime	<Integer>	Read-only	ND6 base reachable time (ms).
reachtime	<Integer>	Read-only	ND6 computed reachable time (ms).
ndreachtime	<Double>	Read-only	ND6 computed reachable time (ms).

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
retransmissiontime	<Integer>	Read-only	ND6 retransmission time (ms).
__count	<Double>	Read-only	count parameter

Operations

(click to see Properties)

UPDATE| UNSET| GET (ALL)| GET| COUNT

Some options that you can use for each operations:

- **Getting warnings in response:**NITRO allows you to get warnings in an operation by specifying the "warning" query parameter as "yes". For example, to get warnings while connecting to the Citrix ADC appliance, the URL is as follows:

  http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/login?warning=yes

  If any, the warnings are displayed in the response payload with the HTTP code "209 X-NITRO-WARNING".

- **Authenticated access for individual NITRO operations:**NITRO allows you to logon to the Citrix ADC appliance to perform individual operations. You can use this option instead of creating a NITRO session (using the login object) and then using that session to perform all operations,

  To do this, you must specify the username and password in the request header of the NITRO request as follows:

  X-NITRO-USER:<username>

  X-NITRO-PASS:<password>

  Note: In such cases, make sure that the request header DOES not include the following:

  Cookie:NITRO_AUTH_TOKEN=<tokenvalue>

`*Note:`*

Mandatory parameters are marked in redand placeholder content is marked in <green>.

update

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/ipv6

HTTP Method:PUT

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Content-Type:application/json

Request Payload:

```
{"ipv6":{

"ralearning":<String_value>,

"routerredirection":<String_value>,

"ndbasereachtime":<Double_value>,

"ndretransmissiontime":<Double_value>,

"natprefix":<String_value>,

"td":<Double_value>,

"dodad":<String_value>,

"usipnatprefix":<String_value>

}}
```

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error

unset

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/ipv6?**action=unset**

HTTP Method:POST

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Content-Type:application/json

Request Payload:

```
{"ipv6":{

"ralearning":true,

"routerredirection":true,

"ndbasereachtime":true,

"ndretransmissiontime":true,

"natprefix":true,

"td":<Double_value>,

"dodad":true,

"usipnatprefix":true

}}
```

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error

get (all)

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/ipv6

Query-parameters:

attrs

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/ipv6?**attrs=property-name1,property-name2**

Use this query parameter to specify the resource details that you want to retrieve.

filter

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/ipv6?**filter=property-name1:property-val1,property-name2:property-val2**

Use this query-parameter to get the filtered set of ipv6 resources configured on Citrix ADC. Filtering can be done on any of the properties of the resource.

view

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/ipv6?**view=summary**

Note: By default, the retrieved results are displayed in detail view (?view=detail).

pagination

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/ipv6?**pagesize=#no;pageno=#no**

Use this query-parameter to get the ipv6 resources in chunks.

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "ipv6": [ {
```

```
"td":<Double_value>,

"ralearning":<String_value>,

"routerredirection":<String_value>,

"basereachtime":<Integer_value>,

"ndbasereachtime":<Double_value>,

"reachtime":<Integer_value>,

"ndreachtime":<Double_value>,

"retransmissiontime":<Integer_value>,

"ndretransmissiontime":<Double_value>,

"natprefix":<String_value>,

"dodad":<String_value>,

"usipnatprefix":<String_value>

}]}
```

get

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/ipv6/td_value<Double>

Query-parameters:

attrs

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/ipv6/td_value<Double>?**attrs=property-name1,property-name2**

Use this query parameter to specify the resource details that you want to retrieve.

view

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/ipv6/td_value<Double>?**view=summary**

Note: By default, the retrieved results are displayed in detail view (?view=detail).

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "ipv6": [ {

"td":<Double_value>,

"ralearning":<String_value>,

"routerredirection":<String_value>,

"basereachtime":<Integer_value>,

"ndbasereachtime":<Double_value>,

"reachtime":<Integer_value>,

"ndreachtime":<Double_value>,

"retransmissiontime":<Integer_value>,

"ndretransmissiontime":<Double_value>,

"natprefix":<String_value>,

"dodad":<String_value>,
```

```
"usipnatprefix":<String_value>

}]}
```

count

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/ipv6?**count=yes**

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "ipv6": [ { "__count": "#no"} ] }
```

l2param

Configuration for Layer 2 related parameter resource.

Properties

(click to see Operations)

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
mbfpeermacupdate	<Double>	Read-write	When mbf_instant_learning is enabled, learn any

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
			changes in peer's MAC after this time interval, which is in 10ms ticks. Default value: 10
maxbridgecollision	<Double>	Read-write	Maximum bridge collision for loop detection . Default value: 20
bdggrpproxyarp	<String>	Read-write	Set/reset proxy ARP in bridge group deployment. Default value: ENABLED Possible values = ENABLED, DISABLED
bdgsetting	<String>	Read-write	Bridging settings for C2C behavior. If enabled, each PE will learn MAC entries independently. Otherwise, when L2 mode is ON, learned MAC entries on a PE will be broadcasted to all other PEs. Default value: DISABLED Possible values = ENABLED, DISABLED

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
garponvridintf	<String>	Read-write	Send GARP messagess on VRID-configured interfaces upon failover . Default value: ENABLED Possible values = ENABLED, DISABLED
macmodefwdmypkt	<String>	Read-write	Allows MAC mode vserver to pick and forward the packets even if it is destined to Citrix ADC owned VIP. Default value: DISABLED Possible values = ENABLED, DISABLED
usemymac	<String>	Read-write	Use Citrix ADC MAC for all outgoing packets. Default value: DISABLED Possible values = ENABLED, DISABLED
proxyarp	<String>	Read-write	Proxies the ARP as Citrix ADC MAC for FreeBSD.

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
			Default value: ENABLED Possible values = ENABLED, DISABLED
garpreply	<String>	Read-write	Set/reset REPLY form of GARP . Default value: DISABLED Possible values = ENABLED, DISABLED
mbfinstlearning	<String>	Read-write	Enable instant learning of MAC changes in MBF mode. Default value: DISABLED Possible values = ENABLED, DISABLED
rstintfonhafo	<String>	Read-write	Enable the reset interface upon HA failover. Default value: DISABLED Possible values = ENABLED, DISABLED
skipproxyingbsdtraffic	<String>	Read-write	Control source parameters (IP and Port) for FreeBSD

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
			initiated traffic. If Enabled, source parameters are retained. Else proxy the source parameters based on next hop. Default value: DISABLED Possible values = ENABLED, DISABLED
returntoethernetsender	<String>	Read-write	Return to ethernet sender. Default value: DISABLED Possible values = ENABLED, DISABLED
stopmacmoveupdate	<String>	Read-write	Stop Update of server mac change to NAT sessions. Default value: DISABLED Possible values = ENABLED, DISABLED
bridgeagetimeout	<Double>	Read-write	Time-out value for the bridge table entries, in seconds. The new value applies only to the entries that are dynamically learned

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
			after the new value is set. Previously existing bridge table entries expire after the previously configured time-out value. Default value: 300 Minimum value = 60 Maximum value = 300

Operations

(click to see Properties)

UPDATE| UNSET| GET (ALL)

Some options that you can use for each operations:

- **Getting warnings in response:**NITRO allows you to get warnings in an operation by specifying the "warning" query parameter as "yes". For example, to get warnings while connecting to the Citrix ADC appliance, the URL is as follows:

  http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/login?warning=yes

  If any, the warnings are displayed in the response payload with the HTTP code "209 X-NITRO-WARNING".

- **Authenticated access for individual NITRO operations:**NITRO allows you to logon to the Citrix ADC appliance to perform individual operations. You can use this option instead of creating a NITRO session (using the login object) and then using that session to perform all operations,

  To do this, you must specify the username and password in the request header of the NITRO request as follows:

  X-NITRO-USER:<username>

  X-NITRO-PASS:<password>

Note: In such cases, make sure that the request header DOES not include the following:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue>

`*Note:` *

Mandatory parameters are marked in redand placeholder content is marked in <green>.

update

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/l2param

HTTP Method:PUT

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Content-Type:application/json

Request Payload:

```
{"l2param":{

"mbfpeermacupdate":<Double_value>,

"maxbridgecollision":<Double_value>,

"bdggrpproxyarp":<String_value>,

"bdgsetting":<String_value>,

"garponvridintf":<String_value>,

"macmodefwdmypkt":<String_value>,

"usemymac":<String_value>,

"proxyarp":<String_value>,

"garpreply":<String_value>,

"mbfinstlearning":<String_value>,

"rstintfonhafo":<String_value>,
```

```
"skipproxyingbsdtraffic":<String_value>,

"returntoethernetsender":<String_value>,

"stopmacmoveupdate":<String_value>,

"bridgeagetimeout":<Double_value>

}}
```

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error

unset

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/l2param?**action=unset**

HTTP Method:POST

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Content-Type:application/json

Request Payload:

```
{"l2param":{

"mbfpeermacupdate":true,

"maxbridgecollision":true,

"bdggrpproxyarp":true,

"bdgsetting":true,

"garponvridintf":true,
```

```
"macmodefwdmypkt":true,

"usemymac":true,

"proxyarp":true,

"garpreply":true,

"mbfinstlearning":true,

"rstintfonhafo":true,

"skipproxyingbsdtraffic":true,

"returntoethernetsender":true,

"stopmacmoveupdate":true,

"bridgeagetimeout":true

}}
```

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error

get (all)

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/l2param

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "l2param": [ {

"maxbridgecollision":<Double_value>,

"mbfpeermacupdate":<Double_value>,

"bdggrpproxyarp":<String_value>,

"bdgsetting":<String_value>,

"garponvridintf":<String_value>,

"macmodefwdmypkt":<String_value>,

"usemymac":<String_value>,

"proxyarp":<String_value>,

"garpreply":<String_value>,

"mbfinstlearning":<String_value>,

"rstintfonhafo":<String_value>,

"skipproxyingbsdtraffic":<String_value>,

"returntoethernetsender":<String_value>,

"stopmacmoveupdate":<String_value>,

"bridgeagetimeout":<Double_value>

}]}
```

l3param

Configuration for Layer 3 related parameter resource.

Properties

(click to see Operations)

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
srcnat	<String>	Read-write	Perform NAT if only the source is in the private network. Default value: ENABLED Possible values = ENABLED, DISABLED
icmpgenratethreshold	<Double>	Read-write	NS generated ICMP pkts per 10ms rate threshold. Default value: 100
overridernat	<String>	Read-write	USNIP/USIP settings override RNAT settings for configured service/virtual server traffic.. . Default value: DISABLED Possible values = ENABLED, DISABLED

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
dropdfflag	<String>	Read-write	Enable dropping the IP DF flag. Default value: DISABLED Possible values = ENABLED, DISABLED
miproundrobin	<String>	Read-write	Enable round robin usage of mapped IPs. Default value: ENABLED Possible values = ENABLED, DISABLED
externalloopback	<String>	Read-write	Enable external loopback. Default value: DISABLED Possible values = ENABLED, DISABLED
tnlpmtuwoconn	<String>	Read-write	Enable/Disable learning PMTU of IP tunnel when ICMP error does not contain connection information. Default value: ENABLED

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
			Possible values = ENABLED, DISABLED
usipserverstraypkt	<String>	Read-write	Enable detection of stray server side pkts in USIP mode. Default value: DISABLED Possible values = ENABLED, DISABLED
forwardicmpfragments	<String>	Read-write	Enable forwarding of ICMP fragments. Default value: DISABLED Possible values = ENABLED, DISABLED
dropipfragments	<String>	Read-write	Enable dropping of IP fragments. Default value: DISABLED Possible values = ENABLED, DISABLED
acllogtime	<Double>	Read-write	Parameter to tune acl logging time. Default value: 5000
implicitaclallow	<String>	Read-write	Do not apply ACLs for internal ports.

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
			Default value: ENABLED Possible values = ENABLED, DISABLED
dynamicrouting	<String>	Read-write	Enable/Disable Dynamic routing on partition. This configuration is not applicable to default partition. Default value: DISABLED Possible values = ENABLED, DISABLED
ipv6dynamicrouting	<String>	Read-write	Enable/Disable IPv6 Dynamic routing. Default value: DISABLED Possible values = ENABLED, DISABLED
allowclasseipv4	<String>	Read-write	Enable/Disable IPv4 Class E address clients. Default value: DISABLED Possible values = ENABLED, DISABLED

Operations

(click to see Properties)

UPDATE| UNSET| GET (ALL)

Some options that you can use for each operations:

- **Getting warnings in response:**NITRO allows you to get warnings in an operation by specifying the "warning" query parameter as "yes". For example, to get warnings while connecting to the Citrix ADC appliance, the URL is as follows:

  http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/login?warning=yes

  If any, the warnings are displayed in the response payload with the HTTP code "209 X-NITRO-WARNING".

- **Authenticated access for individual NITRO operations:**NITRO allows you to logon to the Citrix ADC appliance to perform individual operations. You can use this option instead of creating a NITRO session (using the login object) and then using that session to perform all operations,

  To do this, you must specify the username and password in the request header of the NITRO request as follows:

  X-NITRO-USER:<username>

  X-NITRO-PASS:<password>

  Note: In such cases, make sure that the request header DOES not include the following:

  Cookie:NITRO_AUTH_TOKEN=<tokenvalue>

`*Note:`*

Mandatory parameters are marked in redand placeholder content is marked in <green>.

update

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/l3param

HTTP Method:PUT

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Content-Type:application/json

Request Payload:

```
{"l3param":{
"srcnat":<String_value>,
"icmpgenratethreshold":<Double_value>,
"overridernat":<String_value>,
"dropdfflag":<String_value>,
"miproundrobin":<String_value>,
"externalloopback":<String_value>,
"tnlpmtuwoconn":<String_value>,
"usipserverstraypkt":<String_value>,
"forwardicmpfragments":<String_value>,
"dropipfragments":<String_value>,
"acllogtime":<Double_value>,
"implicitaclallow":<String_value>,
"dynamicrouting":<String_value>,
"ipv6dynamicrouting":<String_value>,
"allowclasseipv4":<String_value>
}}
```

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error

unset

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/l3param?**action=unset**

HTTP Method:POST

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Content-Type:application/json

Request Payload:

```
{"l3param":{

"srcnat":true,

"icmpgenratethreshold":true,

"overridernat":true,

"dropdfflag":true,

"miproundrobin":true,

"externalloopback":true,

"tnlpmtuwoconn":true,

"usipserverstraypkt":true,

"forwardicmpfragments":true,

"dropipfragments":true,

"acllogtime":true,

"implicitaclallow":true,
```

```
"dynamicrouting":true,

"ipv6dynamicrouting":true,

"allowclasseipv4":true

}}
```

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error

get (all)

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/l3param

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "l3param": [ {

"srcnat":<String_value>,

"icmpgenratethreshold":<Double_value>,
```

```
"overridernat":<String_value>,

"dropdfflag":<String_value>,

"miproundrobin":<String_value>,

"externalloopback":<String_value>,

"tnlpmtuwoconn":<String_value>,

"usipserverstraypkt":<String_value>,

"forwardicmpfragments":<String_value>,

"dropipfragments":<String_value>,

"acllogtime":<Double_value>,

"implicitaclallow":<String_value>,

"dynamicrouting":<String_value>,

"ipv6dynamicrouting":<String_value>,

"allowclasseipv4":<String_value>

}]}
```

l4param

Configuration for Layer 4 related parameter resource.

Properties

(click to see Operations)

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
l2connmethod	<String>	Read-write	Layer 2 connection method based on the combination of channel number,

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
			MAC address and VLAN. It is tuned with l2conn param of lb vserver. If l2conn of lb vserver is ON then method specified here will be used to identify a connection in addition to the 4-tuple (<source IP>:<source port>::<destination IP>:<destination port>). Default value: MacVlanChannel Possible values = Channel, Vlan, VlanChannel, Mac, MacChannel, MacVlan, MacVlanChannel
l4switch	<String>	Read-write	In L4 switch topology, always clients and servers are on the same side. Enable l4switch to allow such connections. Default value: DISABLED Possible values = ENABLED, DISABLED

Operations

(click to see Properties)

UPDATE| UNSET| GET (ALL)

Some options that you can use for each operations:

- **Getting warnings in response:**NITRO allows you to get warnings in an operation by specifying the "warning" query parameter as "yes". For example, to get warnings while connecting to the Citrix ADC appliance, the URL is as follows:

  http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/login?warning=yes

  If any, the warnings are displayed in the response payload with the HTTP code "209 X-NITRO-WARNING".

- **Authenticated access for individual NITRO operations:**NITRO allows you to logon to the Citrix ADC appliance to perform individual operations. You can use this option instead of creating a NITRO session (using the login object) and then using that session to perform all operations,

  To do this, you must specify the username and password in the request header of the NITRO request as follows:

  X-NITRO-USER:<username>

  X-NITRO-PASS:<password>

  Note: In such cases, make sure that the request header DOES not include the following:

  Cookie:NITRO_AUTH_TOKEN=<tokenvalue>

`*Note:`*

Mandatory parameters are marked in redand placeholder content is marked in <green>.

update

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/l4param

HTTP Method:PUT

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Content-Type:application/json

Request Payload:

```
{"l4param":{

"l2connmethod":<String_value>,

"l4switch":<String_value>

}}
```

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error

unset

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/l4param?**action=unset**

HTTP Method:POST

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Content-Type:application/json

Request Payload:

```
{"l4param":{

"l2connmethod":true,

"l4switch":true

}}
```

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error

get (all)

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/l4param

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "l4param": [ {

"l2connmethod":<String_value>,

"l4switch":<String_value>

}]}
```

lacp

Configuration for Link aggregation control protocol resource.

Properties

(click to see Operations)

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
syspriority	<Double>	Read-write	Priority number that determines which peer device of an LACP LA channel can have control over the LA channel. This parameter is globally applied to all LACP channels on the Citrix ADC. The lower the number, the higher the priority. Default value: 32768 Minimum value = 1 Maximum value = 65535
ownernode	<Double>	Read-write	The owner node in a cluster for which we want to set the lacp priority. Owner node can vary from 0 to 31. Ownernode value of 254 is used for Cluster. Default value: 255
devicename	<String>	Read-only	Name of the channel.
mac	<String>	Read-only	LACP system MAC.
flags	<Double>	Read-only	Flags of this channel.
lacpkey	<Double>	Read-only	LACP key of this channel.

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
clustersyspriority	<Double>	Read-only	LACP system (Cluster) priority.
clustermac	<String>	Read-only	LACP system (Cluster) mac.
__count	<Double>	Read-only	count parameter

Operations

(click to see Properties)

UPDATE| GET (ALL)| GET| COUNT

Some options that you can use for each operations:

- **Getting warnings in response:**NITRO allows you to get warnings in an operation by specifying the "warning" query parameter as "yes". For example, to get warnings while connecting to the Citrix ADC appliance, the URL is as follows:

  http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/login?warning=yes

  If any, the warnings are displayed in the response payload with the HTTP code "209 X-NITRO-WARNING".

- **Authenticated access for individual NITRO operations:**NITRO allows you to logon to the Citrix ADC appliance to perform individual operations. You can use this option instead of creating a NITRO session (using the login object) and then using that session to perform all operations,

  To do this, you must specify the username and password in the request header of the NITRO request as follows:

  X-NITRO-USER:<username>

  X-NITRO-PASS:<password>

  Note: In such cases, make sure that the request header DOES not include the following:

  Cookie:NITRO_AUTH_TOKEN=<tokenvalue>

`*Note:`*

Mandatory parameters are marked in redand placeholder content is marked in <green>.

update

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/lacp

HTTP Method:PUT

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Content-Type:application/json

Request Payload:

```
{"lacp":{

<b>"syspriority":<Double_value>,

</b>"ownernode":<Double_value>

}}
```

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error

get (all)

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/lacp

Query-parameters:

attrs

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/lacp?**attrs=property-name1,property-name2**

Use this query parameter to specify the resource details that you want to retrieve.

filter

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/lacp?**filter=property-name1:property-val1,property-name2:property-val2**

Use this query-parameter to get the filtered set of lacp resources configured on Citrix ADC. Filtering can be done on any of the properties of the resource.

view

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/lacp?**view=summary**

Note: By default, the retrieved results are displayed in detail view (?view=detail).

pagination

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/lacp?**pagesize=#no;pageno=#no**

Use this query-parameter to get the lacp resources in chunks.

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "lacp": [ {

"ownernode":<Double_value>,

"devicename":<String_value>,

"syspriority":<Double_value>,

"mac":<String_value>,
```

```
"flags":<Double_value>,

"lacpkey":<Double_value>,

"clustersyspriority":<Double_value>,

"clustermac":<String_value>

}]}
```

get

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/lacp/ownernode_value<Double>

Query-parameters:

attrs

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/lacp/ownernode_value<Double>?**attrs=property-name1,property-name2**

Use this query parameter to specify the resource details that you want to retrieve.

view

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/lacp/ownernode_value<Double>?**view=summary**

Note: By default, the retrieved results are displayed in detail view (?view=detail).

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "lacp": [ {

"ownernode":<Double_value>,

"devicename":<String_value>,

"syspriority":<Double_value>,

"mac":<String_value>,

"flags":<Double_value>,

"lacpkey":<Double_value>,

"clustersyspriority":<Double_value>,

"clustermac":<String_value>

}]}
```

count

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/lacp?**count=yes**

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "lacp": [ { "__count": "#no"} ] }
```

linkset

Configuration for link set resource.

Properties

(click to see Operations)

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
id	<String>	Read-write	Unique identifier for the linkset. Must be of the form LS/x, where x can be an integer from 1 to 32.
ifnum	<String>	Read-only	The interfaces to be bound to the linkset.
__count	<Double>	Read-only	count parameter

Operations

(click to see Properties)

ADD| DELETE| GET (ALL)| GET| COUNT

Some options that you can use for each operations:

- **Getting warnings in response:**NITRO allows you to get warnings in an operation by specifying the "warning" query parameter as "yes". For example, to get warnings while connecting to the Citrix ADC appliance, the URL is as follows:

  http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/login?warning=yes

  If any, the warnings are displayed in the response payload with the HTTP code "209 X-NITRO-WARNING".

- **Authenticated access for individual NITRO operations:**NITRO allows you to logon to the Citrix ADC appliance to perform individual operations. You can use this option instead of creating a NITRO session (using the login object) and then using that session to perform all operations,

  To do this, you must specify the username and password in the request header of the NITRO request as follows:

  X-NITRO-USER:<username>

  X-NITRO-PASS:<password>

  Note: In such cases, make sure that the request header DOES not include the following:

  Cookie:NITRO_AUTH_TOKEN=<tokenvalue>

`*Note:`*

Mandatory parameters are marked in redand placeholder content is marked in <green>.

add

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/linkset

HTTP Method:POST

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Content-Type:application/json

Request Payload:

```
{"linkset":{

<b>"id":<String_value>

</b>}}
```

Response:

HTTP Status Code on Success: 201 Created HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error

delete

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/linkset/id_value<String>

HTTP Method:DELETE

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue>

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error

get (all)

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/linkset

Query-parameters:

attrs

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/linkset?**attrs=property-name1,property-name2**

Use this query parameter to specify the resource details that you want to retrieve.

filter

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/linkset?**filter=property-name1:property-val1,property-name2:property-val2**

Use this query-parameter to get the filtered set of linkset resources configured on Citrix ADC. Filtering can be done on any of the properties of the resource.

view

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/linkset?**view=summary**

Note: By default, the retrieved results are displayed in detail view (?view=detail).

pagination

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/linkset?**pagesize=#no;pageno=#no**

Use this query-parameter to get the linkset resources in chunks.

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "linkset": [ {

"id":<String_value>,

"ifnum":<String_value>

}]}
```

get

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/linkset/id_value<String>

Query-parameters:

attrs

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/linkset/id_value<String>?**attrs=property-name1,property-name2**

Use this query parameter to specify the resource details that you want to retrieve.

view

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/linkset/id_value<String>?**view=summary**

Note: By default, the retrieved results are displayed in detail view (?view=detail).

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "linkset": [ {

"id":<String_value>,

"ifnum":<String_value>

}]}
```

count

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/linkset?**count=yes**

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "linkset": [ { "__count": "#no"} ] }
```

linkset_binding

Binding object which returns the resources bound to linkset.

Properties

(click to see Operations)

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
id	<String>	Read-write	ID of the linkset for which to display information. If an ID is not provided, the display includes information about all linksets that are available in the cluster.
linkset_channel_binding	<linkset_channel_binding[]>	Read-only	channel that can be bound to linkset.
linkset_interface_binding	<linkset_interface_binding[]>	Read-only	interface that can be bound to linkset.

Operations

(click to see Properties)

GET| GET (ALL)

Some options that you can use for each operations:

- **Getting warnings in response:**NITRO allows you to get warnings in an operation by specifying the "warning" query parameter as "yes". For example, to get warnings while connecting to the Citrix ADC appliance, the URL is as follows:

  http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/login?warning=yes

  If any, the warnings are displayed in the response payload with the HTTP code "209 X-NITRO-WARNING".

- **Authenticated access for individual NITRO operations:**NITRO allows you to logon to the Citrix ADC appliance to perform individual operations. You can use this option instead of creating a NITRO session (using the login object) and then using that session to perform all operations,

  To do this, you must specify the username and password in the request header of the NITRO request as follows:

  X-NITRO-USER:<username>

  X-NITRO-PASS:<password>

  Note: In such cases, make sure that the request header DOES not include the following:

  Cookie:NITRO_AUTH_TOKEN=<tokenvalue>

`*Note:`*

Mandatory parameters are marked in redand placeholder content is marked in <green>.

get

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/linkset_binding/id_value<String>

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "linkset_binding": [ {

"id":<String_value>,

"linkset_channel_binding":<linkset_channel_binding[]_value>,

"linkset_interface_binding":<linkset_interface_binding[]_value>

}]}
```

get (all)

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/linkset_binding

Query-parameters:

bulkbindings

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/linkset_binding?**bulkbindings=yes**

NITRO allows you to fetch bindings in bulk.

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "linkset_binding": [ {

"id":<String_value>,

"linkset_channel_binding":<linkset_channel_binding[]_value>,

"linkset_interface_binding":<linkset_interface_binding[]_value>

}]}
```

linkset_channel_binding

Binding object showing the channel that can be bound to linkset.

Properties

(click to see Operations)

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
ifnum	<String>	Read-write	The interfaces to be bound to the linkset.
id	<String>	Read-write	ID of the linkset to which to bind the interfaces.
__count	<Double>	Read-write	count parameter

Operations

(click to see Properties)

ADD:| DELETE:| GET| GET (ALL)| COUNT

Some options that you can use for each operations:

- **Getting warnings in response:**NITRO allows you to get warnings in an operation by specifying the "warning" query parameter as "yes". For example, to get warnings while connecting to the Citrix ADC appliance, the URL is as follows:

  http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/login?warning=yes

  If any, the warnings are displayed in the response payload with the HTTP code "209 X-NITRO-WARNING".

- **Authenticated access for individual NITRO operations:**NITRO allows you to logon to the Citrix ADC appliance to perform individual operations. You can use this option instead of creating a NITRO session (using the login object) and then using that session to perform all operations,

  To do this, you must specify the username and password in the request header of the NITRO request as follows:

  X-NITRO-USER:<username>

  X-NITRO-PASS:<password>

  Note: In such cases, make sure that the request header DOES not include the following:

  Cookie:NITRO_AUTH_TOKEN=<tokenvalue>

`*Note:`*

Mandatory parameters are marked in redand placeholder content is marked in <green>.

add

URL:http://<Citrix-ADC-IP-address(NSIP)/nitro/v1/config/linkset_channel_binding

HTTP Method:PUT

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Content-Type:application/json

Request Payload:

```
{
```

```
"linkset_channel_binding":{

<b>"id":<String_value>,

</b><b>"ifnum":<String_value></b>

}}
```

Response:

HTTP Status Code on Success: 201 Created HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error

delete

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/linkset_channel_binding/id_value<String>

Query-parameters:

args

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/linkset_channel_binding/id_value<String>?**args=ifnum:<String_value>**

HTTP Method:DELETE

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue>

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error

get

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/linkset_channel_binding/id_value<String>

Query-parameters:

filter

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/linkset_channel_binding/id_value<String>?**filter=property-name1:property-value1,property-name2:property-value2**

Use this query-parameter to get the filtered set of linkset_channel_binding resources configured on Citrix ADC. Filtering can be done on any of the properties of the resource.

pagination

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/linkset_channel_binding/id_value<String>?**pagesize=#no;pageno=#no**

Use this query-parameter to get the linkset_channel_binding resources in chunks.

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "linkset_channel_binding": [ {

"ifnum":<String_value>,

"id":<String_value>

}]}
```

get (all)

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/linkset_channel_binding

Query-parameters:

bulkbindings

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/linkset_channel_binding?**bulkbindings=yes**

NITRO allows you to fetch bindings in bulk.

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "linkset_channel_binding": [ {

"ifnum":<String_value>,

"id":<String_value>

}]}
```

count

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/linkset_channel_binding/id_value<String>?**count=yes**

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{"linkset_channel_binding": [ { "__count": "#no"} ] }
```

linkset_interface_binding

Binding object showing the interface that can be bound to linkset.

Properties

(click to see Operations)

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
ifnum	<String>	Read-write	The interfaces to be bound to the linkset.
id	<String>	Read-write	ID of the linkset to which to bind the interfaces.
__count	<Double>	Read-write	count parameter

Operations

(click to see Properties)

ADD:| DELETE:| GET| GET (ALL)| COUNT

Some options that you can use for each operations:

- **Getting warnings in response:**NITRO allows you to get warnings in an operation by specifying the "warning" query parameter as "yes". For example, to get warnings while connecting to the Citrix ADC appliance, the URL is as follows:

  http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/login?warning=yes

  If any, the warnings are displayed in the response payload with the HTTP code "209 X-NITRO-WARNING".

- **Authenticated access for individual NITRO operations:**NITRO allows you to logon to the Citrix ADC appliance to perform individual operations. You can use this option instead of creating a NITRO session (using the login object) and then using that session to perform all operations,

  To do this, you must specify the username and password in the request header of the NITRO request as follows:

  X-NITRO-USER:<username>

  X-NITRO-PASS:<password>

  Note: In such cases, make sure that the request header DOES not include the following:

  Cookie:NITRO_AUTH_TOKEN=<tokenvalue>

`*Note:`*

Mandatory parameters are marked in redand placeholder content is marked in <green>.

add

URL:http://<Citrix-ADC-IP-address(NSIP)/nitro/v1/config/linkset_interface_binding

HTTP Method:PUT

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Content-Type:application/json

Request Payload:

```
{

"linkset_interface_binding":{

<b>"id":<String_value>,

</b><b>"ifnum":<String_value></b>

}}
```

Response:

HTTP Status Code on Success: 201 Created HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error

delete

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/linkset_interface_binding/id_value<String>

Query-parameters:

args

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/linkset_interface_binding/id_value<String>?**args=ifnum:<String_value>**

HTTP Method:DELETE

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue>

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error

get

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/linkset_interface_binding/id_value<String>

Query-parameters:

filter

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/linkset_interface_binding/id_value<String>?**filter=property-name1:property-value1,property-name2:property-value2**

Use this query-parameter to get the filtered set of linkset_interface_binding resources configured on Citrix ADC. Filtering can be done on any of the properties of the resource.

pagination

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/linkset_interface_binding/id_value<String>?**pagesize=#no;pageno=#no**

Use this query-parameter to get the linkset_interface_binding resources in chunks.

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "linkset_interface_binding": [ {

"ifnum":<String_value>,

"id":<String_value>
```

```
}]}
```

get (all)

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/linkset_interface_binding

Query-parameters:

bulkbindings

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/linkset_interface_binding?**bulkbindings=yes**

NITRO allows you to fetch bindings in bulk.

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "linkset_interface_binding": [ {

"ifnum":<String_value>,

"id":<String_value>

}]}
```

count

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/linkset_interface_binding/id_value<String>?**count=yes**

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{"linkset_interface_binding": [ { "__count": "#no"} ] }
```

mapbmr

Configuration for MAP-T Basic Mapping rule resource.

Properties

(click to see Operations)

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
name	<String>	Read-write	Name for the Basic Mapping Rule. Must begin with an ASCII alphanumeric or underscore (_) character, and must contain only ASCII alphanumeric,

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
			underscore, hash (#), period (.), space, colon (:), at (@), equals (=), and hyphen (-) characters. Cannot be changed after the MAP Basic Mapping Rule is created. The following requirement applies only to the Citrix ADC CLI: If the name includes one or more spaces, enclose the name in double or single quotation marks (for example, "add network MapBmr bmr1 -natprefix 2005::/64 -EAbitLength 16 -psidoffset 6 -portsharingratio 8"). The Basic Mapping Rule information allows a MAP BR to determine source IPv4 address from the IPv6 packet sent from MAP CE device. Also it allows to determine destination IPv6 address of MAP CE

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
			before sending packets to MAP CE. Minimum length = 1 Maximum length = 127
ruleipv6prefix	<String>	Read-write	IPv6 prefix of Customer Edge(CE) device.MAP-T CE will send ipv6 packets with this ipv6 prefix as source ipv6 address prefix.
psidoffset	<Double>	Read-write	Start bit position of Port Set Identifier(PSID) value in Embedded Address (EA) bits. Default value: 6 Minimum value = 1 Maximum value = 15
eabitlength	<Double>	Read-write	The Embedded Address (EA) bit field encodes the CE-specific IPv4 address and port information. The EA bit field, which is unique for a given Rule IPv6 prefix. Default value: 16

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
			Minimum value = 2 Maximum value = 47
psidlength	<Double>	Read-write	Length of Port Set IdentifierPort Set Identifier(PSID) in Embedded Address (EA) bits. Default value: 8 Minimum value = 1 Maximum value = 16
__count	<Double>	Read-only	count parameter

Operations

(click to see Properties)

ADD| DELETE| GET (ALL)| GET| COUNT

Some options that you can use for each operations:

- **Getting warnings in response:**NITRO allows you to get warnings in an operation by specifying the "warning" query parameter as "yes". For example, to get warnings while connecting to the Citrix ADC appliance, the URL is as follows:

  http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/login?warning=yes

  If any, the warnings are displayed in the response payload with the HTTP code "209 X-NITRO-WARNING".
- **Authenticated access for individual NITRO operations:**NITRO allows you to logon to the Citrix ADC appliance to perform individual operations. You can use this option instead of creating a NITRO session (using the login object) and then using that session to perform all operations,

To do this, you must specify the username and password in the request header of the NITRO request as follows:

X-NITRO-USER:<username>

X-NITRO-PASS:<password>

Note: In such cases, make sure that the request header DOES not include the following:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue>

`*Note:`*

Mandatory parameters are marked in redand placeholder content is marked in <green>.

add

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/mapbmr

HTTP Method:POST

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Content-Type:application/json

Request Payload:

```
{"mapbmr":{

<b>"name":<String_value>,

</b><b>"ruleipv6prefix":<String_value>,

</b>"psidoffset":<Double_value>,

"eabitlength":<Double_value>,

"psidlength":<Double_value>

}}
```

Response:

HTTP Status Code on Success: 201 Created HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error

delete

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/mapbmr/name_value<String>

HTTP Method:DELETE

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue>

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error

get (all)

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/mapbmr

Query-parameters:

attrs

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/mapbmr?**attrs=property-name1,property-name2**

Use this query parameter to specify the resource details that you want to retrieve.

filter

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/mapbmr?**filter=property-name1:property-val1,property-name2:property-val2**

Use this query-parameter to get the filtered set of mapbmr resources configured on Citrix ADC. Filtering can be done on any of the properties of the resource.

view

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/mapbmr?**view=summary**

Note: By default, the retrieved results are displayed in detail view (?view=detail).

pagination

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/mapbmr?**pagesize=#no;pageno=#no**

Use this query-parameter to get the mapbmr resources in chunks.

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "mapbmr": [ {

"name":<String_value>,

"ruleipv6prefix":<String_value>,

"eabitlength":<Double_value>,

"psidoffset":<Double_value>,

"psidlength":<Double_value>

}]}
```

get

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/mapbmr/name_value<String>

Query-parameters:

attrs

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/mapbmr/name_value<String>?**attrs=property-name1,property-name2**

Use this query parameter to specify the resource details that you want to retrieve.

view

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/mapbmr/name_value<String>?**view=summary**

Note: By default, the retrieved results are displayed in detail view (?view=detail).

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "mapbmr": [ {

"name":<String_value>,

"ruleipv6prefix":<String_value>,

"eabitlength":<Double_value>,

"psidoffset":<Double_value>,

"psidlength":<Double_value>
```

```
}]}
```

count

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/mapbmr?**count=yes**

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "mapbmr": [ { "__count": "#no"} ] }
```

mapbmr_binding

Binding object which returns the resources bound to mapbmr.

Properties

(click to see Operations)

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
name	<String>	Read-write	Name for the Basic Mapping Rule. Must begin with an ASCII alphanumeric or underscore (_)

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
			character, and must contain only ASCII alphanumeric, underscore, hash (#), period (.), space, colon (:), at (@), equals (=), and hyphen (-) characters. Cannot be changed after the MAP Basic Mapping Rule is created. The following requirement applies only to the Citrix ADC CLI: If the name includes one or more spaces, enclose the name in double or single quotation marks (for example, "add network MapBmr bmr1 -natprefix 2005::/64 -EAbitLength 16 -psidoffset 6 -portsharingratio 8"). The Basic Mapping Rule information allows a MAP BR to determine source IPv4 address from the IPv6 packet sent from MAP CE device. Also it allows to determine destination IPv6

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
			address of MAP CE before sending packets to MAP CE. Minimum length = 1 Maximum length = 127
mapbmr_bmrv4network_binding	<mapbmr_bmrv4network_binding[]>	Read-only	bmrv4network that can be bound to mapbmr.

Operations

(click to see Properties)

GET| GET (ALL)

Some options that you can use for each operations:

- **Getting warnings in response:**NITRO allows you to get warnings in an operation by specifying the "warning" query parameter as "yes". For example, to get warnings while connecting to the Citrix ADC appliance, the URL is as follows:

  http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/login?warning=yes

  If any, the warnings are displayed in the response payload with the HTTP code "209 X-NITRO-WARNING".
- **Authenticated access for individual NITRO operations:**NITRO allows you to logon to the Citrix ADC appliance to perform individual operations. You can use this option instead of creating a NITRO session (using the login object) and then using that session to perform all operations,

  To do this, you must specify the username and password in the request header of the NITRO request as follows:

  X-NITRO-USER:<username>

  X-NITRO-PASS:<password>

Note: In such cases, make sure that the request header DOES not include the following:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue>

`*Note:` *

Mandatory parameters are marked in redand placeholder content is marked in <green>.

get

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/mapbmr_binding/name_value<String>

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "mapbmr_binding": [ {

"name":<String_value>,

"mapbmr_bmrv4network_binding":<mapbmr_bmrv4network_binding[]_value>

}]}
```

get (all)

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/mapbmr_binding

Query-parameters:

bulkbindings

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/mapbmr_binding?**bulkbindings=yes**

NITRO allows you to fetch bindings in bulk.

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "mapbmr_binding": [ {

"name":<String_value>,

"mapbmr_bmrv4network_binding":<mapbmr_bmrv4network_binding[]_value>

}]}
```

mapbmr_bmrv4network_binding

Binding object showing the bmrv4network that can be bound to mapbmr.

Properties

(click to see Operations)

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
network	<String>	Read-write	IPv4 NAT address range of Customer Edge (CE).

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
			Minimum length = 1
name	<String>	Read-write	Name for the Basic Mapping Rule. Must begin with an ASCII alphanumeric or underscore (_) character, and must contain only ASCII alphanumeric, underscore, hash (#), period (.), space, colon (:), at (@), equals (=), and hyphen (-) characters. Cannot be changed after the MAP Basic Mapping Rule is created. The following requirement applies only to the Citrix ADC CLI: If the name includes one or more spaces, enclose the name in double or single quotation marks (for example, "add network MapBmr bmr1 -natprefix 2005::/64 -EAbitLength 16 -psidoffset 6 -portsharingratio 8"). The Basic Mapping Rule information allows a MAP BR to determine source IPv4 address from the IPv6 packet sent

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
			from MAP CE device. Also it allows to determine destination IPv6 address of MAP CE before sending packets to MAP CE. Minimum length = 1 Maximum length = 127
netmask	<String>	Read-write	Subnet mask for the IPv4 address specified in the Network parameter.
__count	<Double>	Read-write	count parameter

Operations

(click to see Properties)

ADD:| DELETE:| GET| GET (ALL)| COUNT

Some options that you can use for each operations:

- **Getting warnings in response:**NITRO allows you to get warnings in an operation by specifying the "warning" query parameter as "yes". For example, to get warnings while connecting to the Citrix ADC appliance, the URL is as follows:

  http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/login?warning=yes

  If any, the warnings are displayed in the response payload with the HTTP code "209 X-NITRO-WARNING".
- **Authenticated access for individual NITRO operations:**NITRO allows you to logon to the Citrix ADC appliance to perform individual operations. You can use this option instead of creating a NITRO session (using the login object) and then using that session to perform all operations,

To do this, you must specify the username and password in the request header of the NITRO request as follows:

X-NITRO-USER:<username>

X-NITRO-PASS:<password>

Note: In such cases, make sure that the request header DOES not include the following:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue>

`*Note:`*

Mandatory parameters are marked in redand placeholder content is marked in <green>.

add

URL:http://<Citrix-ADC-IP-address(NSIP)/nitro/v1/config/mapbmr_bmrv4network_binding

HTTP Method:PUT

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Content-Type:application/json

Request Payload:

```
{

"mapbmr_bmrv4network_binding":{

<b>"name":<String_value>,

</b><b>"network":<String_value>,

</b>"netmask":<String_value>

}}
```

Response:

HTTP Status Code on Success: 201 Created HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error

delete

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/mapbmr_bmrv4network_binding/name_value<String>

Query-parameters:

args

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/mapbmr_bmrv4network_binding/name_value<String>?**args=network:<String_value>,netmask:<String_value>**

HTTP Method:DELETE

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue>

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error

get

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/mapbmr_bmrv4network_binding/name_value<String>

Query-parameters:

filter

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/mapbmr_bmrv4network_binding/name_value<String>?**filter=property-name1:property-value1,property-name2:property-value2**

Use this query-parameter to get the filtered set of mapbmr_bmrv4network_binding resources configured on Citrix ADC. Filtering can be done on any of the properties of the resource.

pagination

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/mapbmr_bmrv4network_binding/name_value<String>?**pagesize=#no;pageno=#no**

Use this query-parameter to get the mapbmr_bmrv4network_binding resources in chunks.

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "mapbmr_bmrv4network_binding": [ {

"network":<String_value>,

"name":<String_value>,

"netmask":<String_value>

}]}
```

get (all)

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/mapbmr_bmrv4network_binding

Query-parameters:

bulkbindings

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/mapbmr_bmrv4network_binding?**bulkbindings=yes**

NITRO allows you to fetch bindings in bulk.

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "mapbmr_bmrv4network_binding": [ {

"network":<String_value>,

"name":<String_value>,

"netmask":<String_value>

}]}
```

count

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/mapbmr_bmrv4network_binding/name_value<String>?**count=yes**

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{"mapbmr_bmrv4network_binding": [ { "__count": "#no"} ] }
```

mapdmr

Configuration for MAP-T Default Mapping rule resource.

Properties

(click to see Operations)

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
name	<String>	Read-write	Name for the Default Mapping Rule. Must begin with an ASCII alphanumeric or underscore (_) character, and must contain only ASCII alphanumeric, underscore, hash (#), period (.), space, colon (:), at (@), equals (=), and hyphen (-) characters. Cannot be changed after the MAP Default Mapping Rule is created. The following requirement applies

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
			only to the Citrix ADC CLI: If the name includes one or more spaces, enclose the name in double or single quotation marks (for example, "add network MapDmr map1 -BRIpv6Prefix 2003::/96"). Default Mapping Rule (DMR) is defined in terms of the IPv6 prefix advertised by one or more BRs, which provide external connectivity. A typical MAP-T CE will install an IPv4 default route using this rule. A BR will use this rule when translating all outside IPv4 source addresses to the IPv6 MAP domain. Minimum length = 1 Maximum length = 127
bripv6prefix	<String>	Read-write	IPv6 prefix of Border Relay (Citrix ADC) device.MAP-T CE will send ipv6 packets to this ipv6

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
			prefix.The DMR IPv6 prefix length SHOULD be 64 bits long by default and in any case MUST NOT exceed 96 bits.
__count	<Double>	Read-only	count parameter

Operations

(click to see Properties)

ADD| DELETE| GET (ALL)| GET| COUNT

Some options that you can use for each operations:

- **Getting warnings in response:**NITRO allows you to get warnings in an operation by specifying the "warning" query parameter as "yes". For example, to get warnings while connecting to the Citrix ADC appliance, the URL is as follows:

  http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/login?warning=yes

  If any, the warnings are displayed in the response payload with the HTTP code "209 X-NITRO-WARNING".

- **Authenticated access for individual NITRO operations:**NITRO allows you to logon to the Citrix ADC appliance to perform individual operations. You can use this option instead of creating a NITRO session (using the login object) and then using that session to perform all operations,

  To do this, you must specify the username and password in the request header of the NITRO request as follows:

  X-NITRO-USER:<username>

  X-NITRO-PASS:<password>

  Note: In such cases, make sure that the request header DOES not include the following:

  Cookie:NITRO_AUTH_TOKEN=<tokenvalue>

`*Note:`*

Mandatory parameters are marked in redand placeholder content is marked in <green>.

add

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/mapdmr

HTTP Method:POST

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Content-Type:application/json

Request Payload:

```
{"mapdmr":{

<b>"name":<String_value>,

</b><b>"bripv6prefix":<String_value>

</b>}}
```

Response:

HTTP Status Code on Success: 201 Created HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error

delete

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/mapdmr/name_value<String>

HTTP Method:DELETE

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue>

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error

get (all)

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/mapdmr

Query-parameters:

attrs

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/mapdmr?**attrs=property-name1,property-name2**

Use this query parameter to specify the resource details that you want to retrieve.

filter

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/mapdmr?**filter=property-name1:property-val1,property-name2:property-val2**

Use this query-parameter to get the filtered set of mapdmr resources configured on Citrix ADC. Filtering can be done on any of the properties of the resource.

view

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/mapdmr?**view=summary**

Note: By default, the retrieved results are displayed in detail view (?view=detail).

pagination

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/mapdmr?**pagesize=#no;pageno=#no**

Use this query-parameter to get the mapdmr resources in chunks.

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "mapdmr": [ {

"name":<String_value>,

"bripv6prefix":<String_value>

}]}
```

get

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/mapdmr/name_value<String>

Query-parameters:

attrs

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/mapdmr/name_value<String>?**attrs=property-name1,property-name2**

Use this query parameter to specify the resource details that you want to retrieve.

view

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/mapdmr/name_value<String>?**view=summary**

Note: By default, the retrieved results are displayed in detail view (?view=detail).

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "mapdmr": [ {

"name":<String_value>,

"bripv6prefix":<String_value>

}]}
```

count

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/mapdmr?**count=yes**

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "mapdmr": [ { "__count": "#no"} ] }
```

mapdomain

Configuration for MAP-T Map Domain resource.

Properties

(click to see Operations)

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
name	<String>	Read-write	Name for the MAP Domain. Must begin with an ASCII alphanumeric or underscore (_) character, and must contain only ASCII alphanumeric, underscore, hash (#), period (.), space, colon (:), at (@), equals (=), and hyphen (-) characters. Cannot be changed after the MAP Domain is created . The following requirement applies only to the Citrix ADC CLI: If the name includes one or more spaces, enclose the name in double or single quotation marks (for example, "add network MapDomain map1"). Minimum length = 1

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
			Maximum length = 127
mapdmrname	<String>	Read-write	Default Mapping rule name.
__count	<Double>	Read-only	count parameter

Operations

(click to see Properties)

ADD| DELETE| GET (ALL)| GET| COUNT

Some options that you can use for each operations:

- **Getting warnings in response:**NITRO allows you to get warnings in an operation by specifying the "warning" query parameter as "yes". For example, to get warnings while connecting to the Citrix ADC appliance, the URL is as follows:

  http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/login?warning=yes

  If any, the warnings are displayed in the response payload with the HTTP code "209 X-NITRO-WARNING".

- **Authenticated access for individual NITRO operations:**NITRO allows you to logon to the Citrix ADC appliance to perform individual operations. You can use this option instead of creating a NITRO session (using the login object) and then using that session to perform all operations,

  To do this, you must specify the username and password in the request header of the NITRO request as follows:

  X-NITRO-USER:<username>

  X-NITRO-PASS:<password>

  Note: In such cases, make sure that the request header DOES not include the following:

  Cookie:NITRO_AUTH_TOKEN=<tokenvalue>

`*Note:` *

Mandatory parameters are marked in redand placeholder content is marked in <green>.

add

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/mapdomain

HTTP Method:POST

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Content-Type:application/json

Request Payload:

```
{"mapdomain":{

<b>"name":<String_value>,

</b><b>"mapdmrname":<String_value>

</b>}}
```

Response:

HTTP Status Code on Success: 201 Created HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error

delete

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/mapdomain/name_value<String>

HTTP Method:DELETE

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue>

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error

get (all)

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/mapdomain

Query-parameters:

attrs

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/mapdomain?**attrs=property-name1,property-name2**

Use this query parameter to specify the resource details that you want to retrieve.

filter

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/mapdomain?**filter=property-name1:property-val1,property-name2:property-val2**

Use this query-parameter to get the filtered set of mapdomain resources configured on Citrix ADC. Filtering can be done on any of the properties of the resource.

view

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/mapdomain?**view=summary**

Note: By default, the retrieved results are displayed in detail view (?view=detail).

pagination

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/mapdomain?**pagesize=#no;pageno=#no**

Use this query-parameter to get the mapdomain resources in chunks.

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "mapdomain": [ {

"name":<String_value>,

"mapdmrname":<String_value>

}]}
```

get

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/mapdomain/name_value<String>

Query-parameters:

attrs

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/mapdomain/name_value<String>?**attrs=property-name1,property-name2**

Use this query parameter to specify the resource details that you want to retrieve.

view

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/mapdomain/name_value<String>?**view=summary**

Note: By default, the retrieved results are displayed in detail view (?view=detail).

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "mapdomain": [ {

"name":<String_value>,

"mapdmrname":<String_value>

}]}
```

count

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/mapdomain?**count=yes**

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "mapdomain": [ { "__count": "#no"} ] }
```

mapdomain_binding

Binding object which returns the resources bound to mapdomain.

Properties

(click to see Operations)

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
name	<String>	Read-write	Name for the MAP Domain. Must begin with an ASCII alphanumeric or underscore (_) character, and must contain only ASCII alphanumeric, underscore, hash (#), period (.), space, colon (:), at (@), equals (=), and hyphen (-) characters. Cannot be changed after the MAP Domain is created . The following requirement applies only to the Citrix ADC CLI: If the name includes one or more spaces, enclose the name in double or single quotation marks (for example, "add network MapDomain map1"). Minimum length = 1

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
			Maximum length = 127
mapdomain_mapbmr_binding	<mapdomain_mapbmr_binding[]>	Read-only	mapbmr that can be bound to mapdomain.

Operations

(click to see Properties)

GET| GET (ALL)

Some options that you can use for each operations:

- **Getting warnings in response:**NITRO allows you to get warnings in an operation by specifying the "warning" query parameter as "yes". For example, to get warnings while connecting to the Citrix ADC appliance, the URL is as follows:

  http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/login?warning=yes

  If any, the warnings are displayed in the response payload with the HTTP code "209 X-NITRO-WARNING".

- **Authenticated access for individual NITRO operations:**NITRO allows you to logon to the Citrix ADC appliance to perform individual operations. You can use this option instead of creating a NITRO session (using the login object) and then using that session to perform all operations,

  To do this, you must specify the username and password in the request header of the NITRO request as follows:

  X-NITRO-USER:<username>

  X-NITRO-PASS:<password>

  Note: In such cases, make sure that the request header DOES not include the following:

  Cookie:NITRO_AUTH_TOKEN=<tokenvalue>

`*Note:` *

Mandatory parameters are marked in redand placeholder content is marked in <green>.

get

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/mapdomain_binding/name_value<String>

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "mapdomain_binding": [ {

"name":<String_value>,

"mapdomain_mapbmr_binding":<mapdomain_mapbmr_binding[]_value>

}]}
```

get (all)

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/mapdomain_binding

Query-parameters:

bulkbindings

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/mapdomain_binding?**bulkbindings=yes**

NITRO allows you to fetch bindings in bulk.

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "mapdomain_binding": [ {

"name":<String_value>,

"mapdomain_mapbmr_binding":<mapdomain_mapbmr_binding[]_value>

}]}
```

mapdomain_mapbmr_binding

Binding object showing the mapbmr that can be bound to mapdomain.

Properties

(click to see Operations)

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
mapbmrname	<String>	Read-write	Basic Mapping rule name.
name	<String>	Read-write	Name for the MAP Domain. Must begin with an ASCII alphanumeric or underscore (_) character, and must

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
			contain only ASCII alphanumeric, underscore, hash (#), period (.), space, colon (:), at (@), equals (=), and hyphen (-) characters. Cannot be changed after the MAP Domain is created . The following requirement applies only to the Citrix ADC CLI: If the name includes one or more spaces, enclose the name in double or single quotation marks (for example, "add network MapDomain map1"). Minimum length = 1 Maximum length = 127
__count	<Double>	Read-write	count parameter

Operations

(click to see Properties)

ADD:| DELETE:| GET| GET (ALL)| COUNT

Some options that you can use for each operations:

- **Getting warnings in response:**NITRO allows you to get warnings in an operation by specifying the "warning" query parameter as "yes". For example, to get warnings while connecting to the Citrix ADC appliance, the URL is as follows:

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/login?warning=yes

If any, the warnings are displayed in the response payload with the HTTP code "209 X-NITRO-WARNING".

- **Authenticated access for individual NITRO operations:**NITRO allows you to logon to the Citrix ADC appliance to perform individual operations. You can use this option instead of creating a NITRO session (using the login object) and then using that session to perform all operations,

  To do this, you must specify the username and password in the request header of the NITRO request as follows:

  X-NITRO-USER:<username>

  X-NITRO-PASS:<password>

  Note: In such cases, make sure that the request header DOES not include the following:

  Cookie:NITRO_AUTH_TOKEN=<tokenvalue>

`*Note:`*

Mandatory parameters are marked in redand placeholder content is marked in <green>.

add

URL:http://<Citrix-ADC-IP-address(NSIP)/nitro/v1/config/mapdomain_mapbmr_binding

HTTP Method:PUT

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Content-Type:application/json

Request Payload:

```
{

"mapdomain_mapbmr_binding":{

<b>"name":<String_value>,

</b><b>"mapbmrname":<String_value></b>
```

```
}}
```

Response:

HTTP Status Code on Success: 201 Created HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error

delete

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/mapdomain_mapbmr_binding/name_value<String>

Query-parameters:

args

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/mapdomain_mapbmr_binding/name_value<String>?**args=mapbmrname:<String_value>**

HTTP Method:DELETE

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue>

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error

get

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/mapdomain_mapbmr_binding/name_value<String>

Query-parameters:

filter

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/mapdomain_mapbmr_binding/name_value<String>?**filter=property-name1:property-value1,property-name2:property-value2**

Use this query-parameter to get the filtered set of mapdomain_mapbmr_binding resources configured on Citrix ADC. Filtering can be done on any of the properties of the resource.

pagination

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/mapdomain_mapbmr_binding/name_value<String>?**pagesize=#no;pageno=#no**

Use this query-parameter to get the mapdomain_mapbmr_binding resources in chunks.

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "mapdomain_mapbmr_binding": [ {

"mapbmrname":<String_value>,

"name":<String_value>

}]}
```

get (all)

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/mapdomain_mapbmr_binding

Query-parameters:

bulkbindings

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/mapdomain_mapbmr_binding?**bulkbindings=yes**

NITRO allows you to fetch bindings in bulk.

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "mapdomain_mapbmr_binding": [ {

"mapbmrname":<String_value>,

"name":<String_value>

}]}
```

count

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/mapdomain_mapbmr_binding/name_value<String>?**count=yes**

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{"mapdomain_mapbmr_binding": [ { "__count": "#no"} ] }
```

nat64

Configuration for nat64 config resource.

Properties

(click to see Operations)

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
name	<String>	Read-write	Name for the NAT64 rule. Must begin with a letter, number, or the underscore character (_), and can consist of letters, numbers, and the hyphen (-), period (.) pound (#), space ( ), at sign (@), equals (=), colon (:), and underscore characters. Cannot be changed after the rule is created. Choose a name that helps identify the NAT64 rule. Minimum length = 1

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
acl6name	<String>	Read-write	Name of any configured ACL6 whose action is ALLOW. IPv6 Packets matching the condition of this ACL6 rule and destination IP address of these packets matching the NAT64 IPv6 prefix are considered for NAT64 translation. Minimum length = 1
netprofile	<String>	Read-write	Name of the configured netprofile. The Citrix ADC selects one of the IP address in the netprofile as the source IP address of the translated IPv4 packet to be sent to the IPv4 server. Minimum length = 1 Maximum length = 127
__count	<Double>	Read-only	count parameter

Operations

(click to see Properties)

ADD| UPDATE| UNSET| DELETE| GET (ALL)| GET| COUNT

Some options that you can use for each operations:

- **Getting warnings in response:**NITRO allows you to get warnings in an operation by specifying the "warning" query parameter as "yes". For example, to get warnings while connecting to the Citrix ADC appliance, the URL is as follows:

  http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/login?warning=yes

  If any, the warnings are displayed in the response payload with the HTTP code "209 X-NITRO-WARNING".

- **Authenticated access for individual NITRO operations:**NITRO allows you to logon to the Citrix ADC appliance to perform individual operations. You can use this option instead of creating a NITRO session (using the login object) and then using that session to perform all operations,

  To do this, you must specify the username and password in the request header of the NITRO request as follows:

  X-NITRO-USER:<username>

  X-NITRO-PASS:<password>

  Note: In such cases, make sure that the request header DOES not include the following:

  Cookie:NITRO_AUTH_TOKEN=<tokenvalue>

`*Note:`*

Mandatory parameters are marked in redand placeholder content is marked in <green>.

add

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/nat64

HTTP Method:POST

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Content-Type:application/json

Request Payload:

```
{"nat64":{
```

```
<b>"name":<String_value>,

</b><b>"acl6name":<String_value>,

</b>"netprofile":<String_value>

}}
```

Response:

HTTP Status Code on Success: 201 Created HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error

update

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/nat64

HTTP Method:PUT

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Content-Type:application/json

Request Payload:

```
{"nat64":{

<b>"name":<String_value>,

</b>"acl6name":<String_value>,

"netprofile":<String_value>

}}
```

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error

unset

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/nat64?**action=unset**

HTTP Method:POST

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Content-Type:application/json

Request Payload:

```
{"nat64":{

<b>"name":<String_value>,

</b>"netprofile":true

}}
```

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error

delete

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/nat64/name_value<String>

HTTP Method:DELETE

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue>

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error

get (all)

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/nat64

Query-parameters:

attrs

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/nat64?**attrs=property-name1,property-name2**

Use this query parameter to specify the resource details that you want to retrieve.

filter

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/nat64?**filter=property-name1:property-val1,property-name2:property-val2**

Use this query-parameter to get the filtered set of nat64 resources configured on Citrix ADC. Filtering can be done on any of the properties of the resource.

view

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/nat64?**view=summary**

Note: By default, the retrieved results are displayed in detail view (?view=detail).

pagination

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/nat64?**pagesize=#no;pageno=#no**

Use this query-parameter to get the nat64 resources in chunks.

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "nat64": [ {

"name":<String_value>,

"acl6name":<String_value>,

"netprofile":<String_value>

}]}
```

get

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/nat64/name_value<String>

Query-parameters:

attrs

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/nat64/name_value<String>?**attrs=property-name1,property-name2**

Use this query parameter to specify the resource details that you want to retrieve.

view

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/nat64/name_value<String>?**view=summary**

Note: By default, the retrieved results are displayed in detail view (?view=detail).

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "nat64": [ {

"name":<String_value>,

"acl6name":<String_value>,

"netprofile":<String_value>

}]}
```

count

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/nat64?**count=yes**

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "nat64": [ { "__count": "#no"} ] }
```

nat64param

Configuration for NAT64 parameter resource.

Properties

(click to see Operations)

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
td	<Double>	Read-write	Integer value that uniquely identifies the traffic domain in which you want to configure the entity. If you do not specify an ID, the entity becomes part of the default traffic domain, which has an ID of 0. Default value: 0 Minimum value = 0 Maximum value = 4094
nat64ignoretos	<String>	Read-write	Ignore TOS. Default value: NO Possible values = YES, NO
nat64zerochecksum	<String>	Read-write	Calculate checksum for UDP packets with zero checksum.

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
			Default value: ENABLED Possible values = ENABLED, DISABLED
nat64v6mtu	<Double>	Read-write	MTU setting for the IPv6 side. If the incoming IPv4 packet greater than this, either fragment or send icmp need fragmentation error. Default value: 1280 Minimum value = 1280 Maximum value = 9216
nat64fragheader	<String>	Read-write	When disabled, translator will not insert IPv6 fragmentation header for non fragmented IPv4 packets. Default value: ENABLED Possible values = ENABLED, DISABLED
__count	<Double>	Read-only	count parameter

Operations

(click to see Properties)

UPDATE| UNSET| GET (ALL)| GET| COUNT

Some options that you can use for each operations:

- **Getting warnings in response:**NITRO allows you to get warnings in an operation by specifying the "warning" query parameter as "yes". For example, to get warnings while connecting to the Citrix ADC appliance, the URL is as follows:

  http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/login?warning=yes

  If any, the warnings are displayed in the response payload with the HTTP code "209 X-NITRO-WARNING".

- **Authenticated access for individual NITRO operations:**NITRO allows you to logon to the Citrix ADC appliance to perform individual operations. You can use this option instead of creating a NITRO session (using the login object) and then using that session to perform all operations,

  To do this, you must specify the username and password in the request header of the NITRO request as follows:

  X-NITRO-USER:<username>

  X-NITRO-PASS:<password>

  Note: In such cases, make sure that the request header DOES not include the following:

  Cookie:NITRO_AUTH_TOKEN=<tokenvalue>

`*Note:`*

Mandatory parameters are marked in redand placeholder content is marked in <green>.

update

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/nat64param

HTTP Method:PUT

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Content-Type:application/json

Request Payload:

```
{"nat64param":{

"td":<Double_value>,

"nat64ignoretos":<String_value>,

"nat64zerochecksum":<String_value>,

"nat64v6mtu":<Double_value>,

"nat64fragheader":<String_value>

}}
```

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error

unset

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/nat64param?**action=unset**

HTTP Method:POST

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Content-Type:application/json

Request Payload:

```
{"nat64param":{

"td":<Double_value>,

"nat64ignoretos":true,
```

```
"nat64zerochecksum":true,

"nat64v6mtu":true,

"nat64fragheader":true

}}
```

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error

get (all)

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/nat64param

Query-parameters:

attrs

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/nat64param?**attrs=property-name1,property-name2**

Use this query parameter to specify the resource details that you want to retrieve.

filter

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/nat64param?**filter=property-name1:property-val1,property-name2:property-val2**

Use this query-parameter to get the filtered set of nat64param resources configured on Citrix ADC. Filtering can be done on any of the properties of the resource.

view

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/nat64param?**view=summary**

Note: By default, the retrieved results are displayed in detail view (?view=detail).

pagination

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/nat64param?**pagesize=#no;pageno=#no**

Use this query-parameter to get the nat64param resources in chunks.

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "nat64param": [ {

"td":<Double_value>,

"nat64ignoretos":<String_value>,

"nat64zerochecksum":<String_value>,

"nat64v6mtu":<Double_value>,

"nat64fragheader":<String_value>

}]}
```

get

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/nat64param/td_value<Double>

Query-parameters:

attrs

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/nat64param/td_value<Double>?**attrs=property-name1,property-name2**

Use this query parameter to specify the resource details that you want to retrieve.

view

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/nat64param/td_value<Double>?**view=summary**

Note: By default, the retrieved results are displayed in detail view (?view=detail).

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "nat64param": [ {

"td":<Double_value>,

"nat64ignoretos":<String_value>,

"nat64zerochecksum":<String_value>,

"nat64v6mtu":<Double_value>,

"nat64fragheader":<String_value>

}]}
```

count

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/nat64param?**count=yes**

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "nat64param": [ { "__count": "#no"} ] }
```

nd6

Configuration for nd6 resource.

Properties

(click to see Operations)

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
neighbor	<String>	Read-write	Link-local IPv6 address of the adjacent network device to add to the ND6 table.

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
mac	<String>	Read-write	MAC address of the adjacent network device.
ifnum	<String>	Read-write	Interface through which the adjacent network device is available, specified in slot/port notation (for example, 1/3). Use spaces to separate multiple entries.
vlan	<Integer>	Read-write	Integer value that uniquely identifies the VLAN on which the adjacent network device exists. Minimum value = 1 Maximum value = 4094
vxlan	<Double>	Read-write	ID of the VXLAN on which the IPv6 address of this ND6 entry is reachable. Minimum value = 1 Maximum value = 16777215
vtep	<String>	Read-write	IP address of the VXLAN tunnel endpoint (VTEP) through which the IPv6 address of this ND6 entry is reachable.

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
			Minimum length = 1
td	<Double>	Read-write	Integer value that uniquely identifies the traffic domain in which you want to configure the entity. If you do not specify an ID, the entity becomes part of the default traffic domain, which has an ID of 0. Minimum value = 0 Maximum value = 4094
nodeid	<Double>	Read-write	Unique number that identifies the cluster node. Minimum value = 0 Maximum value = 31
state	<String>	Read-only	ND6 state. Default value: REACHABLE Possible values = INCOMPLETE, REACHABLE, STALE, DELAY, PROBE
timeout	<Double>	Read-only	Time elapsed.
flags	<Double>	Read-only	flag for static/ permanent entry.

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
controlplane	<Boolean>	Read-only	This nd6 entry is populated by a control plane protocol.
channel	<Double>	Read-only	The tunnel that is bound to a netbridge.
__count	<Double>	Read-only	count parameter

Operations

(click to see Properties)

ADD| CLEAR| DELETE| GET (ALL)| COUNT

Some options that you can use for each operations:

- **Getting warnings in response:**NITRO allows you to get warnings in an operation by specifying the "warning" query parameter as "yes". For example, to get warnings while connecting to the Citrix ADC appliance, the URL is as follows:

  http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/login?warning=yes

  If any, the warnings are displayed in the response payload with the HTTP code "209 X-NITRO-WARNING".

- **Authenticated access for individual NITRO operations:**NITRO allows you to logon to the Citrix ADC appliance to perform individual operations. You can use this option instead of creating a NITRO session (using the login object) and then using that session to perform all operations,

  To do this, you must specify the username and password in the request header of the NITRO request as follows:

  X-NITRO-USER:<username>

  X-NITRO-PASS:<password>

  Note: In such cases, make sure that the request header DOES not include the following:

  Cookie:NITRO_AUTH_TOKEN=<tokenvalue>

`*Note:` *

Mandatory parameters are marked in redand placeholder content is marked in <green>.

add

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/nd6

HTTP Method:POST

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Content-Type:application/json

Request Payload:

```
{"nd6":{

<b>"neighbor":<String_value>,

</b><b>"mac":<String_value>,

</b>"ifnum":<String_value>,

"vlan":<Integer_value>,

"vxlan":<Double_value>,

"vtep":<String_value>,

"td":<Double_value>

}}
```

Response:

HTTP Status Code on Success: 201 Created HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error

clear

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/nd6?**action=clear**

HTTP Method:POST

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Content-Type:application/json

Request Payload:

```
{"nd6":{

}}
```

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error

delete

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/nd6/neighbor_value<String>

HTTP Method:DELETE

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue>

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error

get (all)

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/nd6

Query-parameters:

args

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/nd6?**args=neighbor:<String_value>,td:<Double_value>,nodeid:<Double_value>**

Use this query-parameter to get nd6 resources based on additional properties.

attrs

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/nd6?**attrs=property-name1,property-name2**

Use this query parameter to specify the resource details that you want to retrieve.

filter

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/nd6?**filter=property-name1:property-val1,property-name2:property-val2**

Use this query-parameter to get the filtered set of nd6 resources configured on Citrix ADC. Filtering can be done on any of the properties of the resource.

view

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/nd6?**view=summary**

Note: By default, the retrieved results are displayed in detail view (?view=detail).

pagination

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/nd6?**pagesize=#no;pageno=#no**

Use this query-parameter to get the nd6 resources in chunks.

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "nd6": [ {

neighbor:<String_value>,td:<Double_value>,nodeid:<Double_value>"mac":<Stri
ng_value>,

"state":<String_value>,

"timeout":<Double_value>,

"ifnum":<String_value>,

"vlan":<Integer_value>,

"vxlan":<Double_value>,

"vtep":<String_value>,

"flags":<Double_value>,

"controlplane":<Boolean_value>,

"channel":<Double_value>

}]}
```

count

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/nd6?**count=yes**

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "nd6": [ { "__count": "#no"} ] }
```

nd6ravariables

Configuration for nd6 Router Advertisment configuration variables resource.

Properties

(click to see Operations)

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
vlan	<Double>	Read-write	The VLAN number. Minimum value = 1 Maximum value = 4094
ceaserouteradv	<String>	Read-write	Cease router advertisements on this vlan. Default value: NO Possible values = YES, NO
sendrouteradv	<String>	Read-write	whether the router sends periodic RAs and responds to Router Solicitations. Default value: NO

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
			Possible values = YES, NO
srclinklayeraddroption	<String>	Read-write	Include source link layer address option in RA messages. Default value: YES Possible values = YES, NO
onlyunicastrtadvresponse	<String>	Read-write	Send only Unicast Router Advertisements in respond to Router Solicitations. Default value: NO Possible values = YES, NO
managedaddrconfig	<String>	Read-write	Value to be placed in the Managed address configuration flag field. Default value: NO Possible values = YES, NO
otheraddrconfig	<String>	Read-write	Value to be placed in the Other configuration flag field. Default value: NO Possible values = YES, NO

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
currhoplimit	<Double>	Read-write	Current Hop limit. Default value: 64 Minimum value = 0 Maximum value = 255
maxrtadvinterval	<Double>	Read-write	Maximum time allowed between unsolicited multicast RAs, in seconds. Default value: 600 Minimum value = 4 Maximum value = 1800
minrtadvinterval	<Double>	Read-write	Minimum time interval between RA messages, in seconds. Default value: 198 Minimum value = 3 Maximum value = 1350
linkmtu	<Double>	Read-write	The Link MTU. Default value: 0 Minimum value = 0 Maximum value = 1500
reachabletime	<Double>	Read-write	Reachable time, in milliseconds.

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
			Default value: 0 Minimum value = 0 Maximum value = 3600000
retranstime	<Double>	Read-write	Retransmission time, in milliseconds. Default value: 0
defaultlifetime	<Integer>	Read-write	Default life time, in seconds. Default value: 1800 Minimum value = 0 Maximum value = 9000
lastrtadvtime	<Double>	Read-only	Last RA sent timestamp.
nextrtadvdelay	<Double>	Read-only	Next RA delay.
__count	<Double>	Read-only	count parameter

Operations

(click to see Properties)

UPDATE| UNSET| GET (ALL)| GET| COUNT

Some options that you can use for each operations:

- **Getting warnings in response:**NITRO allows you to get warnings in an operation by specifying the "warning" query parameter as "yes". For example, to get warnings while connecting to the Citrix ADC appliance, the URL is as follows:

  http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/login?warning=yes

If any, the warnings are displayed in the response payload with the HTTP code "209 X-NITRO-WARNING".

- **Authenticated access for individual NITRO operations:**NITRO allows you to logon to the Citrix ADC appliance to perform individual operations. You can use this option instead of creating a NITRO session (using the login object) and then using that session to perform all operations,

  To do this, you must specify the username and password in the request header of the NITRO request as follows:

  X-NITRO-USER:<username>

  X-NITRO-PASS:<password>

  Note: In such cases, make sure that the request header DOES not include the following:

  Cookie:NITRO_AUTH_TOKEN=<tokenvalue>

`*Note:`*

Mandatory parameters are marked in redand placeholder content is marked in <green>.

update

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/nd6ravariables

HTTP Method:PUT

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Content-Type:application/json

Request Payload:

```
{"nd6ravariables":{

<b>"vlan":<Double_value>,

</b>"ceaserouteradv":<String_value>,

"sendrouteradv":<String_value>,
```

```
"srclinklayeraddroption":<String_value>,

"onlyunicastrtadvresponse":<String_value>,

"managedaddrconfig":<String_value>,

"otheraddrconfig":<String_value>,

"currhoplimit":<Double_value>,

"maxrtadvinterval":<Double_value>,

"minrtadvinterval":<Double_value>,

"linkmtu":<Double_value>,

"reachabletime":<Double_value>,

"retranstime":<Double_value>,

"defaultlifetime":<Integer_value>

}}
```

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error

unset

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/nd6ravariables?**action=unset**

HTTP Method:POST

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Content-Type:application/json

Request Payload:

```
{"nd6ravariables":{
<b>"vlan":<Double_value>,
</b>"ceaserouteradv":true,
"sendrouteradv":true,
"srclinklayeraddroption":true,
"onlyunicastrtadvresponse":true,
"managedaddrconfig":true,
"otheraddrconfig":true,
"currhoplimit":true,
"maxrtadvinterval":true,
"minrtadvinterval":true,
"linkmtu":true,
"reachabletime":true,
"retranstime":true,
"defaultlifetime":true
}}
```

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error

get (all)

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/nd6ravariables

Query-parameters:

attrs

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/nd6ravariables?**attrs=property-name1,property-name2**

Use this query parameter to specify the resource details that you want to retrieve.

filter

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/nd6ravariables?**filter=property-name1:property-val1,property-name2:property-val2**

Use this query-parameter to get the filtered set of nd6ravariables resources configured on Citrix ADC. Filtering can be done on any of the properties of the resource.

view

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/nd6ravariables?**view=summary**

Note: By default, the retrieved results are displayed in detail view (?view=detail).

pagination

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/nd6ravariables?**pagesize=#no;pageno=#no**

Use this query-parameter to get the nd6ravariables resources in chunks.

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "nd6ravariables": [ {
"vlan":<Double_value>,
"ceaserouteradv":<String_value>,
"sendrouteradv":<String_value>,
"srclinklayeraddroption":<String_value>,
"onlyunicastrtadvresponse":<String_value>,
"managedaddrconfig":<String_value>,
"otheraddrconfig":<String_value>,
"currhoplimit":<Double_value>,
"maxrtadvinterval":<Double_value>,
"minrtadvinterval":<Double_value>,
"linkmtu":<Double_value>,
"reachabletime":<Double_value>,
"retranstime":<Double_value>,
"defaultlifetime":<Integer_value>,
"lastrtadvtime":<Double_value>,
"nextrtadvdelay":<Double_value>
}]}
```

get

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/nd6ravariables/vlan_value<Double>

Query-parameters:

attrs

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/nd6ravariables/vlan_value<Double>?**attrs=property-name1,property-name2**

Use this query parameter to specify the resource details that you want to retrieve.

view

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/nd6ravariables/vlan_value<Double>?**view=summary**

Note: By default, the retrieved results are displayed in detail view (?view=detail).

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "nd6ravariables": [ {

"vlan":<Double_value>,

"ceaserouteradv":<String_value>,

"sendrouteradv":<String_value>,
```

```
"srclinklayeraddroption":<String_value>,

"onlyunicastrtadvresponse":<String_value>,

"managedaddrconfig":<String_value>,

"otheraddrconfig":<String_value>,

"currhoplimit":<Double_value>,

"maxrtadvinterval":<Double_value>,

"minrtadvinterval":<Double_value>,

"linkmtu":<Double_value>,

"reachabletime":<Double_value>,

"retranstime":<Double_value>,

"defaultlifetime":<Integer_value>,

"lastrtadvtime":<Double_value>,

"nextrtadvdelay":<Double_value>

}]}
```

count

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/nd6ravariables?**count=yes**

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "nd6ravariables": [ { "__count": "#no"} ] }
```

nd6ravariables_binding

Binding object which returns the resources bound to nd6ravariables.

Properties

(click to see Operations)

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
vlan	<Double>	Read-write	The VLAN number. Minimum value = 1 Maximum value = 4094
nd6ravariables_onlinkipv6prefix_binding	<nd6ravariables_onlinkipv6prefix_binding[]>	Read-only	onlinkipv6prefix that can be bound to nd6ravariables.

Operations

(click to see Properties)

GET| GET (ALL)

Some options that you can use for each operations:

- **Getting warnings in response:**NITRO allows you to get warnings in an operation by specifying the "warning" query parameter as "yes". For example, to get warnings while connecting to the Citrix ADC appliance, the URL is as follows:

  http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/login?warning=yes

  If any, the warnings are displayed in the response payload with the HTTP code "209 X-NITRO-WARNING".

- **Authenticated access for individual NITRO operations:**NITRO allows you to logon to the Citrix ADC appliance to perform individual operations. You can use this option instead of creating a NITRO session (using the login object) and then using that session to perform all operations,

  To do this, you must specify the username and password in the request header of the NITRO request as follows:

  X-NITRO-USER:<username>

  X-NITRO-PASS:<password>

  Note: In such cases, make sure that the request header DOES not include the following:

  Cookie:NITRO_AUTH_TOKEN=<tokenvalue>

`*Note:`*

Mandatory parameters are marked in redand placeholder content is marked in <green>.

get

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/nd6ravariables_binding/vlan_value<Double>

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "nd6ravariables_binding": [ {

"vlan":<Double_value>,

"nd6ravariables_onlinkipv6prefix_binding":<nd6ravariables_onlinkipv6prefix
_binding[]_value>

}]}
```

get (all)

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/nd6ravariables_binding

Query-parameters:

bulkbindings

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/nd6ravariables_binding?**bulkbindings=yes**

NITRO allows you to fetch bindings in bulk.

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "nd6ravariables_binding": [ {

"vlan":<Double_value>,

"nd6ravariables_onlinkipv6prefix_binding":<nd6ravariables_onlinkipv6prefix
_binding[]_value>

}]}
```

nd6ravariables_onlinkipv6prefix_binding

Binding object showing the onlinkipv6prefix that can be bound to nd6ravariables.

Properties

(click to see Operations)

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
ipv6prefix	<String>	Read-write	Onlink prefixes for RA messages.
vlan	<Double>	Read-write	The VLAN number. Minimum value = 1 Maximum value = 4094
__count	<Double>	Read-write	count parameter

Operations

(click to see Properties)

ADD:| DELETE:| GET| GET (ALL)| COUNT

Some options that you can use for each operations:

- **Getting warnings in response:**NITRO allows you to get warnings in an operation by specifying the "warning" query parameter as "yes". For example, to get warnings while connecting to the Citrix ADC appliance, the URL is as follows:

  http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/login?warning=yes

  If any, the warnings are displayed in the response payload with the HTTP code "209 X-NITRO-WARNING".

- **Authenticated access for individual NITRO operations:**NITRO allows you to logon to the Citrix ADC appliance to perform individual operations. You can use this option instead of creating a NITRO session (using the login object) and then using that session to perform all operations,

  To do this, you must specify the username and password in the request header of the NITRO request as follows:

  X-NITRO-USER:<username>

  X-NITRO-PASS:<password>

  Note: In such cases, make sure that the request header DOES not include the following:

  Cookie:NITRO_AUTH_TOKEN=<tokenvalue>

`*Note:`*

Mandatory parameters are marked in redand placeholder content is marked in <green>.

add

URL:http://<Citrix-ADC-IP-address(NSIP)/nitro/v1/config/nd6ravariables_onlinkipv6prefix_binding

HTTP Method:PUT

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Content-Type:application/json

Request Payload:

```
{
```

```
"nd6ravariables_onlinkipv6prefix_binding":{

<b>"vlan":<Double_value>,

</b><b>"ipv6prefix":<String_value></b>

}}
```

Response:

HTTP Status Code on Success: 201 Created HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error

delete

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/nd6ravariables_onlinkipv6prefix_binding/vlan_value<Double>

Query-parameters:

args

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/nd6ravariables_onlinkipv6prefix_binding/vlan_value<Double>?**args=ipv6prefix:<String_value>**

HTTP Method:DELETE

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue>

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error

get

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/nd6ravariables_onlinkipv6prefix_binding/vlan_value<Double>

Query-parameters:

filter

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/nd6ravariables_onlinkipv6prefix_binding/vlan_value<Double>?**filter=property-name1:property-value1,property-name2:property-value2**

Use this query-parameter to get the filtered set of nd6ravariables_onlinkipv6prefix_binding resources configured on Citrix ADC. Filtering can be done on any of the properties of the resource.

pagination

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/nd6ravariables_onlinkipv6prefix_binding/vlan_value<Double>?**pagesize=#no;pageno=#no**

Use this query-parameter to get the nd6ravariables_onlinkipv6prefix_binding resources in chunks.

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "nd6ravariables_onlinkipv6prefix_binding": [ {

"ipv6prefix":<String_value>,

"vlan":<Double_value>

}]}
```

get (all)

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/nd6ravariables_onlinkipv6prefix_binding

Query-parameters:

bulkbindings

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/nd6ravariables_onlinkipv6prefix_binding?**bulkbindings=yes**

NITRO allows you to fetch bindings in bulk.

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "nd6ravariables_onlinkipv6prefix_binding": [ {

"ipv6prefix":<String_value>,

"vlan":<Double_value>

}]}
```

count

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/nd6ravariables_onlinkipv6prefix_binding/vlan_value<Double>?**count=yes**

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{"nd6ravariables_onlinkipv6prefix_binding": [ { "__count": "#no"} ] }
```

netbridge

Configuration for network bridge resource.

Properties

(click to see Operations)

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
name	<String>	Read-write	The name of the network bridge.
vxlanvlanmap	<String>	Read-write	The vlan to vxlan mapping to be applied to this netbridge.
__count	<Double>	Read-only	count parameter

Operations

(click to see Properties)

ADD| UPDATE| UNSET| DELETE| GET (ALL)| GET| COUNT

Some options that you can use for each operations:

- **Getting warnings in response:**NITRO allows you to get warnings in an operation by specifying the "warning" query parameter as "yes". For example, to get warnings while connecting to the Citrix ADC appliance, the URL is as follows:

  http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/login?warning=yes

  If any, the warnings are displayed in the response payload with the HTTP code "209 X-NITRO-WARNING".

- **Authenticated access for individual NITRO operations:**NITRO allows you to logon to the Citrix ADC appliance to perform individual operations. You can use this option instead of creating a NITRO session (using the login object) and then using that session to perform all operations,

  To do this, you must specify the username and password in the request header of the NITRO request as follows:

  X-NITRO-USER:<username>

  X-NITRO-PASS:<password>

  Note: In such cases, make sure that the request header DOES not include the following:

  Cookie:NITRO_AUTH_TOKEN=<tokenvalue>

`*Note:`*

Mandatory parameters are marked in redand placeholder content is marked in <green>.

add

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/netbridge

HTTP Method:POST

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Content-Type:application/json

Request Payload:

```
{"netbridge":{

<b>"name":<String_value>,

</b>"vxlanvlanmap":<String_value>

}}
```

Response:

HTTP Status Code on Success: 201 Created HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error

update

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/netbridge

HTTP Method:PUT

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Content-Type:application/json

Request Payload:

```
{"netbridge":{

<b>"name":<String_value>,

</b>"vxlanvlanmap":<String_value>

}}
```

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error

unset

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/netbridge?**action=unset**

HTTP Method:POST

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Content-Type:application/json

Request Payload:

```
{"netbridge":{

<b>"name":<String_value>,

</b>"vxlanvlanmap":true

}}
```

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error

delete

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/netbridge/name_value<String>

HTTP Method:DELETE

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue>

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error

get (all)

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/netbridge

Query-parameters:

attrs

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/netbridge?**attrs=property-name1,property-name2**

Use this query parameter to specify the resource details that you want to retrieve.

filter

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/netbridge?**filter=property-name1:property-val1,property-name2:property-val2**

Use this query-parameter to get the filtered set of netbridge resources configured on Citrix ADC. Filtering can be done on any of the properties of the resource.

view

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/netbridge?**view=summary**

Note: By default, the retrieved results are displayed in detail view (?view=detail).

pagination

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/netbridge?**pagesize=#no;pageno=#no**

Use this query-parameter to get the netbridge resources in chunks.

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "netbridge": [ {

"name":<String_value>,

"vxlanvlanmap":<String_value>

}]}
```

get

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/netbridge/name_value<String>

Query-parameters:

attrs

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/netbridge/name_value<String>?**attrs=property-name1,property-name2**

Use this query parameter to specify the resource details that you want to retrieve.

view

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/netbridge/name_value<String>?**view=summary**

Note: By default, the retrieved results are displayed in detail view (?view=detail).

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "netbridge": [ {

"name":<String_value>,

"vxlanvlanmap":<String_value>

}]}
```

count

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/netbridge?**count=yes**

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "netbridge": [ { "__count": "#no"} ] }
```

netbridge_binding

Binding object which returns the resources bound to netbridge.

Properties

(click to see Operations)

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
name	<String>	Read-write	The name of the network bridge.
netbridge_iptunnel_binding	<netbridge_iptunnel_binding[]>	Read-only	iptunnel that can be bound to netbridge.
netbridge_vlan_binding	<netbridge_vlan_binding[]>	Read-only	vlan that can be bound to netbridge.
netbridge_nsip_binding	<netbridge_nsip_binding[]>	Read-only	nsip that can be bound to netbridge.
netbridge_nsip6_binding	<netbridge_nsip6_binding[]>	Read-only	nsip6 that can be bound to netbridge.

Operations

(click to see Properties)

GET| GET (ALL)

Some options that you can use for each operations:

- **Getting warnings in response:**NITRO allows you to get warnings in an operation by specifying the "warning" query parameter as "yes". For example, to get warnings while connecting to the Citrix ADC appliance, the URL is as follows:

  http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/login?warning=yes

  If any, the warnings are displayed in the response payload with the HTTP code "209 X-NITRO-WARNING".

- **Authenticated access for individual NITRO operations:**NITRO allows you to logon to the Citrix ADC appliance to perform individual operations. You can use this option instead of creating a NITRO session (using the login object) and then using that session to perform all operations,

  To do this, you must specify the username and password in the request header of the NITRO request as follows:

  X-NITRO-USER:<username>

  X-NITRO-PASS:<password>

  Note: In such cases, make sure that the request header DOES not include the following:

  Cookie:NITRO_AUTH_TOKEN=<tokenvalue>

`*Note:`*

Mandatory parameters are marked in redand placeholder content is marked in <green>.

get

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/netbridge_binding/name_value<String>

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "netbridge_binding": [ {

"name":<String_value>,
```

```
"netbridge_iptunnel_binding":<netbridge_iptunnel_binding[]_value>,

"netbridge_nsip6_binding":<netbridge_nsip6_binding[]_value>,

"netbridge_vlan_binding":<netbridge_vlan_binding[]_value>,

"netbridge_nsip_binding":<netbridge_nsip_binding[]_value>

}]}
```

get (all)

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/netbridge_binding

Query-parameters:

bulkbindings

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/netbridge_binding?**bulkbindings=yes**

NITRO allows you to fetch bindings in bulk.

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "netbridge_binding": [ {

"name":<String_value>,
```

```
"netbridge_iptunnel_binding":<netbridge_iptunnel_binding[]_value>,

"netbridge_nsip6_binding":<netbridge_nsip6_binding[]_value>,

"netbridge_vlan_binding":<netbridge_vlan_binding[]_value>,

"netbridge_nsip_binding":<netbridge_nsip_binding[]_value>

}]}
```

netbridge_iptunnel_binding

Binding object showing the iptunnel that can be bound to netbridge.

Properties

(click to see Operations)

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
name	<String>	Read-write	The name of the network bridge.
tunnel	<String>	Read-write	The name of the tunnel that is a part of this bridge.
__count	<Double>	Read-write	count parameter

Operations

(click to see Properties)

ADD:| DELETE:| GET| GET (ALL)| COUNT

Some options that you can use for each operations:

- **Getting warnings in response:**NITRO allows you to get warnings in an operation by specifying the "warning" query parameter as "yes". For example, to get warnings while connecting to the Citrix ADC appliance, the URL is as follows:

  http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/login?warning=yes

If any, the warnings are displayed in the response payload with the HTTP code "209 X-NITRO-WARNING".

- **Authenticated access for individual NITRO operations:**NITRO allows you to logon to the Citrix ADC appliance to perform individual operations. You can use this option instead of creating a NITRO session (using the login object) and then using that session to perform all operations,

  To do this, you must specify the username and password in the request header of the NITRO request as follows:

  X-NITRO-USER:<username>

  X-NITRO-PASS:<password>

  Note: In such cases, make sure that the request header DOES not include the following:

  Cookie:NITRO_AUTH_TOKEN=<tokenvalue>

`*Note:`*

Mandatory parameters are marked in redand placeholder content is marked in <green>.

add

URL:http://<Citrix-ADC-IP-address(NSIP)/nitro/v1/config/netbridge_iptunnel_binding

HTTP Method:PUT

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Content-Type:application/json

Request Payload:

```
{

"netbridge_iptunnel_binding":{

<b>"name":<String_value>,

</b>"tunnel":<String_value>
```

```
}}
```

Response:

HTTP Status Code on Success: 201 Created HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error

delete

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/netbridge_iptunnel_binding/name_value<String>

Query-parameters:

args

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/netbridge_iptunnel_binding/name_value<String>?**args=tunnel:<String_value>**

HTTP Method:DELETE

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue>

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error

get

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/netbridge_iptunnel_binding/name_value<String>

Query-parameters:

filter

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/netbridge_iptunnel_binding/name_value<String>?**filter=property-name1:property-value1,property-name2:property-value2**

Use this query-parameter to get the filtered set of netbridge_iptunnel_binding resources configured on Citrix ADC. Filtering can be done on any of the properties of the resource.

pagination

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/netbridge_iptunnel_binding/name_value<String>?**pagesize=#no;pageno=#no**

Use this query-parameter to get the netbridge_iptunnel_binding resources in chunks.

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "netbridge_iptunnel_binding": [ {

"name":<String_value>,

"tunnel":<String_value>

}]}
```

get (all)

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/netbridge_iptunnel_binding

Query-parameters:

bulkbindings

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/netbridge_iptunnel_binding?**bulkbindings=yes**

NITRO allows you to fetch bindings in bulk.

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "netbridge_iptunnel_binding": [ {

"name":<String_value>,

"tunnel":<String_value>

}]}
```

count

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/netbridge_iptunnel_binding/name_value<String>?**count=yes**

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{"netbridge_iptunnel_binding": [ { "__count": "#no"} ] }
```

netbridge_nsip6_binding

Binding object showing the nsip6 that can be bound to netbridge.

Properties

(click to see Operations)

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
name	<String>	Read-write	The name of the network bridge.
netmask	<String>	Read-write	The network mask for the subnet.
ipaddress	<String>	Read-write	The subnet that is extended by this network bridge. Minimum length = 1
__count	<Double>	Read-write	count parameter

Operations

(click to see Properties)

ADD:| DELETE:| GET| GET (ALL)| COUNT

Some options that you can use for each operations:

- **Getting warnings in response:**NITRO allows you to get warnings in an operation by specifying the "warning" query parameter as "yes". For example, to get warnings while connecting to the Citrix ADC appliance, the URL is as follows:

  http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/login?warning=yes

  If any, the warnings are displayed in the response payload with the HTTP code "209 X-NITRO-WARNING".

- **Authenticated access for individual NITRO operations:**NITRO allows you to logon to the Citrix ADC appliance to perform individual operations. You can use this option instead of creating a NITRO session (using the login object) and then using that session to perform all operations,

  To do this, you must specify the username and password in the request header of the NITRO request as follows:

  X-NITRO-USER:<username>

  X-NITRO-PASS:<password>

  Note: In such cases, make sure that the request header DOES not include the following:

  Cookie:NITRO_AUTH_TOKEN=<tokenvalue>

`*Note:`*

Mandatory parameters are marked in redand placeholder content is marked in <green>.

add

URL:http://<Citrix-ADC-IP-address(NSIP)/nitro/v1/config/netbridge_nsip6_binding

HTTP Method:PUT

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Content-Type:application/json

Request Payload:

```
{
```

```
"netbridge_nsip6_binding":{

<b>"name":<String_value>,

</b>"ipaddress":<String_value>,

"netmask":<String_value>

}}
```

Response:

HTTP Status Code on Success: 201 Created HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error

delete

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/netbridge_nsip6_binding/name_value<String>

Query-parameters:

args

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/netbridge_nsip6_binding/name_value<String>?**args=ipaddress:<String_value>,netmask:<String_value>**

HTTP Method:DELETE

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue>

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error

get

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/netbridge_nsip6_binding/name_value<String>

Query-parameters:

filter

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/netbridge_nsip6_binding/name_value<String>?**filter=property-name1:property-value1,property-name2:property-value2**

Use this query-parameter to get the filtered set of netbridge_nsip6_binding resources configured on Citrix ADC. Filtering can be done on any of the properties of the resource.

pagination

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/netbridge_nsip6_binding/name_value<String>?**pagesize=#no;pageno=#no**

Use this query-parameter to get the netbridge_nsip6_binding resources in chunks.

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "netbridge_nsip6_binding": [ {

"name":<String_value>,

"netmask":<String_value>,
```

```
"ipaddress":<String_value>

}]}
```

get (all)

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/netbridge_nsip6_binding

Query-parameters:

bulkbindings

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/netbridge_nsip6_binding?**bulkbindings=yes**

NITRO allows you to fetch bindings in bulk.

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "netbridge_nsip6_binding": [ {

"name":<String_value>,

"netmask":<String_value>,

"ipaddress":<String_value>

}]}
```

count

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/netbridge_nsip6_binding/name_value<String>?**count=yes**

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{"netbridge_nsip6_binding": [ { "__count": "#no"} ] }
```

netbridge_nsip_binding

Binding object showing the nsip that can be bound to netbridge.

Properties

(click to see Operations)

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
name	<String>	Read-write	The name of the network bridge.
netmask	<String>	Read-write	The network mask for the subnet.

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
ipaddress	<String>	Read-write	The subnet that is extended by this network bridge. Minimum length = 1
__count	<Double>	Read-write	count parameter

Operations

(click to see Properties)

ADD:| DELETE:| GET| GET (ALL)| COUNT

Some options that you can use for each operations:

- **Getting warnings in response:**NITRO allows you to get warnings in an operation by specifying the "warning" query parameter as "yes". For example, to get warnings while connecting to the Citrix ADC appliance, the URL is as follows:

  http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/login?warning=yes

  If any, the warnings are displayed in the response payload with the HTTP code "209 X-NITRO-WARNING".

- **Authenticated access for individual NITRO operations:**NITRO allows you to logon to the Citrix ADC appliance to perform individual operations. You can use this option instead of creating a NITRO session (using the login object) and then using that session to perform all operations,

  To do this, you must specify the username and password in the request header of the NITRO request as follows:

  X-NITRO-USER:<username>

  X-NITRO-PASS:<password>

  Note: In such cases, make sure that the request header DOES not include the following:

  Cookie:NITRO_AUTH_TOKEN=<tokenvalue>

`*Note:`*

Mandatory parameters are marked in redand placeholder content is marked in <green>.

add

URL:http://<Citrix-ADC-IP-address(NSIP)/nitro/v1/config/netbridge_nsip_binding

HTTP Method:PUT

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Content-Type:application/json

Request Payload:

```
{

"netbridge_nsip_binding":{

<b>"name":<String_value>,

</b>"ipaddress":<String_value>,

"netmask":<String_value>

}}
```

Response:

HTTP Status Code on Success: 201 Created HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error

delete

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/netbridge_nsip_binding/name_value<String>

Query-parameters:

args

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/netbridge_nsip_binding/name_value<String>?**args=ipaddress:<String_value>,netmask:<String_value>**

HTTP Method:DELETE

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue>

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error

get

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/netbridge_nsip_binding/name_value<String>

Query-parameters:

filter

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/netbridge_nsip_binding/name_value<String>?**filter=property-name1:property-value1,property-name2:property-value2**

Use this query-parameter to get the filtered set of netbridge_nsip_binding resources configured on Citrix ADC. Filtering can be done on any of the properties of the resource.

pagination

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/netbridge_nsip_binding/name_value<String>?**pagesize=#no;pageno=#no**

Use this query-parameter to get the netbridge_nsip_binding resources in chunks.

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "netbridge_nsip_binding": [ {

"name":<String_value>,

"netmask":<String_value>,

"ipaddress":<String_value>

}]}
```

get (all)

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/netbridge_nsip_binding

Query-parameters:

bulkbindings

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/netbridge_nsip_binding?**bulkbindings=yes**

NITRO allows you to fetch bindings in bulk.

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "netbridge_nsip_binding": [ {

"name":<String_value>,

"netmask":<String_value>,

"ipaddress":<String_value>

}]}
```

count

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/netbridge_nsip_binding/name_value<String>?**count=yes**

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{"netbridge_nsip_binding": [ { "__count": "#no"} ] }
```

netbridge_vlan_binding

Binding object showing the vlan that can be bound to netbridge.

Properties

(click to see Operations)

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
name	<String>	Read-write	The name of the network bridge.
vlan	<Double>	Read-write	The VLAN that is extended by this network bridge. Minimum value = 1 Maximum value = 4094
__count	<Double>	Read-write	count parameter

Operations

(click to see Properties)

ADD:| DELETE:| GET| GET (ALL)| COUNT

Some options that you can use for each operations:

- **Getting warnings in response:**NITRO allows you to get warnings in an operation by specifying the "warning" query parameter as "yes". For example, to get warnings while connecting to the Citrix ADC appliance, the URL is as follows:

  http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/login?warning=yes

  If any, the warnings are displayed in the response payload with the HTTP code "209 X-NITRO-WARNING".

- **Authenticated access for individual NITRO operations:**NITRO allows you to logon to the Citrix ADC appliance to perform individual operations. You can use this option instead of creating a NITRO session (using the login object) and then using that session to perform all operations,

  To do this, you must specify the username and password in the request header of the NITRO request as follows:

X-NITRO-USER:<username>

X-NITRO-PASS:<password>

Note: In such cases, make sure that the request header DOES not include the following:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue>

`*Note:`*

Mandatory parameters are marked in redand placeholder content is marked in <green>.

add

URL:http://<Citrix-ADC-IP-address(NSIP)/nitro/v1/config/netbridge_vlan_binding

HTTP Method:PUT

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Content-Type:application/json

Request Payload:

```
{

"netbridge_vlan_binding":{

<b>"name":<String_value>,

</b>"vlan":<Double_value>

}}
```

Response:

HTTP Status Code on Success: 201 Created HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error

delete

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/netbridge_vlan_binding/name_value<String>

Query-parameters:

args

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/netbridge_vlan_binding/name_value<String>?**args=vlan:<Double_value>**

HTTP Method:DELETE

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue>

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error

get

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/netbridge_vlan_binding/name_value<String>

Query-parameters:

filter

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/netbridge_vlan_binding/name_value<String>?**filter=property-name1:property-value1,property-name2:property-value2**

Use this query-parameter to get the filtered set of netbridge_vlan_binding resources configured on Citrix ADC. Filtering can be done on any of the properties of the resource.

pagination

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/netbridge_vlan_binding/name_value<String>?**pagesize=#no;pageno=#no**

Use this query-parameter to get the netbridge_vlan_binding resources in chunks.

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "netbridge_vlan_binding": [ {

"name":<String_value>,

"vlan":<Double_value>

}]}
```

get (all)

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/netbridge_vlan_binding

Query-parameters:

bulkbindings

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/netbridge_vlan_binding?**bulkbindings=yes**

NITRO allows you to fetch bindings in bulk.

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "netbridge_vlan_binding": [ {

"name":<String_value>,

"vlan":<Double_value>

}]}
```

count

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/netbridge_vlan_binding/name_value<String>?**count=yes**

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{"netbridge_vlan_binding": [ { "__count": "#no"} ] }
```

netprofile

Configuration for Network profile resource.

Properties

(click to see Operations)

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
name	<String>	Read-write	Name for the net profile. Must begin with a letter, number, or the underscore character (_), and can consist of letters, numbers, and the hyphen (-), period (.) pound (#), space ( ), at sign (@), equals (=), colon (:), and underscore characters. Cannot be changed after the profile is created. Choose a name that helps identify the net profile. Minimum length = 1
td	<Double>	Read-write	Integer value that uniquely identifies the traffic domain in which you want to configure the entity. If you do not specify an ID, the entity

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
			becomes part of the default traffic domain, which has an ID of 0. Minimum value = 0 Maximum value = 4094
srcip	<String>	Read-write	IP address or the name of an IP set.
srcippersistency	<String>	Read-write	When the net profile is associated with a virtual server or its bound services, this option enables the Citrix ADC to use the same address, specified in the net profile, to communicate to servers for all sessions initiated from a particular client to the virtual server. Default value: DISABLED Possible values = ENABLED, DISABLED
overridelsn	<String>	Read-write	USNIP/USIP settings override LSN settings for configured

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
			service/virtual server traffic.. . Default value: DISABLED Possible values = ENABLED, DISABLED
mbf	<String>	Read-write	Response will be sent using learnt info if enabled. When creating a netprofile, if you do not set this parameter, the netprofile inherits the global MBF setting (available in the enable ns mode and disable ns mode CLI commands, or in the System > Settings > Configure modes > Configure Modes dialog box). However, you can override this setting after you create the netprofile. Possible values = ENABLED, DISABLED
proxyprotocol	<String>	Read-write	Proxy Protocol Action (Enabled/ Disabled). Default value: DISABLED

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
			Possible values = ENABLED, DISABLED
proxyprotocoltxversion	<String>	Read-write	Proxy Protocol Version (V1/V2). Default value: V1 Possible values = V1, V2
__count	<Double>	Read-only	count parameter

Operations

(click to see Properties)

ADD| DELETE| UPDATE| UNSET| GET (ALL)| GET| COUNT

Some options that you can use for each operations:

- **Getting warnings in response:**NITRO allows you to get warnings in an operation by specifying the "warning" query parameter as "yes". For example, to get warnings while connecting to the Citrix ADC appliance, the URL is as follows:

  http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/login?warning=yes

  If any, the warnings are displayed in the response payload with the HTTP code "209 X-NITRO-WARNING".

- **Authenticated access for individual NITRO operations:**NITRO allows you to logon to the Citrix ADC appliance to perform individual operations. You can use this option instead of creating a NITRO session (using the login object) and then using that session to perform all operations,

  To do this, you must specify the username and password in the request header of the NITRO request as follows:

  X-NITRO-USER:<username>

  X-NITRO-PASS:<password>

Note: In such cases, make sure that the request header DOES not include the following:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue>

`*Note:` *

Mandatory parameters are marked in redand placeholder content is marked in <green>.

add

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/netprofile

HTTP Method:POST

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Content-Type:application/json

Request Payload:

```
{"netprofile":{

<b>"name":<String_value>,

</b>"td":<Double_value>,

"srcip":<String_value>,

"srcippersistency":<String_value>,

"overridelsn":<String_value>,

"mbf":<String_value>,

"proxyprotocol":<String_value>,

"proxyprotocoltxversion":<String_value>

}}
```

Response:

HTTP Status Code on Success: 201 Created HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error

delete

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/netprofile/name_value<String>

HTTP Method:DELETE

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue>

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error

update

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/netprofile

HTTP Method:PUT

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Content-Type:application/json

Request Payload:

```
{"netprofile":{

<b>"name":<String_value>,

</b>"srcip":<String_value>,

"srcippersistency":<String_value>,

"overridelsn":<String_value>,

"mbf":<String_value>,
```

```
"proxyprotocol":<String_value>,

"proxyprotocoltxversion":<String_value>

}}
```

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error

unset

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/netprofile?**action=unset**

HTTP Method:POST

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Content-Type:application/json

Request Payload:

```
{"netprofile":{

<b>"name":<String_value>,

</b>"srcip":true,

"srcippersistency":true,

"overridelsn":true,

"mbf":true,

"proxyprotocol":true,

"proxyprotocoltxversion":true
```

```
}}
```

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error

get (all)

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/netprofile

Query-parameters:

attrs

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/netprofile?**attrs=property-name1,property-name2**

Use this query parameter to specify the resource details that you want to retrieve.

filter

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/netprofile?**filter=property-name1:property-val1,property-name2:property-val2**

Use this query-parameter to get the filtered set of netprofile resources configured on Citrix ADC. Filtering can be done on any of the properties of the resource.

view

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/netprofile?**view=summary**

Note: By default, the retrieved results are displayed in detail view (?view=detail).

pagination

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/netprofile?**pagesize=#no;pageno=#no**

Use this query-parameter to get the netprofile resources in chunks.

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "netprofile": [ {

"name":<String_value>,

"srcip":<String_value>,

"td":<Double_value>,

"srcippersistency":<String_value>,

"overridelsn":<String_value>,

"mbf":<String_value>,

"proxyprotocol":<String_value>,

"proxyprotocoltxversion":<String_value>

}]}
```

get

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/netprofile/name_value<String>

Query-parameters:

attrs

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/netprofile/name_value<String>?**attrs=property-name1,property-name2**

Use this query parameter to specify the resource details that you want to retrieve.

view

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/netprofile/name_value<String>?**view=summary**

Note: By default, the retrieved results are displayed in detail view (?view=detail).

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "netprofile": [ {

"name":<String_value>,

"srcip":<String_value>,

"td":<Double_value>,

"srcippersistency":<String_value>,

"overridelsn":<String_value>,

"mbf":<String_value>,
```

```
"proxyprotocol":<String_value>,

"proxyprotocoltxversion":<String_value>

}]}
```

count

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/netprofile?**count=yes**

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "netprofile": [ { "__count": "#no"} ] }
```

netprofile_binding

Binding object which returns the resources bound to netprofile.

Properties

(click to see Operations)

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
name	<String>	Read-write	Name of the net profile whose details you want to display. Minimum length = 1
netprofile_srcportset_binding	<netprofile_srcportset_binding[]>	Read-only	srcportset that can be bound to netprofile.
netprofile_natrule_binding	<netprofile_natrule_binding[]>	Read-only	natrule that can be bound to netprofile.

Operations

(click to see Properties)

GET| GET (ALL)

Some options that you can use for each operations:

- **Getting warnings in response:**NITRO allows you to get warnings in an operation by specifying the "warning" query parameter as "yes". For example, to get warnings while connecting to the Citrix ADC appliance, the URL is as follows:

  http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/login?warning=yes

  If any, the warnings are displayed in the response payload with the HTTP code "209 X-NITRO-WARNING".

- **Authenticated access for individual NITRO operations:**NITRO allows you to logon to the Citrix ADC appliance to perform individual operations. You can use this option instead of creating a NITRO session (using the login object) and then using that session to perform all operations,

  To do this, you must specify the username and password in the request header of the NITRO request as follows:

  X-NITRO-USER:<username>

  X-NITRO-PASS:<password>

  Note: In such cases, make sure that the request header DOES not include the following:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue>

`*Note:`*

Mandatory parameters are marked in redand placeholder content is marked in <green>.

get

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/netprofile_binding/name_value<String>

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "netprofile_binding": [ {

"name":<String_value>,

"netprofile_srcportset_binding":<netprofile_srcportset_binding[]_value>,

"netprofile_natrule_binding":<netprofile_natrule_binding[]_value>

}]}
```

get (all)

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/netprofile_binding

Query-parameters:

bulkbindings

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/netprofile_binding?**bulkbindings=yes**

NITRO allows you to fetch bindings in bulk.

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "netprofile_binding": [ {

"name":<String_value>,

"netprofile_srcportset_binding":<netprofile_srcportset_binding[]_value>,

"netprofile_natrule_binding":<netprofile_natrule_binding[]_value>

}]}
```

netprofile_natrule_binding

Binding object showing the natrule that can be bound to netprofile.

Properties

(click to see Operations)

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
rewriteip	<String>	Read-write	.

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
name	<String>	Read-write	Name of the netprofile to which to bind port ranges. Minimum length = 1
netmask	<String>	Read-write	.
natrule	<String>	Read-write	IPv4 network address on whose traffic you want the Citrix ADC to do rewrite ip prefix.
__count	<Double>	Read-write	count parameter

Operations

(click to see Properties)

ADD:| DELETE:| GET| GET (ALL)| COUNT

Some options that you can use for each operations:

- **Getting warnings in response:**NITRO allows you to get warnings in an operation by specifying the "warning" query parameter as "yes". For example, to get warnings while connecting to the Citrix ADC appliance, the URL is as follows:

  http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/login?warning=yes

  If any, the warnings are displayed in the response payload with the HTTP code "209 X-NITRO-WARNING".

- **Authenticated access for individual NITRO operations:**NITRO allows you to logon to the Citrix ADC appliance to perform individual operations. You can use this option instead of creating a NITRO session (using the login object) and then using that session to perform all operations,

  To do this, you must specify the username and password in the request header of the NITRO request as follows:

  X-NITRO-USER:<username>

X-NITRO-PASS:<password>

Note: In such cases, make sure that the request header DOES not include the following:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue>

`*Note:`*

Mandatory parameters are marked in redand placeholder content is marked in <green>.

add

URL:http://<Citrix-ADC-IP-address(NSIP)/nitro/v1/config/netprofile_natrule_binding

HTTP Method:PUT

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Content-Type:application/json

Request Payload:

```
{
"netprofile_natrule_binding":{
<b>"name":<String_value>,
</b>"natrule":<String_value>,
"netmask":<String_value>,
"rewriteip":<String_value>
}}
```

Response:

HTTP Status Code on Success: 201 Created HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error

delete

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/netprofile_natrule_binding/name_value<String>

Query-parameters:

args

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/netprofile_natrule_binding/name_value<String>?**args=natrule:<String_value>,netmask:<String_value>**

HTTP Method:DELETE

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue>

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error

get

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/netprofile_natrule_binding/name_value<String>

Query-parameters:

filter

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/netprofile_natrule_binding/name_value<String>?**filter=property-name1:property-value1,property-name2:property-value2**

Use this query-parameter to get the filtered set of netprofile_natrule_binding resources configured on Citrix ADC. Filtering can be done on any of the properties of the resource.

pagination

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/netprofile_natrule_binding/name_value<String>?**pagesize=#no;pageno=#no**

Use this query-parameter to get the netprofile_natrule_binding resources in chunks.

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "netprofile_natrule_binding": [ {

"rewriteip":<String_value>,

"name":<String_value>,

"netmask":<String_value>,

"natrule":<String_value>

}]}
```

get (all)

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/netprofile_natrule_binding

Query-parameters:

bulkbindings

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/netprofile_natrule_binding?**bulkbindings=yes**

NITRO allows you to fetch bindings in bulk.

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "netprofile_natrule_binding": [ {

"rewriteip":<String_value>,

"name":<String_value>,

"netmask":<String_value>,

"natrule":<String_value>

}]}
```

count

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/netprofile_natrule_binding/name_value<String>?**count=yes**

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{"netprofile_natrule_binding": [ { "__count": "#no"} ] }
```

netprofile_srcportset_binding

Binding object showing the srcportset that can be bound to netprofile.

Properties

(click to see Operations)

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
srcportrange	<String>	Read-write	When the source port range is configured and associated with the netprofile bound to a service group, Citrix ADC will choose a port from the range configured for connection establishment at the backend servers. Minimum length = 1024 Maximum length = 65535

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
name	<String>	Read-write	Name of the netprofile to which to bind port ranges. Minimum length = 1
__count	<Double>	Read-write	count parameter

Operations

(click to see Properties)

ADD:| DELETE:| GET| GET (ALL)| COUNT

Some options that you can use for each operations:

- **Getting warnings in response:**NITRO allows you to get warnings in an operation by specifying the "warning" query parameter as "yes". For example, to get warnings while connecting to the Citrix ADC appliance, the URL is as follows:

  http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/login?warning=yes

  If any, the warnings are displayed in the response payload with the HTTP code "209 X-NITRO-WARNING".

- **Authenticated access for individual NITRO operations:**NITRO allows you to logon to the Citrix ADC appliance to perform individual operations. You can use this option instead of creating a NITRO session (using the login object) and then using that session to perform all operations,

  To do this, you must specify the username and password in the request header of the NITRO request as follows:

  X-NITRO-USER:<username>

  X-NITRO-PASS:<password>

  Note: In such cases, make sure that the request header DOES not include the following:

  Cookie:NITRO_AUTH_TOKEN=<tokenvalue>

`*Note:`*

Mandatory parameters are marked in redand placeholder content is marked in <green>.

add

URL:http://<Citrix-ADC-IP-address(NSIP)/nitro/v1/config/netprofile_srcportset_binding

HTTP Method:PUT

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Content-Type:application/json

Request Payload:

```
{

"netprofile_srcportset_binding":{

<b>"name":<String_value>,

</b>"srcportrange":<String_value>

}}
```

Response:

HTTP Status Code on Success: 201 Created HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error

delete

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/netprofile_srcportset_binding/name_value<String>

Query-parameters:

args

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/netprofile_srcportset_binding/name_value<String>?**args=srcportrange:<String_value>**

HTTP Method:DELETE

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue>

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error

get

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/netprofile_srcportset_binding/name_value<String>

Query-parameters:

filter

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/netprofile_srcportset_binding/name_value<String>?**filter=property-name1:property-value1,property-name2:property-value2**

Use this query-parameter to get the filtered set of netprofile_srcportset_binding resources configured on Citrix ADC. Filtering can be done on any of the properties of the resource.

pagination

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/netprofile_srcportset_binding/name_value<String>?**pagesize=#no;pageno=#no**

Use this query-parameter to get the netprofile_srcportset_binding resources in chunks.

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "netprofile_srcportset_binding": [ {

"srcportrange":<String_value>,

"name":<String_value>

}]}
```

get (all)

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/netprofile_srcportset_binding

Query-parameters:

bulkbindings

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/netprofile_srcportset_binding?**bulkbindings=yes**

NITRO allows you to fetch bindings in bulk.

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "netprofile_srcportset_binding": [ {

"srcportrange":<String_value>,

"name":<String_value>

}]}
```

count

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/netprofile_srcportset_binding/name_value<String>?**count=yes**

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{"netprofile_srcportset_binding": [ { "__count": "#no"} ] }
```

onlinkipv6prefix

Configuration for on-link IPv6 global prefixes for Router Advertisment resource.

Properties

(click to see Operations)

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
ipv6prefix	<String>	Read-write	Onlink prefixes for RA messages.
onlinkprefix	<String>	Read-write	RA Prefix onlink flag. Default value: YES Possible values = YES, NO
autonomusprefix	<String>	Read-write	RA Prefix Autonomus flag. Default value: YES Possible values = YES, NO
depricateprefix	<String>	Read-write	Depricate the prefix. Default value: NO Possible values = YES, NO
decrementprefixlifetimes	<String>	Read-write	RA Prefix Autonomus flag. Default value: NO Possible values = YES, NO
prefixvalidelifetime	<Double>	Read-write	Valide life time of the prefix, in seconds. Default value: 2592000
prefixpreferredlifetime	<Double>	Read-write	Preferred life time of the prefix, in seconds.

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
			Default value: 604800
prefixcurrvalidelft	<Double>	Read-only	Prefix current valid life time.
prefixcurrpreferredlft	<Double>	Read-only	Prefix current prefered life time.
__count	<Double>	Read-only	count parameter

Operations

(click to see Properties)

ADD| DELETE| UPDATE| UNSET| GET (ALL)| GET| COUNT

Some options that you can use for each operations:

- **Getting warnings in response:**NITRO allows you to get warnings in an operation by specifying the "warning" query parameter as "yes". For example, to get warnings while connecting to the Citrix ADC appliance, the URL is as follows:

  http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/login?warning=yes

  If any, the warnings are displayed in the response payload with the HTTP code "209 X-NITRO-WARNING".

- **Authenticated access for individual NITRO operations:**NITRO allows you to logon to the Citrix ADC appliance to perform individual operations. You can use this option instead of creating a NITRO session (using the login object) and then using that session to perform all operations,

  To do this, you must specify the username and password in the request header of the NITRO request as follows:

  X-NITRO-USER:<username>

  X-NITRO-PASS:<password>

  Note: In such cases, make sure that the request header DOES not include the following:

  Cookie:NITRO_AUTH_TOKEN=<tokenvalue>

`*Note:` *

Mandatory parameters are marked in redand placeholder content is marked in <green>.

add

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/onlinkipv6prefix

HTTP Method:POST

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Content-Type:application/json

Request Payload:

```
{"onlinkipv6prefix":{

<b>"ipv6prefix":<String_value>,

</b>"onlinkprefix":<String_value>,

"autonomusprefix":<String_value>,

"depricateprefix":<String_value>,

"decrementprefixlifetimes":<String_value>,

"prefixvalidelifetime":<Double_value>,

"prefixpreferredlifetime":<Double_value>

}}
```

Response:

HTTP Status Code on Success: 201 Created HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error

delete

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/onlinkipv6prefix/ipv6prefix_value<String>

HTTP Method:DELETE

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue>

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error

update

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/onlinkipv6prefix

HTTP Method:PUT

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Content-Type:application/json

Request Payload:

```
{"onlinkipv6prefix":{

<b>"ipv6prefix":<String_value>,

</b>"onlinkprefix":<String_value>,

"autonomusprefix":<String_value>,

"depricateprefix":<String_value>,

"decrementprefixlifetimes":<String_value>,

"prefixvalidelifetime":<Double_value>,

"prefixpreferredlifetime":<Double_value>
```

```
}}
```

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error

unset

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/onlinkipv6prefix?**action=unset**

HTTP Method:POST

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Content-Type:application/json

Request Payload:

```
{"onlinkipv6prefix":{

<b>"ipv6prefix":<String_value>,

</b>"onlinkprefix":true,

"autonomusprefix":true,

"depricateprefix":true,

"decrementprefixlifetimes":true,

"prefixvalidelifetime":true,

"prefixpreferredlifetime":true

}}
```

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error

get (all)

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/onlinkipv6prefix

Query-parameters:

attrs

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/onlinkipv6prefix?**attrs=property-name1,property-name2**

Use this query parameter to specify the resource details that you want to retrieve.

filter

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/onlinkipv6prefix?**filter=property-name1:property-val1,property-name2:property-val2**

Use this query-parameter to get the filtered set of onlinkipv6prefix resources configured on Citrix ADC. Filtering can be done on any of the properties of the resource.

view

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/onlinkipv6prefix?**view=summary**

Note: By default, the retrieved results are displayed in detail view (?view=detail).

pagination

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/onlinkipv6prefix?**pagesize=#no;pageno=#no**

Use this query-parameter to get the onlinkipv6prefix resources in chunks.

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "onlinkipv6prefix": [ {

"ipv6prefix":<String_value>,

"onlinkprefix":<String_value>,

"autonomusprefix":<String_value>,

"depricateprefix":<String_value>,

"decrementprefixlifetimes":<String_value>,

"prefixvalidelifetime":<Double_value>,

"prefixpreferredlifetime":<Double_value>,

"prefixcurrvalidelft":<Double_value>,

"prefixcurrpreferredlft":<Double_value>

}]}
```

get

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/onlinkipv6prefix/ipv6prefix_value<String>

Query-parameters:

attrs

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/onlinkipv6prefix/ipv6prefix_value<String>?**attrs=property-name1,property-name2**

Use this query parameter to specify the resource details that you want to retrieve.

view

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/onlinkipv6prefix/ipv6prefix_value<String>?**view=summary**

Note: By default, the retrieved results are displayed in detail view (?view=detail).

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "onlinkipv6prefix": [ {

"ipv6prefix":<String_value>,

"onlinkprefix":<String_value>,

"autonomusprefix":<String_value>,

"depricateprefix":<String_value>,

"decrementprefixlifetimes":<String_value>,

"prefixvalidelifetime":<Double_value>,
```

```
"prefixpreferredlifetime":<Double_value>,

"prefixcurrvalidelft":<Double_value>,

"prefixcurrpreferredlft":<Double_value>

}]}
```

count

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/onlinkipv6prefix?**count=yes**

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "onlinkipv6prefix": [ { "__count": "#no"} ] }
```

ptp

Configuration for Precision Time Protocol resource.

Properties

(click to see Operations)

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
state	<String>	Read-write	Enables or disables Precision Time Protocol (PTP) on the appliance. If you disable PTP, make sure you enable Network Time Protocol (NTP) on the cluster. Default value: ENABLE Possible values = DISABLE, ENABLE

Operations

(click to see Properties)

UPDATE| GET (ALL)

Some options that you can use for each operations:

- **Getting warnings in response:**NITRO allows you to get warnings in an operation by specifying the "warning" query parameter as "yes". For example, to get warnings while connecting to the Citrix ADC appliance, the URL is as follows:

  http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/login?warning=yes

  If any, the warnings are displayed in the response payload with the HTTP code "209 X-NITRO-WARNING".

- **Authenticated access for individual NITRO operations:**NITRO allows you to logon to the Citrix ADC appliance to perform individual operations. You can use this option instead of creating a NITRO session (using the login object) and then using that session to perform all operations,

  To do this, you must specify the username and password in the request header of the NITRO request as follows:

  X-NITRO-USER:<username>

X-NITRO-PASS:<password>

Note: In such cases, make sure that the request header DOES not include the following:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue>

`*Note:`*

Mandatory parameters are marked in redand placeholder content is marked in <green>.

update

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/ptp

HTTP Method:PUT

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Content-Type:application/json

Request Payload:

```
{"ptp":{

<b>"state":<String_value>

</b>}}
```

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error

get (all)

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/ptp

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "ptp": [ {

"state":<String_value>

}]}
```

rnat

Configuration for RNAT configured route resource.

Properties

(click to see Operations)

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
network	<String>	Read-write	The network address defined for the RNAT entry. Minimum length = 1
netmask	<String>	Read-write	The subnet mask for the network address. Minimum length = 1
aclname	<String>	Read-write	An extended ACL defined for the RNAT entry. Minimum length = 1

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
td	<Double>	Read-write	Integer value that uniquely identifies the traffic domain in which you want to configure the entity. If you do not specify an ID, the entity becomes part of the default traffic domain, which has an ID of 0. Minimum value = 0 Maximum value = 4094
ownergroup	<String>	Read-write	The owner node group in a Cluster for this rnat rule. Default value: DEFAULT_NG Minimum length = 1
name	<String>	Read-write	Name for the RNAT4 rule. Must begin with a letter, number, or the underscore character (_), and can consist of letters, numbers, and the hyphen (-), period (.) pound (#), space ( ), at sign (@), equals (=), colon (:), and underscore characters. Cannot be changed after the rule is created. Choose a name that

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
			helps identify the RNAT4 rule. Minimum length = 1
redirectport	<Integer>	Read-write	Port number to which the IPv4 packets are redirected. Applicable to TCP and UDP protocols. Minimum value = 1 Maximum value = 65535
natip	<String>	Read-write	Any Citrix ADC owned IPv4 address except the NSIP address. The Citrix ADC appliance replaces the source IP addresses of server-generated packets with the IP address specified. The IP address must be a public Citrix ADC owned IP address. If you specify multiple addresses for this field, NATIP selection uses the round robin algorithm for each session. By specifying a range of IP addresses, you can specify all Citrix ADC owned IP addresses, except the NSIP, that fall

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
			within the specified range. Minimum length = 1
srcippersistency	<String>	Read-write	Enables the Citrix ADC to use the same NAT IP address for all RNAT sessions initiated from a particular server. Default value: DISABLED Possible values = ENABLED, DISABLED
useproxyport	<String>	Read-write	Enable source port proxying, which enables the Citrix ADC to use the RNAT ips using proxied source port. Default value: ENABLED Possible values = ENABLED, DISABLED
connfailover	<String>	Read-write	Synchronize connection information with the secondary appliance in a high availability (HA) pair. That is, synchronize all connection-related information for the

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
			RNAT session. In order for this to work, tcpproxy should be DISABLED. To disable tcpproxy use "set rnatparam tcpproxy DISABLED". Default value: DISABLED Possible values = ENABLED, DISABLED
newname	<String>	Read-write	New name for the RNAT4 rule. Must begin with an ASCII alphabetic or underscore (_) character, and must contain only ASCII alphanumeric, underscore, hash (#), period (.), space, colon (:), at (@), equals (=), and hyphen (-) characters. Minimum length = 1
__count	<Double>	Read-only	count parameter

Operations

(click to see Properties)

CLEAR| UPDATE| UNSET| ADD| RENAME| DELETE| GET (ALL)| GET| COUNT

Some options that you can use for each operations:

- **Getting warnings in response:**NITRO allows you to get warnings in an operation by specifying the "warning" query parameter as "yes". For example, to get warnings while connecting to the Citrix ADC appliance, the URL is as follows:

  http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/login?warning=yes

  If any, the warnings are displayed in the response payload with the HTTP code "209 X-NITRO-WARNING".

- **Authenticated access for individual NITRO operations:**NITRO allows you to logon to the Citrix ADC appliance to perform individual operations. You can use this option instead of creating a NITRO session (using the login object) and then using that session to perform all operations,

  To do this, you must specify the username and password in the request header of the NITRO request as follows:

  X-NITRO-USER:<username>

  X-NITRO-PASS:<password>

  Note: In such cases, make sure that the request header DOES not include the following:

  Cookie:NITRO_AUTH_TOKEN=<tokenvalue>

`*Note:`*

Mandatory parameters are marked in redand placeholder content is marked in <green>.

clear

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/rnat?**action=clear**

HTTP Method:POST

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Content-Type:application/json

Request Payload:

```
{"rnat":{
```

```
"network":<String_value>,

"netmask":<String_value>,

"aclname":<String_value>,

"td":<Double_value>,

"ownergroup":<String_value>

}}
```

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error

update

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/rnat

HTTP Method:PUT

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Content-Type:application/json

Request Payload:

```
{"rnat":{

"name":<String_value>,

"network":<String_value>,

"netmask":<String_value>,

"aclname":<String_value>,
```

```
"redirectport":<Integer_value>,

"td":<Double_value>,

"natip":<String_value>,

"srcippersistency":<String_value>,

"useproxyport":<String_value>,

"ownergroup":<String_value>,

"connfailover":<String_value>

}}
```

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error

unset

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/rnat?**action=unset**

HTTP Method:POST

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Content-Type:application/json

Request Payload:

```
{"rnat":{

"name":<String_value>,

"network":true,
```

```
"netmask":true,

"aclname":true,

"td":true,

"redirectport":true,

"natip":true,

"srcippersistency":true,

"useproxyport":true,

"connfailover":true,

"ownergroup":true

}}
```

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error

add

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/rnat

HTTP Method:POST

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Content-Type:application/json

Request Payload:

```
{"rnat":{
```

```
<b>"name":<String_value>,

</b>"network":<String_value>,

"netmask":<String_value>,

"aclname":<String_value>,

"redirectport":<Integer_value>,

"td":<Double_value>,

"srcippersistency":<String_value>,

"useproxyport":<String_value>,

"connfailover":<String_value>,

"ownergroup":<String_value>

}}
```

Response:

HTTP Status Code on Success: 201 Created HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error

rename

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/rnat?**action=rename**

HTTP Method:POST

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Content-Type:application/json

Request Payload:

```
{"rnat":{

<b>"name":<String_value>,

</b><b>"newname":<String_value>

</b>}}
```

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error

delete

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/rnat/name_value<String>

HTTP Method:DELETE

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue>

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error

get (all)

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/rnat

Query-parameters:

attrs

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/rnat?**attrs=property-name1,property-name2**

Use this query parameter to specify the resource details that you want to retrieve.

filter

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/rnat?**filter=property-name1:property-val1,property-name2:property-val2**

Use this query-parameter to get the filtered set of rnat resources configured on Citrix ADC. Filtering can be done on any of the properties of the resource.

view

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/rnat?**view=summary**

Note: By default, the retrieved results are displayed in detail view (?view=detail).

pagination

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/rnat?**pagesize=#no;pageno=#no**

Use this query-parameter to get the rnat resources in chunks.

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "rnat": [ {

"name":<String_value>,

"network":<String_value>,

"netmask":<String_value>,
```

```
"td":<Double_value>,

"natip":<String_value>,

"aclname":<String_value>,

"redirectport":<Integer_value>,

"srcippersistency":<String_value>,

"useproxyport":<String_value>,

"ownergroup":<String_value>,

"connfailover":<String_value>

}]}
```

get

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/rnat/name_value<String>

Query-parameters:

attrs

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/rnat/name_value<String>?**attrs=property-name1,property-name2**

Use this query parameter to specify the resource details that you want to retrieve.

view

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/rnat/name_value<String>?**view=summary**

Note: By default, the retrieved results are displayed in detail view (?view=detail).

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "rnat": [ {

"name":<String_value>,

"network":<String_value>,

"netmask":<String_value>,

"td":<Double_value>,

"natip":<String_value>,

"aclname":<String_value>,

"redirectport":<Integer_value>,

"srcippersistency":<String_value>,

"useproxyport":<String_value>,

"ownergroup":<String_value>,

"connfailover":<String_value>

}]}
```

count

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/rnat?**count=yes**

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "rnat": [ { "__count": "#no"} ] }
```

rnatsession

Configuration for RNAT session resource.

Properties

(click to see Operations)

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
network	<String>	Read-write	IPv4 network address on whose traffic you want the Citrix ADC to do RNAT processing. Minimum length = 1
netmask	<String>	Read-write	Subnet mask associated with the network address. Minimum length = 1

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
natip	<String>	Read-write	The NAT IP address defined for the RNAT entry. Minimum length = 1
aclname	<String>	Read-write	Name of any configured extended ACL whose action is ALLOW. Minimum length = 1

Operations

(click to see Properties)

FLUSH

Some options that you can use for each operations:

- **Getting warnings in response:**NITRO allows you to get warnings in an operation by specifying the "warning" query parameter as "yes". For example, to get warnings while connecting to the Citrix ADC appliance, the URL is as follows:

  http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/login?warning=yes

  If any, the warnings are displayed in the response payload with the HTTP code "209 X-NITRO-WARNING".

- **Authenticated access for individual NITRO operations:**NITRO allows you to logon to the Citrix ADC appliance to perform individual operations. You can use this option instead of creating a NITRO session (using the login object) and then using that session to perform all operations,

  To do this, you must specify the username and password in the request header of the NITRO request as follows:

  X-NITRO-USER:<username>

  X-NITRO-PASS:<password>

  Note: In such cases, make sure that the request header DOES not include the following:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue>

`*Note:` *

Mandatory parameters are marked in redand placeholder content is marked in <green>.

flush

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/rnatsession?**action=flush**

HTTP Method:POST

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Content-Type:application/json

Request Payload:

```
{"rnatsession":{

"network":<String_value>,

"netmask":<String_value>,

"natip":<String_value>,

"aclname":<String_value>

}}
```

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error

rnat6

Configuration for IPv6 RNAT configured route resource.

Properties

(click to see Operations)

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
name	<String>	Read-write	Name for the RNAT6 rule. Must begin with a letter, number, or the underscore character (_), and can consist of letters, numbers, and the hyphen (-), period (.) pound (#), space ( ), at sign (@), equals (=), colon (:), and underscore characters. Cannot be changed after the rule is created. Choose a name that helps identify the RNAT6 rule. Minimum length = 1
network	<String>	Read-write	IPv6 address of the network on whose traffic you want the Citrix ADC to do RNAT processing. Minimum length = 1
acl6name	<String>	Read-write	Name of any configured ACL6 whose action is ALLOW. The rule of the ACL6 is used as an RNAT6 rule. Minimum length = 1
redirectport	<Integer>	Read-write	Port number to which the IPv6 packets are

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
			redirected. Applicable to TCP and UDP protocols. Minimum value = 1 Maximum value = 65535
td	<Double>	Read-write	Integer value that uniquely identifies the traffic domain in which you want to configure the entity. If you do not specify an ID, the entity becomes part of the default traffic domain, which has an ID of 0. Minimum value = 0 Maximum value = 4094
srcippersistency	<String>	Read-write	Enable source ip persistency, which enables the Citrix ADC to use the RNAT ips using source ip. Default value: DISABLED Possible values = ENABLED, DISABLED
ownergroup	<String>	Read-write	The owner node group in a Cluster for this rnat rule.

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
			Default value: DEFAULT_NG Minimum length = 1
__count	<Double>	Read-only	count parameter

Operations

(click to see Properties)

ADD| UPDATE| UNSET| CLEAR| GET (ALL)| GET| COUNT

Some options that you can use for each operations:

- **Getting warnings in response:**NITRO allows you to get warnings in an operation by specifying the "warning" query parameter as "yes". For example, to get warnings while connecting to the Citrix ADC appliance, the URL is as follows:

  http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/login?warning=yes

  If any, the warnings are displayed in the response payload with the HTTP code "209 X-NITRO-WARNING".

- **Authenticated access for individual NITRO operations:**NITRO allows you to logon to the Citrix ADC appliance to perform individual operations. You can use this option instead of creating a NITRO session (using the login object) and then using that session to perform all operations,

  To do this, you must specify the username and password in the request header of the NITRO request as follows:

  X-NITRO-USER:<username>

  X-NITRO-PASS:<password>

  Note: In such cases, make sure that the request header DOES not include the following:

  Cookie:NITRO_AUTH_TOKEN=<tokenvalue>

`*Note:`*

Mandatory parameters are marked in redand placeholder content is marked in <green>.

add

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/rnat6

HTTP Method:POST

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Content-Type:application/json

Request Payload:

```
{"rnat6":{

<b>"name":<String_value>,

</b>"network":<String_value>,

"acl6name":<String_value>,

"redirectport":<Integer_value>,

"td":<Double_value>,

"srcippersistency":<String_value>,

"ownergroup":<String_value>

}}
```

Response:

HTTP Status Code on Success: 201 Created HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error

update

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/rnat6

HTTP Method:PUT

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Content-Type:application/json

Request Payload:

```
{"rnat6":{

<b>"name":<String_value>,

</b>"redirectport":<Integer_value>,

"srcippersistency":<String_value>,

"ownergroup":<String_value>

}}
```

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error

unset

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/rnat6?**action=unset**

HTTP Method:POST

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Content-Type:application/json

Request Payload:

```
{"rnat6":{

<b>"name":<String_value>,
```

```
</b>"redirectport":true,

"srcippersistency":true,

"ownergroup":true

}}
```

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error

clear

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/rnat6?**action=clear**

HTTP Method:POST

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Content-Type:application/json

Request Payload:

```
{"rnat6":{

<b>"name":<String_value>,

</b>"ownergroup":<String_value>

}}
```

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error

get (all)

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/rnat6

Query-parameters:

attrs

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/rnat6?**attrs=property-name1,property-name2**

Use this query parameter to specify the resource details that you want to retrieve.

filter

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/rnat6?**filter=property-name1:property-val1,property-name2:property-val2**

Use this query-parameter to get the filtered set of rnat6 resources configured on Citrix ADC. Filtering can be done on any of the properties of the resource.

view

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/rnat6?**view=summary**

Note: By default, the retrieved results are displayed in detail view (?view=detail).

pagination

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/rnat6?**pagesize=#no;pageno=#no**

Use this query-parameter to get the rnat6 resources in chunks.

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "rnat6": [ {

"name":<String_value>,

"network":<String_value>,

"td":<Double_value>,

"acl6name":<String_value>,

"redirectport":<Integer_value>,

"srcippersistency":<String_value>,

"ownergroup":<String_value>

}]}
```

get

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/rnat6/name_value<String>

Query-parameters:

attrs

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/rnat6/name_value<String>?**attrs=property-name1,property-name2**

Use this query parameter to specify the resource details that you want to retrieve.

view

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/rnat6/name_value<String>?**view=summary**

Note: By default, the retrieved results are displayed in detail view (?view=detail).

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "rnat6": [ {

"name":<String_value>,

"network":<String_value>,

"td":<Double_value>,

"acl6name":<String_value>,

"redirectport":<Integer_value>,

"srcippersistency":<String_value>,

"ownergroup":<String_value>

}]}
```

count

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/rnat6?**count=yes**

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "rnat6": [ { "__count": "#no"} ] }
```

rnat6_binding

Binding object which returns the resources bound to rnat6.

Properties

(click to see Operations)

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
name	<String>	Read-write	Name of the RNAT6 rule whose details you want to display. Minimum length = 1
rnat6_nsip6_binding	<rnat6_nsip6_binding[]>	Read-only	nsip6 that can be bound to rnat6.

Operations

(click to see Properties)

GET| GET (ALL)

Some options that you can use for each operations:

- **Getting warnings in response:**NITRO allows you to get warnings in an operation by specifying the "warning" query parameter as "yes". For example, to get warnings while connecting to the Citrix ADC appliance, the URL is as follows:

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/login?warning=yes

If any, the warnings are displayed in the response payload with the HTTP code "209 X-NITRO-WARNING".

- **Authenticated access for individual NITRO operations:**NITRO allows you to logon to the Citrix ADC appliance to perform individual operations. You can use this option instead of creating a NITRO session (using the login object) and then using that session to perform all operations,

  To do this, you must specify the username and password in the request header of the NITRO request as follows:

  X-NITRO-USER:<username>

  X-NITRO-PASS:<password>

  Note: In such cases, make sure that the request header DOES not include the following:

  Cookie:NITRO_AUTH_TOKEN=<tokenvalue>

`*Note:`*

Mandatory parameters are marked in redand placeholder content is marked in <green>.

get

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/rnat6_binding/name_value<String>

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "rnat6_binding": [ {

"name":<String_value>,

"rnat6_nsip6_binding":<rnat6_nsip6_binding[]_value>

}]}
```

get (all)

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/rnat6_binding

Query-parameters:

bulkbindings

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/rnat6_binding?**bulkbindings=yes**

NITRO allows you to fetch bindings in bulk.

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "rnat6_binding": [ {

"name":<String_value>,
```

```
"rnat6_nsip6_binding":<rnat6_nsip6_binding[]_value>

}]}
```

rnat6_nsip6_binding

Binding object showing the nsip6 that can be bound to rnat6.

Properties

(click to see Operations)

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
natip6	<String>	Read-write	Nat IP Address. Minimum length = 1
name	<String>	Read-write	Name of the RNAT6 rule to which to bind NAT IPs. Minimum length = 1
ownergroup	<String>	Read-write	The owner node group in a Cluster for this rnat rule. Default value: DEFAULT_NG Minimum length = 1
td	<Double>	Read-only	Integer value that uniquely identifies the traffic domain in which you want to configure the entity. If you do not specify an ID, the entity becomes part of the default traffic domain, which has an ID of 0.

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
			Minimum value = 0 Maximum value = 4094
__count	<Double>	Read-only	count parameter

Operations

(click to see Properties)

ADD:| DELETE:| GET| GET (ALL)| COUNT

Some options that you can use for each operations:

- **Getting warnings in response:**NITRO allows you to get warnings in an operation by specifying the "warning" query parameter as "yes". For example, to get warnings while connecting to the Citrix ADC appliance, the URL is as follows:

  http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/login?warning=yes

  If any, the warnings are displayed in the response payload with the HTTP code "209 X-NITRO-WARNING".

- **Authenticated access for individual NITRO operations:**NITRO allows you to logon to the Citrix ADC appliance to perform individual operations. You can use this option instead of creating a NITRO session (using the login object) and then using that session to perform all operations,

  To do this, you must specify the username and password in the request header of the NITRO request as follows:

  X-NITRO-USER:<username>

  X-NITRO-PASS:<password>

  Note: In such cases, make sure that the request header DOES not include the following:

  Cookie:NITRO_AUTH_TOKEN=<tokenvalue>

`*Note:`*

Mandatory parameters are marked in redand placeholder content is marked in <green>.

add

URL:http://<Citrix-ADC-IP-address(NSIP)/nitro/v1/config/rnat6_nsip6_binding

HTTP Method:PUT

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Content-Type:application/json

Request Payload:

```
{

"rnat6_nsip6_binding":{

<b>"name":<String_value>,

</b><b>"natip6":<String_value>,

</b>"ownergroup":<String_value>

}}
```

Response:

HTTP Status Code on Success: 201 Created HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error

delete

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/rnat6_nsip6_binding/name_value<String>

Query-parameters:

args

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/rnat6_nsip6_binding/name_value<String>?**args=natip6:<String_value>,ownergroup:<String_value>**

HTTP Method:DELETE

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue>

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error

get

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/rnat6_nsip6_binding/name_value<String>

Query-parameters:

filter

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/rnat6_nsip6_binding/name_value<String>?**filter=property-name1:property-value1,property-name2:property-value2**

Use this query-parameter to get the filtered set of rnat6_nsip6_binding resources configured on Citrix ADC. Filtering can be done on any of the properties of the resource.

pagination

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/rnat6_nsip6_binding/name_value<String>?**pagesize=#no;pageno=#no**

Use this query-parameter to get the rnat6_nsip6_binding resources in chunks.

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "rnat6_nsip6_binding": [ {

"natip6":<String_value>,

"name":<String_value>,

"ownergroup":<String_value>,

"td":<Double_value>

}]}
```

get (all)

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/rnat6_nsip6_binding

Query-parameters:

bulkbindings

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/rnat6_nsip6_binding?**bulkbindings=yes**

NITRO allows you to fetch bindings in bulk.

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "rnat6_nsip6_binding": [ {

"natip6":<String_value>,

"name":<String_value>,

"ownergroup":<String_value>,

"td":<Double_value>

}]}
```

count

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/rnat6_nsip6_binding/name_value<String>?**count=yes**

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{"rnat6_nsip6_binding": [ { "__count": "#no"} ] }
```

rnatglobal_auditsyslogpolicy_binding

Binding object showing the auditsyslogpolicy that can be bound to rnatglobal.

Properties

(click to see Operations)

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
priority	<Double>	Read-write	The priority of the policy.
all	<Boolean>	Read-write	Remove all RNAT global config.
policy	<String>	Read-write	The policy Name.
__count	<Double>	Read-write	count parameter

Operations

(click to see Properties)

ADD:| DELETE:| GET| GET (ALL)| COUNT

Some options that you can use for each operations:

- **Getting warnings in response:**NITRO allows you to get warnings in an operation by specifying the "warning" query parameter as "yes". For example, to get warnings while connecting to the Citrix ADC appliance, the URL is as follows:

  http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/login?warning=yes

  If any, the warnings are displayed in the response payload with the HTTP code "209 X-NITRO-WARNING".

- **Authenticated access for individual NITRO operations:**NITRO allows you to logon to the Citrix ADC appliance to perform individual operations. You can use this option instead of creating a NITRO session (using the login object) and then using that session to perform all operations,

  To do this, you must specify the username and password in the request header of the NITRO request as follows:

X-NITRO-USER:<username>

X-NITRO-PASS:<password>

Note: In such cases, make sure that the request header DOES not include the following:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue>

`*Note:`*

Mandatory parameters are marked in redand placeholder content is marked in <green>.

add

URL:http://<Citrix-ADC-IP-address(NSIP)/nitro/v1/config/rnatglobal_auditsyslogpolicy_binding

HTTP Method:PUT

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Content-Type:application/json

Request Payload:

```
{
"rnatglobal_auditsyslogpolicy_binding":{
"policy":<String_value>,
"priority":<Double_value>
}}
```

Response:

HTTP Status Code on Success: 201 Created HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error

delete

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/rnatglobal_auditsyslogpolicy_binding

Query-parameters:

args

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/rnatglobal_auditsyslogpolicy_binding?**args=policy:<String_value>,all:<Boolean_value>**

HTTP Method:DELETE

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue>

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error

get

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/rnatglobal_auditsyslogpolicy_binding

Query-parameters:

filter

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/rnatglobal_auditsyslogpolicy_binding?**filter=property-name1:property-value1,property-name2:property-value2**

Use this query-parameter to get the filtered set of rnatglobal_auditsyslogpolicy_binding resources configured on Citrix ADC. Filtering can be done on any of the properties of the resource.

pagination

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/rnatglobal_auditsyslogpolicy_binding?**pagesize=#no;pageno=#no**

Use this query-parameter to get the rnatglobal_auditsyslogpolicy_binding resources in chunks.

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "rnatglobal_auditsyslogpolicy_binding": [ {

"priority":<Double_value>,

"all":<Boolean_value>,

"policy":<String_value>

}]}
```

get (all)

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/rnatglobal_auditsyslogpolicy_binding

Query-parameters:

bulkbindings

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/rnatglobal_auditsyslogpolicy_binding?**bulkbindings=yes**

NITRO allows you to fetch bindings in bulk.

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "rnatglobal_auditsyslogpolicy_binding": [ {

"priority":<Double_value>,

"all":<Boolean_value>,

"policy":<String_value>

}]}
```

count

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/rnatglobal_auditsyslogpolicy_binding?**count=yes**

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "rnatglobal_auditsyslogpolicy_binding": [ { "__count": "#no"} ] }
```

rnatglobal_binding

Binding object which returns the resources bound to rnatglobal.

Properties

(click to see Operations)

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
rnatglobal_auditsyslogpolicy_binding	<rnatglobal_auditsyslogpolicy_binding[]>	Read-only	auditsyslogpolicy that can be bound to rnatglobal.

Operations

(click to see Properties)

GET| GET (ALL)

Some options that you can use for each operations:

- **Getting warnings in response:**NITRO allows you to get warnings in an operation by specifying the "warning" query parameter as "yes". For example, to get warnings while connecting to the Citrix ADC appliance, the URL is as follows:

  http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/login?warning=yes

  If any, the warnings are displayed in the response payload with the HTTP code "209 X-NITRO-WARNING".

- **Authenticated access for individual NITRO operations:**NITRO allows you to logon to the Citrix ADC appliance to perform individual operations. You can use this option instead of creating a NITRO session (using the login object) and then using that session to perform all operations,

  To do this, you must specify the username and password in the request header of the NITRO request as follows:

X-NITRO-USER:<username>

X-NITRO-PASS:<password>

Note: In such cases, make sure that the request header DOES not include the following:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue>

`*Note:` *

Mandatory parameters are marked in redand placeholder content is marked in <green>.

get

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/rnatglobal_binding

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "rnatglobal_binding": [ {

"rnatglobal_auditsyslogpolicy_binding":<rnatglobal_auditsyslogpolicy_binding[]_value>

}]}
```

get (all)

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/rnatglobal_binding

Query-parameters:

bulkbindings

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/rnatglobal_binding?**bulkbindings=yes**

NITRO allows you to fetch bindings in bulk.

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "rnatglobal_binding": [ {

"rnatglobal_auditsyslogpolicy_binding":<rnatglobal_auditsyslogpolicy_binding[]_value>

}]}
```

rnatparam

Configuration for RNAT parameter resource.

Properties

(click to see Operations)

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
tcpproxy	<String>	Read-write	Enable TCP proxy, which enables the Citrix ADC to optimize the RNAT TCP traffic by using Layer 4 features. Default value: ENABLED Possible values = ENABLED, DISABLED
srcippersistency	<String>	Read-write	Enable source ip persistency, which enables the Citrix ADC to use the RNAT ips using source ip. Default value: DISABLED Possible values = ENABLED, DISABLED

Operations

(click to see Properties)

UPDATE| UNSET| GET (ALL)

Some options that you can use for each operations:

- **Getting warnings in response:**NITRO allows you to get warnings in an operation by specifying the "warning" query parameter as "yes". For example, to get warnings while connecting to the Citrix ADC appliance, the URL is as follows:

  http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/login?warning=yes

If any, the warnings are displayed in the response payload with the HTTP code "209 X-NITRO-WARNING".

- **Authenticated access for individual NITRO operations:**NITRO allows you to logon to the Citrix ADC appliance to perform individual operations. You can use this option instead of creating a NITRO session (using the login object) and then using that session to perform all operations,

  To do this, you must specify the username and password in the request header of the NITRO request as follows:

  X-NITRO-USER:<username>

  X-NITRO-PASS:<password>

  Note: In such cases, make sure that the request header DOES not include the following:

  Cookie:NITRO_AUTH_TOKEN=<tokenvalue>

`*Note:`*

Mandatory parameters are marked in redand placeholder content is marked in <green>.

update

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/rnatparam

HTTP Method:PUT

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Content-Type:application/json

Request Payload:

```
{"rnatparam":{

"tcpproxy":<String_value>,

"srcippersistency":<String_value>

}}
```

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error

unset

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/rnatparam?**action=unset**

HTTP Method:POST

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Content-Type:application/json

Request Payload:

```
{"rnatparam":{

"tcpproxy":true,

"srcippersistency":true

}}
```

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error

get (all)

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/rnatparam

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "rnatparam": [ {

"tcpproxy":<String_value>,

"srcippersistency":<String_value>

}]}
```

route

Configuration for route resource.

Properties

(click to see Operations)

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
network	<String>	Read-write	IPv4 network address for which to add a route entry in the routing table of the Citrix ADC.
netmask	<String>	Read-write	The subnet mask associated with the network address.

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
gateway	<String>	Read-write	IP address of the gateway for this route. Can be either the IP address of the gateway, or can be null to specify a null interface route. Minimum length = 1
vlan	<Double>	Read-write	VLAN as the gateway for this route. Minimum value = 1 Maximum value = 4094
cost	<Double>	Read-write	Positive integer used by the routing algorithms to determine preference for using this route. The lower the cost, the higher the preference. Minimum value = 0 Maximum value = 65535
td	<Double>	Read-write	Integer value that uniquely identifies the traffic domain in which you want to configure the entity. If you do not specify an ID, the entity becomes part of the default traffic

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
			domain, which has an ID of 0. Minimum value = 0 Maximum value = 4094
distance	<Double>	Read-write	Administrative distance of this route, which determines the preference of this route over other routes, with same destination, from different routing protocols. A lower value is preferred. Default value: 1 Minimum value = 0 Maximum value = 255
cost1	<Double>	Read-write	The cost of a route is used to compare routes of the same type. The route having the lowest cost is the most preferred route. Possible values: 0 through 65535. Default: 0. Minimum value = 0 Maximum value = 65535

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
weight	<Double>	Read-write	Positive integer used by the routing algorithms to determine preference for this route over others of equal cost. The lower the weight, the higher the preference. Default value: 1 Minimum value = 1 Maximum value = 65535
advertise	<String>	Read-write	Advertise this route. Possible values = DISABLED, ENABLED
protocol	<String[]>	Read-write	Routing protocol used for advertising this route. Default value: ADV_ROUTE_FLAGS Possible values = OSPF, ISIS, RIP, BGP
msr	<String>	Read-write	Monitor this route using a monitor of type ARP or PING. Default value: DISABLED

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
			Possible values = ENABLED, DISABLED
monitor	<String>	Read-write	Name of the monitor, of type ARP or PING, configured on the Citrix ADC to monitor this route. Minimum length = 1
ownergroup	<String>	Read-write	The owner node group in a Cluster for this route. If owner node group is not specified then the route is treated as Striped route. Default value: DEFAULT_NG Minimum length = 1
routetype	<String>	Read-write	Protocol used by routes that you want to remove from the routing table of the Citrix ADC. Possible values = CONNECTED, STATIC, DYNAMIC, OSPF, ISIS, RIP, BGP
detail	<Boolean>	Read-write	Display a detailed view.
gatewayname	<String>	Read-only	The name of the gateway for this route. For a route

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
			other than a link load balancing (LLB) route, this value is null.
type	<Boolean>	Read-only	State of the RNAT.
dynamic	<Boolean>	Read-only	State of the route.
Static	<Boolean>	Read-only	.
permanent	<Boolean>	Read-only	.
direct	<Boolean>	Read-only	.
nat	<Boolean>	Read-only	.
lbroute	<Boolean>	Read-only	.
adv	<Boolean>	Read-only	.
tunnel	<Boolean>	Read-only	Show whether it is a tunnel route or not.
data	<Boolean>	Read-only	Internal data of this route.
data0	<Boolean>	Read-only	Internal route type is stored, used for get.
flags	<Boolean>	Read-only	If this route is dynamic, the name of the routing protocol from which it was learned.
routeowners	<String[]>	Read-only	Use this option with -dynamic and in a cluster only to specify the set of nodes from which

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
			this dynamic route has been learnt. Possible values = 0, 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16, 17, 18, 19, 20, 21, 22, 23, 24, 25, 26, 27, 28, 29, 30, 31
retain	<Integer>	Read-only	.
ospf	<Boolean>	Read-only	OSPF protocol.
isis	<Boolean>	Read-only	ISIS protocol.
rip	<Boolean>	Read-only	RIP protocol.
bgp	<Boolean>	Read-only	BGP protocol.
dhcp	<Boolean>	Read-only	.
advospf	<Boolean>	Read-only	Advertised through OSPF protocol.
advisis	<Boolean>	Read-only	Advertised through ISIS protocol.
advrip	<Boolean>	Read-only	Advertised through RIP protocol.
advbgp	<Boolean>	Read-only	Advertised through BGP protocol.
state	<Double>	Read-only	The state of the static route. Possible values: UP, DOWN.
totalprobes	<Double>	Read-only	The total number of probes sent.

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
totalfailedprobes	<Double>	Read-only	The total number of failed probes.
failedprobes	<Double>	Read-only	Number of the current failed monitoring probes.
monstatcode	<Integer>	Read-only	The code indicating the monitor response.
monstatparam1	<Integer>	Read-only	First parameter used with the message code.
monstatparam2	<Integer>	Read-only	Second parameter used with the message code.
monstatparam3	<Integer>	Read-only	Third parameter used with the message code.
__count	<Double>	Read-only	count parameter

Operations

(click to see Properties)

ADD| CLEAR| DELETE| UPDATE| UNSET| GET (ALL)| COUNT

Some options that you can use for each operations:

- **Getting warnings in response:**NITRO allows you to get warnings in an operation by specifying the "warning" query parameter as "yes". For example, to get warnings while connecting to the Citrix ADC appliance, the URL is as follows:

  http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/login?warning=yes

  If any, the warnings are displayed in the response payload with the HTTP code "209 X-NITRO-WARNING".

- **Authenticated access for individual NITRO operations:**NITRO allows you to logon to the Citrix ADC appliance to perform individual operations. You can use this option instead of creating a NITRO session (using the login object) and then using that session to perform all operations,

  To do this, you must specify the username and password in the request header of the NITRO request as follows:

  X-NITRO-USER:<username>

  X-NITRO-PASS:<password>

  Note: In such cases, make sure that the request header DOES not include the following:

  Cookie:NITRO_AUTH_TOKEN=<tokenvalue>

`*Note:`*

Mandatory parameters are marked in redand placeholder content is marked in <green>.

add

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/route

HTTP Method:POST

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Content-Type:application/json

Request Payload:

```
{"route":{

<b>"network":<String_value>,

</b><b>"netmask":<String_value>,

</b>"gateway":<String_value>,

"vlan":<Double_value>,

"cost":<Double_value>,
```

```
"td":<Double_value>,

"distance":<Double_value>,

"cost1":<Double_value>,

"weight":<Double_value>,

"advertise":<String_value>,

"protocol":<String[]_value>,

"msr":<String_value>,

"monitor":<String_value>,

"ownergroup":<String_value>

}}
```

Response:

HTTP Status Code on Success: 201 Created HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error

clear

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/route?**action=clear**

HTTP Method:POST

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Content-Type:application/json

Request Payload:

```
{"route":{
```

```
<b>"routetype":<String_value>

</b>}}
```

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error

delete

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/route/network_value<String>

HTTP Method:DELETE

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue>

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error

update

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/route

HTTP Method:PUT

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Content-Type:application/json

Request Payload:

```
{"route":{

<b>"network":<String_value>,
```

```
</b><b>"netmask":<String_value>,

</b><b>"gateway":<String_value>,

</b>"td":<Double_value>,

"distance":<Double_value>,

"cost1":<Double_value>,

"weight":<Double_value>,

"advertise":<String_value>,

"protocol":<String[]_value>,

"msr":<String_value>,

"monitor":<String_value>

}}
```

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error

unset

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/route?**action=unset**

HTTP Method:POST

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Content-Type:application/json

Request Payload:

```
{"route":{

<b>"network":<String_value>,

</b><b>"netmask":<String_value>,

</b><b>"gateway":<String_value>,

</b>"td":<Double_value>,

"advertise":true,

"distance":true,

"cost1":true,

"weight":true,

"protocol":true,

"msr":true,

"monitor":true

}}
```

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error

get (all)

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/route

Query-parameters:

args

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/route?**args=network:<String_value>,netmask:<String_value>,gateway:<String_value>,vlan:<Double_value>,td**

Use this query-parameter to get route resources based on additional properties.

attrs

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/route?**attrs=property-name1,property-name2**

Use this query parameter to specify the resource details that you want to retrieve.

filter

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/route?**filter=property-name1:property-val1,property-name2:property-val2**

Use this query-parameter to get the filtered set of route resources configured on Citrix ADC. Filtering can be done on any of the properties of the resource.

view

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/route?**view=summary**

Note: By default, the retrieved results are displayed in detail view (?view=detail).

pagination

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/route?**pagesize=#no;pageno=#no**

Use this query-parameter to get the route resources in chunks.

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "route": [ {

network:<String_value>,netmask:<String_value>,gateway:<String_value>,vlan:
<Double_value>,td:<Double_value>,routetype:<String_value>,detail:<Boolean_
value>"gatewayname":<String_value>,

"advertise":<String_value>,

"type":<Boolean_value>,

"dynamic":<Boolean_value>,

"Static":<Boolean_value>,

"permanent":<Boolean_value>,

"direct":<Boolean_value>,

"nat":<Boolean_value>,

"lbroute":<Boolean_value>,

"adv":<Boolean_value>,

"tunnel":<Boolean_value>,

"cost":<Double_value>,

"distance":<Double_value>,

"cost1":<Double_value>,

"weight":<Double_value>,

"protocol":<String[]_value>,

"data":<Boolean_value>,

"data0":<Boolean_value>,
```

```
"flags":<Boolean_value>,

"routeowners":<String[]_value>,

"retain":<Integer_value>,

"ospf":<Boolean_value>,

"isis":<Boolean_value>,

"rip":<Boolean_value>,

"bgp":<Boolean_value>,

"dhcp":<Boolean_value>,

"advospf":<Boolean_value>,

"advisis":<Boolean_value>,

"advrip":<Boolean_value>,

"advbgp":<Boolean_value>,

"msr":<String_value>,

"monitor":<String_value>,

"state":<Double_value>,

"totalprobes":<Double_value>,

"totalfailedprobes":<Double_value>,

"failedprobes":<Double_value>,

"monstatcode":<Integer_value>,

"monstatparam1":<Integer_value>,

"monstatparam2":<Integer_value>,

"monstatparam3":<Integer_value>,
```

```
"ownergroup":<String_value>

}]}
```

count

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/route?**count=yes**

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "route": [ { "__count": "#no"} ] }
```

route6

Configuration for route 6 resource.

Properties

(click to see Operations)

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
network	<String>	Read-write	IPv6 network address for which to add a route entry to

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
			the routing table of the Citrix ADC.
gateway	<String>	Read-write	The gateway for this route. The value for this parameter is either an IPv6 address or null. Default value: 0
vlan	<Double>	Read-write	Integer value that uniquely identifies a VLAN through which the Citrix ADC forwards the packets for this route. Default value: 0 Minimum value = 0 Maximum value = 4094
vxlan	<Double>	Read-write	Integer value that uniquely identifies a VXLAN through which the Citrix ADC forwards the packets for this route. Minimum value = 1 Maximum value = 16777215
weight	<Double>	Read-write	Positive integer used by the routing algorithms to determine preference for this route over others of equal cost. The

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
			lower the weight, the higher the preference. Default value: 1 Minimum value = 1 Maximum value = 65535
distance	<Double>	Read-write	Administrative distance of this route from the appliance. Default value: 1 Minimum value = 1 Maximum value = 254
cost	<Double>	Read-write	Positive integer used by the routing algorithms to determine preference for this route. The lower the cost, the higher the preference. Default value: 1 Minimum value = 0 Maximum value = 65535
advertise	<String>	Read-write	Advertise this route. Possible values = DISABLED, ENABLED

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
msr	<String>	Read-write	Monitor this route with a monitor of type ND6 or PING. Default value: DISABLED Possible values = ENABLED, DISABLED
monitor	<String>	Read-write	Name of the monitor, of type ND6 or PING, configured on the Citrix ADC to monitor this route. Minimum length = 1
td	<Double>	Read-write	Integer value that uniquely identifies the traffic domain in which you want to configure the entity. If you do not specify an ID, the entity becomes part of the default traffic domain, which has an ID of 0. Minimum value = 0 Maximum value = 4094
ownergroup	<String>	Read-write	The owner node group in a Cluster for this route6. If owner node group is not specified then the route is treated as Striped route.

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
			Default value: DEFAULT_NG Minimum length = 1
routetype	<String>	Read-write	Type of IPv6 routes to remove from the routing table of the Citrix ADC. Possible values = CONNECTED, STATIC, DYNAMIC, OSPF, ISIS, BGP, RIP, ND-RA-ROUTE, FIB6
detail	<Boolean>	Read-write	To get a detailed view.
gatewayname	<String>	Read-only	The name of the gateway for this route.
type	<Boolean>	Read-only	State of the RNAT.
dynamic	<Boolean>	Read-only	Whether this route is dynamically learned or not.
data	<Boolean>	Read-only	Internal data of this route.
flags	<Boolean>	Read-only	For a dynamic route, the routing protocol from which the route was learned.
state	<Double>	Read-only	Whether this route is UP or DOWN.

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
totalprobes	<Double>	Read-only	The total number of probes sent.
totalfailedprobes	<Double>	Read-only	The total number of failed probes.
failedprobes	<Double>	Read-only	Current number of failed monitoring probes.
monstatcode	<Integer>	Read-only	The code indicating the monitor response.
monstatparam1	<Integer>	Read-only	First parameter for use with message code.
monstatparam2	<Integer>	Read-only	Second parameter for use with message code.
monstatparam3	<Integer>	Read-only	Third parameter for use with message code.
data1	<String>	Read-only	Internal data of this route. Possible values = ENABLED, DISABLED
routeowners	<String[]>	Read-only	Use this option with -dynamic and in a cluster only to specify the set of nodes from which this dynamic route has been learnt.

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
			Possible values = 0, 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16, 17, 18, 19, 20, 21, 22, 23, 24, 25, 26, 27, 28, 29, 30, 31
retain	<Integer>	Read-only	.
Static	<Boolean>	Read-only	Static route.
permanent	<Boolean>	Read-only	Permanent Route.
connected	<Boolean>	Read-only	Connected Route.
ospfv3	<Boolean>	Read-only	For a dynamic route, the routing protocol from which the route was learned.
isis	<Boolean>	Read-only	If this route is dynamic then which routing protocol was it learnt from.
active	<Boolean>	Read-only	For a dynamic route, the routing protocol from which the route was learned.
bgp	<Boolean>	Read-only	For a dynamic route, the routing protocol from which the route was learned.
rip	<Boolean>	Read-only	For a dynamic route, the routing protocol from which the route was learned.

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
raroute	<Boolean>	Read-only	For a dynamic route, the routing protocol from which the route was learned.
__count	<Double>	Read-only	count parameter

Operations

(click to see Properties)

ADD| CLEAR| DELETE| UPDATE| UNSET| GET (ALL)| COUNT

Some options that you can use for each operations:

- **Getting warnings in response:**NITRO allows you to get warnings in an operation by specifying the "warning" query parameter as "yes". For example, to get warnings while connecting to the Citrix ADC appliance, the URL is as follows:

  http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/login?warning=yes

  If any, the warnings are displayed in the response payload with the HTTP code "209 X-NITRO-WARNING".

- **Authenticated access for individual NITRO operations:**NITRO allows you to logon to the Citrix ADC appliance to perform individual operations. You can use this option instead of creating a NITRO session (using the login object) and then using that session to perform all operations,

  To do this, you must specify the username and password in the request header of the NITRO request as follows:

  X-NITRO-USER:<username>

  X-NITRO-PASS:<password>

  Note: In such cases, make sure that the request header DOES not include the following:

  Cookie:NITRO_AUTH_TOKEN=<tokenvalue>

`*Note:`*

Mandatory parameters are marked in redand placeholder content is marked in <green>.

add

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/route6

HTTP Method:POST

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Content-Type:application/json

Request Payload:

```
{"route6":{

<b>"network":<String_value>,

</b>"gateway":<String_value>,

"vlan":<Double_value>,

"vxlan":<Double_value>,

"weight":<Double_value>,

"distance":<Double_value>,

"cost":<Double_value>,

"advertise":<String_value>,

"msr":<String_value>,

"monitor":<String_value>,

"td":<Double_value>,

"ownergroup":<String_value>

}}
```

Response:

HTTP Status Code on Success: 201 Created HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error

clear

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/route6?**action=clear**

HTTP Method:POST

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Content-Type:application/json

Request Payload:

```
{"route6":{

<b>"routetype":<String_value>

</b>}}
```

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error

delete

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/route6/network_value<String>

HTTP Method:DELETE

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue>

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error

update

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/route6

HTTP Method:PUT

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Content-Type:application/json

Request Payload:

```
{"route6":{

<b>"network":<String_value>,

</b>"gateway":<String_value>,

"vlan":<Double_value>,

"vxlan":<Double_value>,

"weight":<Double_value>,

"distance":<Double_value>,

"cost":<Double_value>,

"advertise":<String_value>,

"msr":<String_value>,

"monitor":<String_value>,

"td":<Double_value>

}}
```

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error

unset

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/route6?**action=unset**

HTTP Method:POST

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Content-Type:application/json

Request Payload:

```
{"route6":{

<b>"network":<String_value>,

</b>"gateway":<String_value>,

"vlan":<Double_value>,

"vxlan":<Double_value>,

"td":<Double_value>,

"weight":true,

"distance":true,

"cost":true,

"advertise":true,

"msr":true,
```

```
"monitor":true

}}
```

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error

get (all)

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/route6

Query-parameters:

args

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/route6?**args=network:<String_value>,gateway:<String_value>,vlan:<Double_value>,vxlan:<Double_value>,td:<**

Use this query-parameter to get route6 resources based on additional properties.

attrs

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/route6?**attrs=property-name1,property-name2**

Use this query parameter to specify the resource details that you want to retrieve.

filter

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/route6?**filter=property-name1:property-val1,property-name2:property-val2**

Use this query-parameter to get the filtered set of route6 resources configured on Citrix ADC. Filtering can be done on any of the properties of the resource.

view

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/route6?**view=summary**

Note: By default, the retrieved results are displayed in detail view (?view=detail).

pagination

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/route6?**pagesize=#no;pageno=#no**

Use this query-parameter to get the route6 resources in chunks.

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "route6": [ {

network:<String_value>,gateway:<String_value>,vlan:<Double_value>,vxlan:<D
ouble_value>,td:<Double_value>,routetype:<String_value>,detail:<Boolean_va
lue>"gatewayname":<String_value>,

"advertise":<String_value>,

"type":<Boolean_value>,

"dynamic":<Boolean_value>,

"weight":<Double_value>,

"distance":<Double_value>,

"cost":<Double_value>,

"data":<Boolean_value>,
```

```
"flags":<Boolean_value>,

"msr":<String_value>,

"monitor":<String_value>,

"state":<Double_value>,

"totalprobes":<Double_value>,

"totalfailedprobes":<Double_value>,

"failedprobes":<Double_value>,

"monstatcode":<Integer_value>,

"monstatparam1":<Integer_value>,

"monstatparam2":<Integer_value>,

"monstatparam3":<Integer_value>,

"data1":<String_value>,

"routeowners":<String[]_value>,

"retain":<Integer_value>,

"Static":<Boolean_value>,

"permanent":<Boolean_value>,

"connected":<Boolean_value>,

"ospfv3":<Boolean_value>,

"isis":<Boolean_value>,

"active":<Boolean_value>,

"bgp":<Boolean_value>,

"rip":<Boolean_value>,
```

```
"raroute":<Boolean_value>,

"ownergroup":<String_value>

}]}
```

count

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/route6?**count=yes**

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "route6": [ { "__count": "#no"} ] }
```

rsskeytype

Configuration for RSS key type resource.

Properties

(click to see Operations)

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
rsstype	<String>	Read-write	Type of RSS key, possible values are SYMMETRIC and ASYMMETRIC. Default value: ASYMMETRIC Possible values = ASYMMETRIC, SYMMETRIC

Operations

(click to see Properties)

UPDATE| GET (ALL)

Some options that you can use for each operations:

- **Getting warnings in response:**NITRO allows you to get warnings in an operation by specifying the "warning" query parameter as "yes". For example, to get warnings while connecting to the Citrix ADC appliance, the URL is as follows:

  http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/login?warning=yes

  If any, the warnings are displayed in the response payload with the HTTP code "209 X-NITRO-WARNING".

- **Authenticated access for individual NITRO operations:**NITRO allows you to logon to the Citrix ADC appliance to perform individual operations. You can use this option instead of creating a NITRO session (using the login object) and then using that session to perform all operations,

  To do this, you must specify the username and password in the request header of the NITRO request as follows:

  X-NITRO-USER:<username>

  X-NITRO-PASS:<password>

  Note: In such cases, make sure that the request header DOES not include the following:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue>

`*Note:` *

Mandatory parameters are marked in redand placeholder content is marked in <green>.

update

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/rsskeytype

HTTP Method:PUT

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Content-Type:application/json

Request Payload:

```
{"rsskeytype":{

<b>"rsstype":<String_value>

</b>}}
```

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error

get (all)

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/rsskeytype

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "rsskeytype": [ {

"rsstype":<String_value>

}]}
```

vlan

Configuration for "VLAN" resource.

Properties

(click to see Operations)

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
id	<Double>	Read-write	A positive integer that uniquely identifies a VLAN. Minimum value = 1 Maximum value = 4094
aliasname	<String>	Read-write	A name for the VLAN. Must begin with a letter, a number, or the underscore symbol, and can consist of from 1 to 31 letters, numbers, and the

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
			hyphen (-), period (.) pound (#), space ( ), at sign (@), equals (=), colon (:), and underscore (_) characters. You should choose a name that helps identify the VLAN. However, you cannot perform any VLAN operation by specifying this name instead of the VLAN ID. Maximum length = 31
dynamicrouting	<String>	Read-write	Enable dynamic routing on this VLAN. Default value: DISABLED Possible values = ENABLED, DISABLED
ipv6dynamicrouting	<String>	Read-write	Enable all IPv6 dynamic routing protocols on this VLAN. Note: For the ENABLED setting to work, you must configure IPv6 dynamic routing protocols from the VTYSH command line.

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
			Default value: DISABLED Possible values = ENABLED, DISABLED
mtu	<Double>	Read-write	Specifies the maximum transmission unit (MTU), in bytes. The MTU is the largest packet size, excluding 14 bytes of ethernet header and 4 bytes of crc, that can be transmitted and received over this VLAN. Default value: 0 Minimum value = 500 Maximum value = 9216
sharing	<String>	Read-write	If sharing is enabled, then this vlan can be shared across multiple partitions by binding it to all those partitions. If sharing is disabled, then this vlan can be bound to only one of the partitions. Default value: DISABLED

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
			Possible values = ENABLED, DISABLED
linklocalipv6addr	<String>	Read-only	The link-local IP address assigned to the VLAN.
rnat	<Boolean>	Read-only	Temporary flag used for internal purpose.
portbitmap	<Double>	Read-only	Member interfaces of this vlan.
lsbitmap	<Double>	Read-only	Member linksets of this vlan.
tagbitmap	<Double>	Read-only	Tagged members of this vlan.
lstagbitmap	<Double>	Read-only	Tagged linksets of this vlan.
ifaces	<String>	Read-only	Names of all member interfaces of this vlan.
tagifaces	<String>	Read-only	Names of all tagged member interfaces of this vlan.
ifnum	<String>	Read-only	The interface to be bound to the VLAN, specified in slot/port notation (for example, 1/3). Minimum length = 1
tagged	<Boolean>	Read-only	Make the interface an 802.1q tagged interface. Packets sent on this interface

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
			on this VLAN have an additional 4-byte 802.1q tag, which identifies the VLAN. To use 802.1q tagging, you must also configure the switch connected to the appliance's interfaces.
vlantd	<Double>	Read-only	Traffic domain associated with vlan. Minimum value = 0 Maximum value = 4094
sdxvlan	<String>	Read-only	SDX vlan. Default value: NO Possible values = YES, NO
partitionname	<String>	Read-only	Name of the Partition to which this vlan bound to. Minimum length = 1
vxlan	<Double>	Read-only	The VXLAN that extends this vlan.
__count	<Double>	Read-only	count parameter

Operations

(click to see Properties)

ADD| DELETE| UPDATE| UNSET| GET (ALL)| GET| COUNT

Some options that you can use for each operations:

- **Getting warnings in response:**NITRO allows you to get warnings in an operation by specifying the "warning" query parameter as "yes". For example, to get warnings while connecting to the Citrix ADC appliance, the URL is as follows:

  http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/login?warning=yes

  If any, the warnings are displayed in the response payload with the HTTP code "209 X-NITRO-WARNING".

- **Authenticated access for individual NITRO operations:**NITRO allows you to logon to the Citrix ADC appliance to perform individual operations. You can use this option instead of creating a NITRO session (using the login object) and then using that session to perform all operations,

  To do this, you must specify the username and password in the request header of the NITRO request as follows:

  X-NITRO-USER:<username>

  X-NITRO-PASS:<password>

  Note: In such cases, make sure that the request header DOES not include the following:

  Cookie:NITRO_AUTH_TOKEN=<tokenvalue>

`*Note:`*

Mandatory parameters are marked in redand placeholder content is marked in <green>.

add

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/vlan

HTTP Method:POST

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Content-Type:application/json

Request Payload:

```
{"vlan":{
```

```
<b>"id":<Double_value>,

</b>"aliasname":<String_value>,

"dynamicrouting":<String_value>,

"ipv6dynamicrouting":<String_value>,

"mtu":<Double_value>,

"sharing":<String_value>

}}
```

Response:

HTTP Status Code on Success: 201 Created HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error

delete

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/vlan/id_value<Double>

HTTP Method:DELETE

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue>

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error

update

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/vlan

HTTP Method:PUT

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Content-Type:application/json

Request Payload:

```
{"vlan":{

<b>"id":<Double_value>,

</b>"aliasname":<String_value>,

"dynamicrouting":<String_value>,

"ipv6dynamicrouting":<String_value>,

"mtu":<Double_value>,

"sharing":<String_value>

}}
```

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error

unset

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/vlan?**action=unset**

HTTP Method:POST

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Content-Type:application/json

Request Payload:

```
{"vlan":{
```

```
<b>"id":<Double_value>,

</b>"aliasname":true,

"dynamicrouting":true,

"ipv6dynamicrouting":true,

"mtu":true,

"sharing":true

}}
```

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error

get (all)

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/vlan

Query-parameters:

attrs

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/vlan?**attrs=property-name1,property-name2**

Use this query parameter to specify the resource details that you want to retrieve.

filter

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/vlan?**filter=property-name1:property-val1,property-name2:property-val2**

Use this query-parameter to get the filtered set of vlan resources configured on Citrix ADC. Filtering can be done on any of the properties of the resource.

view

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/vlan?**view=summary**

Note: By default, the retrieved results are displayed in detail view (?view=detail).

pagination

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/vlan?**pagesize=#no;pageno=#no**

Use this query-parameter to get the vlan resources in chunks.

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "vlan": [ {

"id":<Double_value>,

"aliasname":<String_value>,

"linklocalipv6addr":<String_value>,

"rnat":<Boolean_value>,

"portbitmap":<Double_value>,

"lsbitmap":<Double_value>,

"tagbitmap":<Double_value>,

"lstagbitmap":<Double_value>,
```

```
"ifaces":<String_value>,

"tagifaces":<String_value>,

"dynamicrouting":<String_value>,

"ipv6dynamicrouting":<String_value>,

"ifnum":<String_value>,

"tagged":<Boolean_value>,

"vlantd":<Double_value>,

"sdxvlan":<String_value>,

"mtu":<Double_value>,

"sharing":<String_value>,

"partitionname":<String_value>,

"vxlan":<Double_value>

}]}
```

get

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/vlan/id_value<Double>

Query-parameters:

attrs

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/vlan/id_value<Double>?**attrs=property-name1,property-name2**

Use this query parameter to specify the resource details that you want to retrieve.

view

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/vlan/id_value<Double>?**view=summary**

Note: By default, the retrieved results are displayed in detail view (?view=detail).

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "vlan": [ {

"id":<Double_value>,

"aliasname":<String_value>,

"linklocalipv6addr":<String_value>,

"rnat":<Boolean_value>,

"portbitmap":<Double_value>,

"lsbitmap":<Double_value>,

"tagbitmap":<Double_value>,

"lstagbitmap":<Double_value>,

"ifaces":<String_value>,

"tagifaces":<String_value>,

"dynamicrouting":<String_value>,
```

```
"ipv6dynamicrouting":<String_value>,

"ifnum":<String_value>,

"tagged":<Boolean_value>,

"vlantd":<Double_value>,

"sdxvlan":<String_value>,

"mtu":<Double_value>,

"sharing":<String_value>,

"partitionname":<String_value>,

"vxlan":<Double_value>

}]}
```

count

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/vlan?**count=yes**

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "vlan": [ { "__count": "#no"} ] }
```

vlan_binding

Binding object which returns the resources bound to vlan.

Properties

(click to see Operations)

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
id	<Double>	Read-write	Integer that uniquely identifies the VLAN for which the details are to be displayed. Minimum value = 1 Maximum value = 4094
vlan_channel_binding	<vlan_channel_binding[]>	Read-only	channel that can be bound to vlan.
vlan_nsip_binding	<vlan_nsip_binding[]>	Read-only	nsip that can be bound to vlan.
vlan_interface_binding	<vlan_interface_binding[]>	Read-only	interface that can be bound to vlan.
vlan_linkset_binding	<vlan_linkset_binding[]>	Read-only	linkset that can be bound to vlan.
vlan_nsip6_binding	<vlan_nsip6_binding[]>	Read-only	nsip6 that can be bound to vlan.

Operations

(click to see Properties)

GET| GET (ALL)

Some options that you can use for each operations:

- **Getting warnings in response:**NITRO allows you to get warnings in an operation by specifying the "warning" query parameter as "yes". For example, to get warnings while connecting to the Citrix ADC appliance, the URL is as follows:

  http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/login?warning=yes

  If any, the warnings are displayed in the response payload with the HTTP code "209 X-NITRO-WARNING".

- **Authenticated access for individual NITRO operations:**NITRO allows you to logon to the Citrix ADC appliance to perform individual operations. You can use this option instead of creating a NITRO session (using the login object) and then using that session to perform all operations,

  To do this, you must specify the username and password in the request header of the NITRO request as follows:

  X-NITRO-USER:<username>

  X-NITRO-PASS:<password>

  Note: In such cases, make sure that the request header DOES not include the following:

  Cookie:NITRO_AUTH_TOKEN=<tokenvalue>

`*Note:`*

Mandatory parameters are marked in redand placeholder content is marked in <green>.

get

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/vlan_binding/id_value<Double>

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "vlan_binding": [ {

"id":<Double_value>,

"vlan_interface_binding":<vlan_interface_binding[]_value>,

"vlan_nsip6_binding":<vlan_nsip6_binding[]_value>,

"vlan_nsip_binding":<vlan_nsip_binding[]_value>,

"vlan_linkset_binding":<vlan_linkset_binding[]_value>,

"vlan_channel_binding":<vlan_channel_binding[]_value>

}]}
```

get (all)

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/vlan_binding

Query-parameters:

bulkbindings

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/vlan_binding?**bulkbindings=yes**

NITRO allows you to fetch bindings in bulk.

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "vlan_binding": [ {

"id":<Double_value>,

"vlan_interface_binding":<vlan_interface_binding[]_value>,

"vlan_nsip6_binding":<vlan_nsip6_binding[]_value>,

"vlan_nsip_binding":<vlan_nsip_binding[]_value>,

"vlan_linkset_binding":<vlan_linkset_binding[]_value>,

"vlan_channel_binding":<vlan_channel_binding[]_value>

}]}
```

vlan_channel_binding

Binding object showing the channel that can be bound to vlan.

Properties

(click to see Operations)

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
ownergroup	<String>	Read-write	The owner node group in a Cluster for this vlan.

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
			Default value: DEFAULT_NG Minimum length = 1
id	<Double>	Read-write	Specifies the virtual LAN ID. Minimum value = 1 Maximum value = 4094
ifnum	<String>	Read-write	The interface to be bound to the VLAN, specified in slot/port notation (for example, 1/3). Minimum length = 1
tagged	<Boolean>	Read-write	Make the interface an 802.1q tagged interface. Packets sent on this interface on this VLAN have an additional 4-byte 802.1q tag, which identifies the VLAN. To use 802.1q tagging, you must also configure the switch connected to the appliance's interfaces.
__count	<Double>	Read-write	count parameter

Operations

(click to see Properties)

ADD:| DELETE:| GET| GET (ALL)| COUNT

Some options that you can use for each operations:

- **Getting warnings in response:**NITRO allows you to get warnings in an operation by specifying the "warning" query parameter as "yes". For example, to get warnings while connecting to the Citrix ADC appliance, the URL is as follows:

  http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/login?warning=yes

  If any, the warnings are displayed in the response payload with the HTTP code "209 X-NITRO-WARNING".

- **Authenticated access for individual NITRO operations:**NITRO allows you to logon to the Citrix ADC appliance to perform individual operations. You can use this option instead of creating a NITRO session (using the login object) and then using that session to perform all operations,

  To do this, you must specify the username and password in the request header of the NITRO request as follows:

  X-NITRO-USER:<username>

  X-NITRO-PASS:<password>

  Note: In such cases, make sure that the request header DOES not include the following:

  Cookie:NITRO_AUTH_TOKEN=<tokenvalue>

`*Note:`*

Mandatory parameters are marked in redand placeholder content is marked in <green>.

add

URL:http://<Citrix-ADC-IP-address(NSIP)/nitro/v1/config/vlan_channel_binding

HTTP Method:PUT

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Content-Type:application/json

Request Payload:

```
{

"vlan_channel_binding":{

<b>"id":<Double_value>,

</b>"ifnum":<String_value>,

"tagged":<Boolean_value>,

"ownergroup":<String_value>

}}
```

Response:

HTTP Status Code on Success: 201 Created HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error

delete

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/vlan_channel_binding/id_value<Double>

Query-parameters:

args

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/vlan_channel_binding/id_value<Double>?**args=ifnum:<String_value>,tagged:<Boolean_value>,ownergroup:<String_value>**

HTTP Method:DELETE

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue>

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error

get

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/vlan_channel_binding/id_value<Double>

Query-parameters:

filter

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/vlan_channel_binding/id_value<Double>?**filter=property-name1:property-value1,property-name2:property-value2**

Use this query-parameter to get the filtered set of vlan_channel_binding resources configured on Citrix ADC. Filtering can be done on any of the properties of the resource.

pagination

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/vlan_channel_binding/id_value<Double>?**pagesize=#no;pageno=#no**

Use this query-parameter to get the vlan_channel_binding resources in chunks.

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "vlan_channel_binding": [ {
```

```
"ownergroup":<String_value>,

"id":<Double_value>,

"ifnum":<String_value>,

"tagged":<Boolean_value>

}]}
```

get (all)

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/vlan_channel_binding

Query-parameters:

bulkbindings

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/vlan_channel_binding?**bulkbindings=yes**

NITRO allows you to fetch bindings in bulk.

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "vlan_channel_binding": [ {

"ownergroup":<String_value>,
```

```
"id":<Double_value>,

"ifnum":<String_value>,

"tagged":<Boolean_value>

}]}
```

count

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/vlan_channel_binding/id_value<Double>?**count=yes**

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{"vlan_channel_binding": [ { "__count": "#no"} ] }
```

vlan_interface_binding

Binding object showing the interface that can be bound to vlan.

Properties

(click to see Operations)

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
ownergroup	<String>	Read-write	The owner node group in a Cluster for this vlan. Default value: DEFAULT_NG Minimum length = 1
id	<Double>	Read-write	Specifies the virtual LAN ID. Minimum value = 1 Maximum value = 4094
ifnum	<String>	Read-write	The interface to be bound to the VLAN, specified in slot/port notation (for example, 1/3). Minimum length = 1
tagged	<Boolean>	Read-write	Make the interface an 802.1q tagged interface. Packets sent on this interface on this VLAN have an additional 4-byte 802.1q tag, which identifies the VLAN. To use 802.1q tagging, you must also configure the switch connected to the appliance's interfaces.
__count	<Double>	Read-write	count parameter

Operations

(click to see Properties)

ADD:| DELETE:| GET| GET (ALL)| COUNT

Some options that you can use for each operations:

- **Getting warnings in response:**NITRO allows you to get warnings in an operation by specifying the "warning" query parameter as "yes". For example, to get warnings while connecting to the Citrix ADC appliance, the URL is as follows:

  http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/login?warning=yes

  If any, the warnings are displayed in the response payload with the HTTP code "209 X-NITRO-WARNING".

- **Authenticated access for individual NITRO operations:**NITRO allows you to logon to the Citrix ADC appliance to perform individual operations. You can use this option instead of creating a NITRO session (using the login object) and then using that session to perform all operations,

  To do this, you must specify the username and password in the request header of the NITRO request as follows:

  X-NITRO-USER:<username>

  X-NITRO-PASS:<password>

  Note: In such cases, make sure that the request header DOES not include the following:

  Cookie:NITRO_AUTH_TOKEN=<tokenvalue>

`*Note:`*

Mandatory parameters are marked in redand placeholder content is marked in <green>.

add

URL:http://<Citrix-ADC-IP-address(NSIP)/nitro/v1/config/vlan_interface_binding

HTTP Method:PUT

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Content-Type:application/json

Request Payload:

```
{

"vlan_interface_binding":{

<b>"id":<Double_value>,

</b>"ifnum":<String_value>,

"tagged":<Boolean_value>,

"ownergroup":<String_value>

}}
```

Response:

HTTP Status Code on Success: 201 Created HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error

delete

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/vlan_interface_binding/id_value<Double>

Query-parameters:

args

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/vlan_interface_binding/id_value<Double>?**args=ifnum:<String_value>,tagged:<Boolean_value>,ownergroup:<String_value>**

HTTP Method:DELETE

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue>

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error

get

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/vlan_interface_binding/id_value<Double>

Query-parameters:

filter

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/vlan_interface_binding/id_value<Double>?**filter=property-name1:property-value1,property-name2:property-value2**

Use this query-parameter to get the filtered set of vlan_interface_binding resources configured on Citrix ADC. Filtering can be done on any of the properties of the resource.

pagination

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/vlan_interface_binding/id_value<Double>?**pagesize=#no;pageno=#no**

Use this query-parameter to get the vlan_interface_binding resources in chunks.

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "vlan_interface_binding": [ {
```

```
"ownergroup":<String_value>,

"id":<Double_value>,

"ifnum":<String_value>,

"tagged":<Boolean_value>

}]}
```

get (all)

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/vlan_interface_binding

Query-parameters:

bulkbindings

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/vlan_interface_binding?**bulkbindings=yes**

NITRO allows you to fetch bindings in bulk.

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "vlan_interface_binding": [ {

"ownergroup":<String_value>,
```

```
"id":<Double_value>,

"ifnum":<String_value>,

"tagged":<Boolean_value>

}]}
```

count

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/vlan_interface_binding/id_value<Double>?**count=yes**

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{"vlan_interface_binding": [ { "__count": "#no"} ] }
```

vlan_linkset_binding

Binding object showing the linkset that can be bound to vlan.

Properties

(click to see Operations)

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
ownergroup	<String>	Read-write	The owner node group in a Cluster for this vlan. Default value: DEFAULT_NG Minimum length = 1
id	<Double>	Read-write	Specifies the virtual LAN ID. Minimum value = 1 Maximum value = 4094
ifnum	<String>	Read-write	The interface to be bound to the VLAN, specified in slot/port notation (for example, 1/3). Minimum length = 1
tagged	<Boolean>	Read-write	Make the interface an 802.1q tagged interface. Packets sent on this interface on this VLAN have an additional 4-byte 802.1q tag, which identifies the VLAN. To use 802.1q tagging, you must also configure the switch connected to the appliance's interfaces.
__count	<Double>	Read-write	count parameter

Operations

(click to see Properties)

ADD:| DELETE:| GET| GET (ALL)| COUNT

Some options that you can use for each operations:

- **Getting warnings in response:**NITRO allows you to get warnings in an operation by specifying the "warning" query parameter as "yes". For example, to get warnings while connecting to the Citrix ADC appliance, the URL is as follows:

  http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/login?warning=yes

  If any, the warnings are displayed in the response payload with the HTTP code "209 X-NITRO-WARNING".

- **Authenticated access for individual NITRO operations:**NITRO allows you to logon to the Citrix ADC appliance to perform individual operations. You can use this option instead of creating a NITRO session (using the login object) and then using that session to perform all operations,

  To do this, you must specify the username and password in the request header of the NITRO request as follows:

  X-NITRO-USER:<username>

  X-NITRO-PASS:<password>

  Note: In such cases, make sure that the request header DOES not include the following:

  Cookie:NITRO_AUTH_TOKEN=<tokenvalue>

`*Note:`*

Mandatory parameters are marked in redand placeholder content is marked in <green>.

add

URL:http://<Citrix-ADC-IP-address(NSIP)/nitro/v1/config/vlan_linkset_binding

HTTP Method:PUT

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Content-Type:application/json

Request Payload:

```
{

"vlan_linkset_binding":{

<b>"id":<Double_value>,

</b>"ifnum":<String_value>,

"tagged":<Boolean_value>,

"ownergroup":<String_value>

}}
```

Response:

HTTP Status Code on Success: 201 Created HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error

delete

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/vlan_linkset_binding/id_value<Double>

Query-parameters:

args

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/vlan_linkset_binding/id_value<Double>?**args=ifnum:<String_value>,tagged:<Boolean_value>,ownergroup:<String_value>**

HTTP Method:DELETE

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue>

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error

get

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/vlan_linkset_binding/id_value<Double>

Query-parameters:

filter

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/vlan_linkset_binding/id_value<Double>?**filter=property-name1:property-value1,property-name2:property-value2**

Use this query-parameter to get the filtered set of vlan_linkset_binding resources configured on Citrix ADC. Filtering can be done on any of the properties of the resource.

pagination

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/vlan_linkset_binding/id_value<Double>?**pagesize=#no;pageno=#no**

Use this query-parameter to get the vlan_linkset_binding resources in chunks.

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "vlan_linkset_binding": [ {
```

```
"ownergroup":<String_value>,

"id":<Double_value>,

"ifnum":<String_value>,

"tagged":<Boolean_value>

}]}
```

get (all)

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/vlan_linkset_binding

Query-parameters:

bulkbindings

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/vlan_linkset_binding?**bulkbindings=yes**

NITRO allows you to fetch bindings in bulk.

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "vlan_linkset_binding": [ {

"ownergroup":<String_value>,
```

```
"id":<Double_value>,

"ifnum":<String_value>,

"tagged":<Boolean_value>

}]}
```

count

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/vlan_linkset_binding/id_value<Double>?**count=yes**

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{"vlan_linkset_binding": [ { "__count": "#no"} ] }
```

vlan_nsip6_binding

Binding object showing the nsip6 that can be bound to vlan.

Properties

(click to see Operations)

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
ownergroup	<String>	Read-write	The owner node group in a Cluster for this vlan. Default value: DEFAULT_NG Minimum length = 1
id	<Double>	Read-write	Specifies the virtual LAN ID. Minimum value = 1 Maximum value = 4094
td	<Double>	Read-write	Integer value that uniquely identifies the traffic domain in which you want to configure the entity. If you do not specify an ID, the entity becomes part of the default traffic domain, which has an ID of 0. Minimum value = 0 Maximum value = 4094
netmask	<String>	Read-write	Subnet mask for the network address defined for this VLAN. Minimum length = 1
ipaddress	<String>	Read-write	The IP address assigned to the VLAN.

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
__count	<Double>	Read-write	count parameter

Operations

(click to see Properties)

ADD:| DELETE:| GET| GET (ALL)| COUNT

Some options that you can use for each operations:

- **Getting warnings in response:**NITRO allows you to get warnings in an operation by specifying the "warning" query parameter as "yes". For example, to get warnings while connecting to the Citrix ADC appliance, the URL is as follows:

  http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/login?warning=yes

  If any, the warnings are displayed in the response payload with the HTTP code "209 X-NITRO-WARNING".
- **Authenticated access for individual NITRO operations:**NITRO allows you to logon to the Citrix ADC appliance to perform individual operations. You can use this option instead of creating a NITRO session (using the login object) and then using that session to perform all operations,

  To do this, you must specify the username and password in the request header of the NITRO request as follows:

  X-NITRO-USER:<username>

  X-NITRO-PASS:<password>

  Note: In such cases, make sure that the request header DOES not include the following:

  Cookie:NITRO_AUTH_TOKEN=<tokenvalue>

`*Note:`*

Mandatory parameters are marked in redand placeholder content is marked in <green>.

add

URL:http://<Citrix-ADC-IP-address(NSIP)/nitro/v1/config/vlan_nsip6_binding

HTTP Method:PUT

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Content-Type:application/json

Request Payload:

```
{

"vlan_nsip6_binding":{

<b>"id":<Double_value>,

</b>"ipaddress":<String_value>,

"netmask":<String_value>,

"td":<Double_value>,

"ownergroup":<String_value>

}}
```

Response:

HTTP Status Code on Success: 201 Created HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error

delete

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/vlan_nsip6_binding/id_value<Double>

Query-parameters:

args

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/vlan_nsip6_binding/id_value<Double>?
args=ipaddress:<String_value>,netmask:<String_value>,td:<Double_value>,ownergroup:<String_value

HTTP Method:DELETE

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue>

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error

get

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/vlan_nsip6_binding/id_value<Double>

Query-parameters:

filter

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/vlan_nsip6_binding/id_value<Double>?**filter=property-name1:property-value1,property-name2:property-value2**

Use this query-parameter to get the filtered set of vlan_nsip6_binding resources configured on Citrix ADC. Filtering can be done on any of the properties of the resource.

pagination

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/vlan_nsip6_binding/id_value<Double>?**pagesize=#no;pageno=#no**

Use this query-parameter to get the vlan_nsip6_binding resources in chunks.

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "vlan_nsip6_binding": [ {

"ownergroup":<String_value>,

"id":<Double_value>,

"td":<Double_value>,

"netmask":<String_value>,

"ipaddress":<String_value>

}]}
```

get (all)

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/vlan_nsip6_binding

Query-parameters:

bulkbindings

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/vlan_nsip6_binding?**bulkbindings=yes**

NITRO allows you to fetch bindings in bulk.

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "vlan_nsip6_binding": [ {

"ownergroup":<String_value>,

"id":<Double_value>,

"td":<Double_value>,

"netmask":<String_value>,

"ipaddress":<String_value>

}]}
```

count

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/vlan_nsip6_binding/id_value<Double>?**count=yes**

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{"vlan_nsip6_binding": [ { "__count": "#no"} ] }
```

vlan_nsip_binding

Binding object showing the nsip that can be bound to vlan.

Properties

(click to see Operations)

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
ownergroup	<String>	Read-write	The owner node group in a Cluster for this vlan. Default value: DEFAULT_NG Minimum length = 1
id	<Double>	Read-write	Specifies the virtual LAN ID. Minimum value = 1 Maximum value = 4094
td	<Double>	Read-write	Integer value that uniquely identifies the traffic domain in which you want to configure the entity. If you do not specify an ID, the entity becomes part of the default traffic domain, which has an ID of 0. Minimum value = 0 Maximum value = 4094
netmask	<String>	Read-write	Subnet mask for the network address

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
			defined for this VLAN.
ipaddress	<String>	Read-write	The IP address assigned to the VLAN.
__count	<Double>	Read-write	count parameter

Operations

(click to see Properties)

ADD:| DELETE:| GET| GET (ALL)| COUNT

Some options that you can use for each operations:

- **Getting warnings in response:**NITRO allows you to get warnings in an operation by specifying the "warning" query parameter as "yes". For example, to get warnings while connecting to the Citrix ADC appliance, the URL is as follows:

  http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/login?warning=yes

  If any, the warnings are displayed in the response payload with the HTTP code "209 X-NITRO-WARNING".

- **Authenticated access for individual NITRO operations:**NITRO allows you to logon to the Citrix ADC appliance to perform individual operations. You can use this option instead of creating a NITRO session (using the login object) and then using that session to perform all operations,

  To do this, you must specify the username and password in the request header of the NITRO request as follows:

  X-NITRO-USER:<username>

  X-NITRO-PASS:<password>

  Note: In such cases, make sure that the request header DOES not include the following:

  Cookie:NITRO_AUTH_TOKEN=<tokenvalue>

`*Note:` *

Mandatory parameters are marked in redand placeholder content is marked in <green>.

add

URL:http://<Citrix-ADC-IP-address(NSIP)/nitro/v1/config/vlan_nsip_binding

HTTP Method:PUT

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Content-Type:application/json

Request Payload:

```
{

"vlan_nsip_binding":{

<b>"id":<Double_value>,

</b>"ipaddress":<String_value>,

"netmask":<String_value>,

"td":<Double_value>,

"ownergroup":<String_value>

}}
```

Response:

HTTP Status Code on Success: 201 Created HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error

delete

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/vlan_nsip_binding/id_value<Double>

Query-parameters:

args

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/vlan_nsip_binding/id_value<Double>?
args=ipaddress:<String_value>,netmask:<String_value>,td:<Double_value>,ownergroup:<String_value

HTTP Method:DELETE

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue>

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error

get

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/vlan_nsip_binding/id_value<Double>

Query-parameters:

filter

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/vlan_nsip_binding/id_value<Double>?
filter=property-name1:property-value1,property-name2:property-value2

Use this query-parameter to get the filtered set of vlan_nsip_binding resources configured on Citrix ADC. Filtering can be done on any of the properties of the resource.

pagination

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/vlan_nsip_binding/id_value<Double>?
pagesize=#no;pageno=#no

Use this query-parameter to get the vlan_nsip_binding resources in chunks.

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "vlan_nsip_binding": [ {

"ownergroup":<String_value>,

"id":<Double_value>,

"td":<Double_value>,

"netmask":<String_value>,

"ipaddress":<String_value>

}]}
```

get (all)

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/vlan_nsip_binding

Query-parameters:

bulkbindings

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/vlan_nsip_binding?**bulkbindings=yes**

NITRO allows you to fetch bindings in bulk.

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "vlan_nsip_binding": [ {

"ownergroup":<String_value>,

"id":<Double_value>,

"td":<Double_value>,

"netmask":<String_value>,

"ipaddress":<String_value>

}]}
```

count

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/vlan_nsip_binding/id_value<Double>?**count=yes**

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{"vlan_nsip_binding": [ { "__count": "#no"} ] }
```

vrid

Configuration for Virtual Router ID resource.

Properties

(click to see Operations)

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
id	<Double>	Read-write	Integer that uniquely identifies the VMAC address. The generic VMAC address is in the form of 00:00:5e:00:01:<VRID>. For example, if you add a VRID with a value of 60 and bind it to an interface, the resulting VMAC address is 00:00:5e:00:01:3c, where 3c is the hexadecimal representation of 60. Minimum value = 1 Maximum value = 255
priority	<Double>	Read-write	Base priority (BP), in an active-active mode configuration, which ordinarily

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
			determines the master VIP address. Default value: 255 Minimum value = 0 Maximum value = 255
preemption	<String>	Read-write	In an active-active mode configuration, make a backup VIP address the master if its priority becomes higher than that of a master VIP address bound to this VMAC address. If you disable pre-emption while a backup VIP address is the master, the backup VIP address remains master until the original master VIP's priority becomes higher than that of the current master. Default value: ENABLED Possible values = ENABLED, DISABLED
sharing	<String>	Read-write	In an active-active mode configuration, enable the backup

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
			VIP address to process any traffic instead of dropping it. Default value: DISABLED Possible values = ENABLED, DISABLED
tracking	<String>	Read-write	The effective priority (EP) value, relative to the base priority (BP) value in an active-active mode configuration. When EP is set to a value other than None, it is EP, not BP, which determines the master VIP address. Available settings function as follows: * NONE - No tracking. EP = BP * ALL - If the status of all virtual servers is UP, EP = BP. Otherwise, EP = 0. * ONE - If the status of at least one virtual server is UP, EP = BP. Otherwise, EP = 0.

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
			* PROGRESSIVE - If the status of all virtual servers is UP, EP = BP. If the status of all virtual servers is DOWN, EP = 0. Otherwise EP = BP (1 - K/N), where N is the total number of virtual servers associated with the VIP address and K is the number of virtual servers for which the status is DOWN. Default: NONE. Default value: NONE Possible values = NONE, ONE, ALL, PROGRESSIVE
ownernode	<Double>	Read-write	In a cluster setup, assign a cluster node as the owner of this VMAC address for IP based VRRP configuration. If no owner is configured, owner node is displayed as ALL and one node is

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
			dynamically elected as the owner. Minimum value = 0 Maximum value = 31
trackifnumpriority	<Double>	Read-write	Priority by which the Effective priority will be reduced if any of the tracked interfaces goes down in an active-active configuration. Default value: 0 Minimum value = 1 Maximum value = 255
preemptiondelaytimer	<Double>	Read-write	Preemption delay time, in seconds, in an active-active configuration. If any high priority node will come in network, it will wait for these many seconds before becoming master. Default value: 0 Minimum value = 1 Maximum value = 36000
all	<Boolean>	Read-write	Remove all the configured VMAC

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
			addresses from the Citrix ADC.
ifaces	<String>	Read-only	Interfaces bound to this VRID.
type	<String>	Read-only	Indicates whether this VRID entry was added manually or dynamically. When you manually add a VRID entry, the value for this parameter is STATIC. Otherwise, it is DYNAMIC. Possible values = STATIC, DYNAMIC
effectivepriority	<Double>	Read-only	The effective priority of this VRID.
flags	<Double>	Read-only	Flags.
ipaddress	<String>	Read-only	The IP address bound to the VRID.
state	<Double>	Read-only	State of this VRID.
operationalownernode	<Double>	Read-only	Run time owner node of the vrid. Minimum value = 0 Maximum value = 31
__count	<Double>	Read-only	count parameter

Operations

(click to see Properties)

ADD| DELETE| UPDATE| UNSET| GET (ALL)| GET| COUNT

Some options that you can use for each operations:

- **Getting warnings in response:**NITRO allows you to get warnings in an operation by specifying the "warning" query parameter as "yes". For example, to get warnings while connecting to the Citrix ADC appliance, the URL is as follows:

  http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/login?warning=yes

  If any, the warnings are displayed in the response payload with the HTTP code "209 X-NITRO-WARNING".

- **Authenticated access for individual NITRO operations:**NITRO allows you to logon to the Citrix ADC appliance to perform individual operations. You can use this option instead of creating a NITRO session (using the login object) and then using that session to perform all operations,

  To do this, you must specify the username and password in the request header of the NITRO request as follows:

  X-NITRO-USER:<username>

  X-NITRO-PASS:<password>

  Note: In such cases, make sure that the request header DOES not include the following:

  Cookie:NITRO_AUTH_TOKEN=<tokenvalue>

`*Note:`*

Mandatory parameters are marked in redand placeholder content is marked in <green>.

add

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/vrid

HTTP Method:POST

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Content-Type:application/json

Request Payload:

```
{"vrid":{

<b>"id":<Double_value>,

</b>"priority":<Double_value>,

"preemption":<String_value>,

"sharing":<String_value>,

"tracking":<String_value>,

"ownernode":<Double_value>,

"trackifnumpriority":<Double_value>,

"preemptiondelaytimer":<Double_value>

}}
```

Response:

HTTP Status Code on Success: 201 Created HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error

delete

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/vrid/id_value<Double>

HTTP Method:DELETE

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue>

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error

update

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/vrid

HTTP Method:PUT

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Content-Type:application/json

Request Payload:

```
{"vrid":{

<b>"id":<Double_value>,

</b>"priority":<Double_value>,

"preemption":<String_value>,

"sharing":<String_value>,

"tracking":<String_value>,

"ownernode":<Double_value>,

"trackifnumpriority":<Double_value>,

"preemptiondelaytimer":<Double_value>

}}
```

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error

unset

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/vrid?**action=unset**

HTTP Method:POST

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Content-Type:application/json

Request Payload:

```
{"vrid":{

<b>"id":<Double_value>,

</b>"priority":true,

"preemption":true,

"sharing":true,

"tracking":true,

"ownernode":true,

"trackifnumpriority":true,

"preemptiondelaytimer":true

}}
```

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error

get (all)

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/vrid

Query-parameters:

attrs

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/vrid?**attrs=property-name1,property-name2**

Use this query parameter to specify the resource details that you want to retrieve.

filter

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/vrid?**filter=property-name1:property-val1,property-name2:property-val2**

Use this query-parameter to get the filtered set of vrid resources configured on Citrix ADC. Filtering can be done on any of the properties of the resource.

view

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/vrid?**view=summary**

Note: By default, the retrieved results are displayed in detail view (?view=detail).

pagination

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/vrid?**pagesize=#no;pageno=#no**

Use this query-parameter to get the vrid resources in chunks.

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "vrid": [ {

"id":<Double_value>,
```

```
"ifaces":<String_value>,

"type":<String_value>,

"priority":<Double_value>,

"effectivepriority":<Double_value>,

"preemption":<String_value>,

"sharing":<String_value>,

"tracking":<String_value>,

"flags":<Double_value>,

"ipaddress":<String_value>,

"state":<Double_value>,

"operationalownernode":<Double_value>,

"ownernode":<Double_value>,

"trackifnumpriority":<Double_value>,

"preemptiondelaytimer":<Double_value>

}]}
```

get

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/vrid/id_value<Double>

Query-parameters:

attrs

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/vrid/id_value<Double>?**attrs=property-name1,property-name2**

Use this query parameter to specify the resource details that you want to retrieve.

view

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/vrid/id_value<Double>?**view=summary**

Note: By default, the retrieved results are displayed in detail view (?view=detail).

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "vrid": [ {

"id":<Double_value>,

"ifaces":<String_value>,

"type":<String_value>,

"priority":<Double_value>,

"effectivepriority":<Double_value>,

"preemption":<String_value>,

"sharing":<String_value>,

"tracking":<String_value>,

"flags":<Double_value>,
```

```
"ipaddress":<String_value>,

"state":<Double_value>,

"operationalownernode":<Double_value>,

"ownernode":<Double_value>,

"trackifnumpriority":<Double_value>,

"preemptiondelaytimer":<Double_value>

}]}
```

count

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/vrid?**count=yes**

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "vrid": [ { "__count": "#no"} ] }
```

vrid6

Configuration for Virtual Router ID for IPv6 resource.

Properties

(click to see Operations)

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
id	<Double>	Read-write	Integer value that uniquely identifies a VMAC6 address. Minimum value = 1 Maximum value = 255
priority	<Double>	Read-write	Base priority (BP), in an active-active mode configuration, which ordinarily determines the master VIP address. Default value: 255 Minimum value = 0 Maximum value = 255
preemption	<String>	Read-write	In an active-active mode configuration, make a backup VIP address the master if its priority becomes higher than that of a master VIP address bound to this VMAC address. If you disable pre-emption while a backup VIP address is the master, the backup VIP address remains master until the original master

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
			VIP's priority becomes higher than that of the current master. Default value: ENABLED Possible values = ENABLED, DISABLED
sharing	<String>	Read-write	In an active-active mode configuration, enable the backup VIP address to process any traffic instead of dropping it. Default value: DISABLED Possible values = ENABLED, DISABLED
tracking	<String>	Read-write	The effective priority (EP) value, relative to the base priority (BP) value in an active-active mode configuration. When EP is set to a value other than None, it is EP, not BP, which determines the master VIP address. Available settings function as follows:

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
			* NONE - No tracking. EP = BP * ALL - If the status of all virtual servers is UP, EP = BP. Otherwise, EP = 0. * ONE - If the status of at least one virtual server is UP, EP = BP. Otherwise, EP = 0. * PROGRESSIVE - If the status of all virtual servers is UP, EP = BP. If the status of all virtual servers is DOWN, EP = 0. Otherwise EP = BP (1 - K/N), where N is the total number of virtual servers associated with the VIP address and K is the number of virtual servers for which the status is DOWN. Default: NONE. Default value: NONE

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
			Possible values = NONE, ONE, ALL, PROGRESSIVE
preemptiondelaytimer	<Double>	Read-write	Preemption delay time in seconds, in an active-active configuration. If any high priority node will come in network, it will wait for these many seconds before becoming master. Default value: 0 Minimum value = 1 Maximum value = 36000
trackifnumpriority	<Double>	Read-write	Priority by which the Effective priority will be reduced if any of the tracked interfaces goes down in an active-active configuration. Default value: 0 Minimum value = 1 Maximum value = 255
ownernode	<Double>	Read-write	In a cluster setup, assign a cluster node as the owner of this VMAC address for IP based VRRP configuration. If no

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
			owner is configured, ow ner node is displayed as ALL and one node is dynamically elected as the owner. Minimum value = 0 Maximum value = 31
all	<Boolean>	Read-write	Remove all configured VMAC6 addresses from the Citrix ADC.
ifaces	<String>	Read-only	Interfaces bound to this VRID.
ifnum	<String>	Read-only	Interfaces to bind to the VMAC6, specified in (slot/port) notation (for example, 1/2).Use spaces to separate multiple entries.
type	<String>	Read-only	Type (static or dynamic) of this VRID. Possible values = STATIC, DYNAMIC
state	<Double>	Read-only	State of this VRID.
flags	<Double>	Read-only	Flags.
ipaddress	<String>	Read-only	The IP address bound to the VRID6.

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
effectivepriority	<Double>	Read-only	The effective priority of this VRID.
operationalownernode	<Double>	Read-only	Run time owner node of the vrid. Minimum value = 0 Maximum value = 31
__count	<Double>	Read-only	count parameter

Operations

(click to see Properties)

ADD| DELETE| UPDATE| UNSET| GET (ALL)| GET| COUNT

Some options that you can use for each operations:

- **Getting warnings in response:**NITRO allows you to get warnings in an operation by specifying the "warning" query parameter as "yes". For example, to get warnings while connecting to the Citrix ADC appliance, the URL is as follows:

  http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/login?warning=yes

  If any, the warnings are displayed in the response payload with the HTTP code "209 X-NITRO-WARNING".

- **Authenticated access for individual NITRO operations:**NITRO allows you to logon to the Citrix ADC appliance to perform individual operations. You can use this option instead of creating a NITRO session (using the login object) and then using that session to perform all operations,

  To do this, you must specify the username and password in the request header of the NITRO request as follows:

  X-NITRO-USER:<username>

  X-NITRO-PASS:<password>

  Note: In such cases, make sure that the request header DOES not include the following:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue>

`*Note:`*

Mandatory parameters are marked in redand placeholder content is marked in <green>.

add

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/vrid6

HTTP Method:POST

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Content-Type:application/json

Request Payload:

```
{"vrid6":{

<b>"id":<Double_value>,

</b>"priority":<Double_value>,

"preemption":<String_value>,

"sharing":<String_value>,

"tracking":<String_value>,

"preemptiondelaytimer":<Double_value>,

"trackifnumpriority":<Double_value>,

"ownernode":<Double_value>

}}
```

Response:

HTTP Status Code on Success: 201 Created HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error

delete

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/vrid6/id_value<Double>

HTTP Method:DELETE

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue>

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error

update

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/vrid6

HTTP Method:PUT

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Content-Type:application/json

Request Payload:

```
{"vrid6":{

<b>"id":<Double_value>,

</b>"priority":<Double_value>,

"preemption":<String_value>,

"sharing":<String_value>,

"tracking":<String_value>,
```

```
"preemptiondelaytimer":<Double_value>,

"trackifnumpriority":<Double_value>,

"ownernode":<Double_value>

}}
```

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error

unset

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/vrid6?**action=unset**

HTTP Method:POST

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Content-Type:application/json

Request Payload:

```
{"vrid6":{

<b>"id":<Double_value>,

</b>"priority":true,

"preemption":true,

"sharing":true,

"tracking":true,

"preemptiondelaytimer":true,
```

```
"trackifnumpriority":true,

"ownernode":true

}}
```

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error

get (all)

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/vrid6

Query-parameters:

attrs

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/vrid6?**attrs=property-name1,property-name2**

Use this query parameter to specify the resource details that you want to retrieve.

filter

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/vrid6?**filter=property-name1:property-val1,property-name2:property-val2**

Use this query-parameter to get the filtered set of vrid6 resources configured on Citrix ADC. Filtering can be done on any of the properties of the resource.

view

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/vrid6?**view=summary**

Note: By default, the retrieved results are displayed in detail view (?view=detail).

pagination

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/vrid6?**pagesize=#no;pageno=#no**

Use this query-parameter to get the vrid6 resources in chunks.

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "vrid6": [ {

"id":<Double_value>,

"ifaces":<String_value>,

"ifnum":<String_value>,

"type":<String_value>,

"priority":<Double_value>,

"state":<Double_value>,

"flags":<Double_value>,

"ipaddress":<String_value>,

"preemption":<String_value>,

"sharing":<String_value>,

"preemptiondelaytimer":<Double_value>,
```

```
"trackifnumpriority":<Double_value>,

"effectivepriority":<Double_value>,

"tracking":<String_value>,

"operationalownernode":<Double_value>,

"ownernode":<Double_value>

}]}
```

get

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/vrid6/id_value<Double>

Query-parameters:

attrs

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/vrid6/id_value<Double>?**attrs=property-name1,property-name2**

Use this query parameter to specify the resource details that you want to retrieve.

view

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/vrid6/id_value<Double>?**view=summary**

Note: By default, the retrieved results are displayed in detail view (?view=detail).

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "vrid6": [ {

"id":<Double_value>,

"ifaces":<String_value>,

"ifnum":<String_value>,

"type":<String_value>,

"priority":<Double_value>,

"state":<Double_value>,

"flags":<Double_value>,

"ipaddress":<String_value>,

"preemption":<String_value>,

"sharing":<String_value>,

"preemptiondelaytimer":<Double_value>,

"trackifnumpriority":<Double_value>,

"effectivepriority":<Double_value>,

"tracking":<String_value>,

"operationalownernode":<Double_value>,

"ownernode":<Double_value>

}]}
```

count

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/vrid6?**count=yes**

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "vrid6": [ { "__count": "#no"} ] }
```

vrid6_binding

Binding object which returns the resources bound to vrid6.

Properties

(click to see Operations)

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
id	<Double>	Read-write	Integer value that uniquely identifies a VMAC6 address. Minimum value = 1 Maximum value = 255

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
vrid6_interface_binding	<vrid6_interface_binding[]>	Read-only	interface that can be bound to vrid6.
vrid6_trackinterface_binding	<vrid6_trackinterface_binding[]>	Read-only	trackinterface that can be bound to vrid6.
vrid6_channel_binding	<vrid6_channel_binding[]>	Read-only	channel that can be bound to vrid6.
vrid6_nsip_binding	<vrid6_nsip_binding[]>	Read-only	nsip that can be bound to vrid6.
vrid6_nsip6_binding	<vrid6_nsip6_binding[]>	Read-only	nsip6 that can be bound to vrid6.

Operations

(click to see Properties)

GET| GET (ALL)

Some options that you can use for each operations:

- **Getting warnings in response:**NITRO allows you to get warnings in an operation by specifying the "warning" query parameter as "yes". For example, to get warnings while connecting to the Citrix ADC appliance, the URL is as follows:

  http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/login?warning=yes

  If any, the warnings are displayed in the response payload with the HTTP code "209 X-NITRO-WARNING".

- **Authenticated access for individual NITRO operations:**NITRO allows you to logon to the Citrix ADC appliance to perform individual operations. You can use this option instead of creating a NITRO session (using the login object) and then using that session to perform all operations,

  To do this, you must specify the username and password in the request header of the NITRO request as follows:

  X-NITRO-USER:<username>

X-NITRO-PASS:<password>

Note: In such cases, make sure that the request header DOES not include the following:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue>

`*Note:`*

Mandatory parameters are marked in redand placeholder content is marked in <green>.

get

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/vrid6_binding/id_value<Double>

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "vrid6_binding": [ {

"id":<Double_value>,

"vrid6_nsip6_binding":<vrid6_nsip6_binding[]_value>,

"vrid6_nsip_binding":<vrid6_nsip_binding[]_value>,

"vrid6_channel_binding":<vrid6_channel_binding[]_value>,

"vrid6_interface_binding":<vrid6_interface_binding[]_value>,

"vrid6_trackinterface_binding":<vrid6_trackinterface_binding[]_value>
```

```
}]}
```

get (all)

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/vrid6_binding

Query-parameters:

bulkbindings

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/vrid6_binding?**bulkbindings=yes**

NITRO allows you to fetch bindings in bulk.

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "vrid6_binding": [ {

"id":<Double_value>,

"vrid6_nsip6_binding":<vrid6_nsip6_binding[]_value>,

"vrid6_nsip_binding":<vrid6_nsip_binding[]_value>,

"vrid6_channel_binding":<vrid6_channel_binding[]_value>,

"vrid6_interface_binding":<vrid6_interface_binding[]_value>,
```

```
"vrid6_trackinterface_binding":<vrid6_trackinterface_binding[]_value>

}]}
```

vrid6_channel_binding

Binding object showing the channel that can be bound to vrid6.

Properties

(click to see Operations)

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
id	<Double>	Read-write	Integer value that uniquely identifies a VMAC6 address. Minimum value = 1 Maximum value = 255
ifnum	<String>	Read-write	Interfaces to bind to the VMAC6, specified in (slot/port) notation (for example, 1/2).Use spaces to separate multiple entries.
flags	<Double>	Read-only	Flags.
vlan	<Double>	Read-only	The VLAN in which this VRID resides.
__count	<Double>	Read-only	count parameter

Operations

(click to see Properties)

ADD:| DELETE:| GET| GET (ALL)| COUNT

Some options that you can use for each operations:

- **Getting warnings in response:**NITRO allows you to get warnings in an operation by specifying the "warning" query parameter as "yes". For example, to get warnings while connecting to the Citrix ADC appliance, the URL is as follows:

  http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/login?warning=yes

  If any, the warnings are displayed in the response payload with the HTTP code "209 X-NITRO-WARNING".

- **Authenticated access for individual NITRO operations:**NITRO allows you to logon to the Citrix ADC appliance to perform individual operations. You can use this option instead of creating a NITRO session (using the login object) and then using that session to perform all operations,

  To do this, you must specify the username and password in the request header of the NITRO request as follows:

  X-NITRO-USER:<username>

  X-NITRO-PASS:<password>

  Note: In such cases, make sure that the request header DOES not include the following:

  Cookie:NITRO_AUTH_TOKEN=<tokenvalue>

`*Note:`*

Mandatory parameters are marked in redand placeholder content is marked in <green>.

add

URL:http://<Citrix-ADC-IP-address(NSIP)/nitro/v1/config/vrid6_channel_binding

HTTP Method:PUT

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Content-Type:application/json

Request Payload:

```
{

"vrid6_channel_binding":{

<b>"id":<Double_value>,

</b>"ifnum":<String_value>

}}
```

Response:

HTTP Status Code on Success: 201 Created HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error

delete

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/vrid6_channel_binding/id_value<Double>

Query-parameters:

args

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/vrid6_channel_binding/id_value<Double>?**args=ifnum:<String_value>**

HTTP Method:DELETE

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue>

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error

get

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/vrid6_channel_binding/id_value<Double>

Query-parameters:

filter

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/vrid6_channel_binding/id_value<Double>?**filter=property-name1:property-value1,property-name2:property-value2**

Use this query-parameter to get the filtered set of vrid6_channel_binding resources configured on Citrix ADC. Filtering can be done on any of the properties of the resource.

pagination

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/vrid6_channel_binding/id_value<Double>?**pagesize=#no;pageno=#no**

Use this query-parameter to get the vrid6_channel_binding resources in chunks.

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "vrid6_channel_binding": [ {

"id":<Double_value>,

"ifnum":<String_value>,
```

```
"flags":<Double_value>,

"vlan":<Double_value>

}]}
```

get (all)

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/vrid6_channel_binding

Query-parameters:

bulkbindings

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/vrid6_channel_binding?**bulkbindings=yes**

NITRO allows you to fetch bindings in bulk.

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "vrid6_channel_binding": [ {

"id":<Double_value>,

"ifnum":<String_value>,

"flags":<Double_value>,
```

```
"vlan":<Double_value>

}]}
```

count

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/vrid6_channel_binding/id_value<Double>?**count=yes**

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{"vrid6_channel_binding": [ { "__count": "#no"} ] }
```

vrid6_interface_binding

Binding object showing the interface that can be bound to vrid6.

Properties

(click to see Operations)

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
id	<Double>	Read-write	Integer value that uniquely identifies a VMAC6 address. Minimum value = 1 Maximum value = 255
ifnum	<String>	Read-write	Interfaces to bind to the VMAC6, specified in (slot/port) notation (for example, 1/2).Use spaces to separate multiple entries.
flags	<Double>	Read-only	Flags.
vlan	<Double>	Read-only	The VLAN in which this VRID resides.
__count	<Double>	Read-only	count parameter

Operations

(click to see Properties)

ADD:| DELETE:| GET| GET (ALL)| COUNT

Some options that you can use for each operations:

- **Getting warnings in response:**NITRO allows you to get warnings in an operation by specifying the "warning" query parameter as "yes". For example, to get warnings while connecting to the Citrix ADC appliance, the URL is as follows:

  http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/login?warning=yes

  If any, the warnings are displayed in the response payload with the HTTP code "209 X-NITRO-WARNING".

- **Authenticated access for individual NITRO operations:**NITRO allows you to logon to the Citrix ADC appliance to perform individual operations. You can use this option instead of creating a NITRO session (using the login object) and then using that session to perform all operations,

  To do this, you must specify the username and password in the request header of the NITRO request as follows:

  X-NITRO-USER:<username>

  X-NITRO-PASS:<password>

  Note: In such cases, make sure that the request header DOES not include the following:

  Cookie:NITRO_AUTH_TOKEN=<tokenvalue>

`*Note:`*

Mandatory parameters are marked in redand placeholder content is marked in <green>.

add

URL:http://<Citrix-ADC-IP-address(NSIP)/nitro/v1/config/vrid6_interface_binding

HTTP Method:PUT

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Content-Type:application/json

Request Payload:

```
{

"vrid6_interface_binding":{

<b>"id":<Double_value>,

</b>"ifnum":<String_value>

}}
```

Response:

HTTP Status Code on Success: 201 Created HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error

delete

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/vrid6_interface_binding/id_value<Double>

Query-parameters:

args

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/vrid6_interface_binding/id_value<Double>?**args=ifnum:<String_value>**

HTTP Method:DELETE

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue>

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error

get

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/vrid6_interface_binding/id_value<Double>

Query-parameters:

filter

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/vrid6_interface_binding/id_value<Double>?**filter=property-name1:property-value1,property-name2:property-value2**

Use this query-parameter to get the filtered set of vrid6_interface_binding resources configured on Citrix ADC. Filtering can be done on any of the properties of the resource.

pagination

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/vrid6_interface_binding/id_value<Double>?**pagesize=#no;pageno=#no**

Use this query-parameter to get the vrid6_interface_binding resources in chunks.

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "vrid6_interface_binding": [ {

"id":<Double_value>,

"ifnum":<String_value>,

"flags":<Double_value>,

"vlan":<Double_value>

}]}
```

get (all)

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/vrid6_interface_binding

Query-parameters:

bulkbindings

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/vrid6_interface_binding?**bulkbindings=yes**

NITRO allows you to fetch bindings in bulk.

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "vrid6_interface_binding": [ {

"id":<Double_value>,

"ifnum":<String_value>,

"flags":<Double_value>,

"vlan":<Double_value>

}]}
```

count

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/vrid6_interface_binding/id_value<Double>?**count=yes**

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{"vrid6_interface_binding": [ { "__count": "#no"} ] }
```

vrid6_nsip6_binding

Binding object showing the nsip6 that can be bound to vrid6.

Properties

(click to see Operations)

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
id	<Double>	Read-write	Integer value that uniquely identifies a VMAC6 address. Minimum value = 1 Maximum value = 255
ipaddress	<String>	Read-write	The IP address bound to the VRID6.
flags	<Double>	Read-only	Flags.
__count	<Double>	Read-only	count parameter

Operations

(click to see Properties)

GET| GET (ALL)| COUNT

Some options that you can use for each operations:

- **Getting warnings in response:**NITRO allows you to get warnings in an operation by specifying the "warning" query parameter as "yes". For example, to get warnings while connecting to the Citrix ADC appliance, the URL is as follows:

  http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/login?warning=yes

  If any, the warnings are displayed in the response payload with the HTTP code "209 X-NITRO-WARNING".

- **Authenticated access for individual NITRO operations:**NITRO allows you to logon to the Citrix ADC appliance to perform individual operations. You can use this option instead of creating a NITRO session (using the login object) and then using that session to perform all operations,

  To do this, you must specify the username and password in the request header of the NITRO request as follows:

  X-NITRO-USER:<username>

  X-NITRO-PASS:<password>

  Note: In such cases, make sure that the request header DOES not include the following:

  Cookie:NITRO_AUTH_TOKEN=<tokenvalue>

`*Note:`*

Mandatory parameters are marked in redand placeholder content is marked in <green>.

get

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/vrid6_nsip6_binding/id_value<Double>

Query-parameters:

filter

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/vrid6_nsip6_binding/id_value<Double>?**filter=property-name1:property-value1,property-name2:property-value2**

Use this query-parameter to get the filtered set of vrid6_nsip6_binding resources configured on Citrix ADC. Filtering can be done on any of the properties of the resource.

pagination

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/vrid6_nsip6_binding/id_value<Double>?**pagesize=#no;pageno=#no**

Use this query-parameter to get the vrid6_nsip6_binding resources in chunks.

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "vrid6_nsip6_binding": [ {

"id":<Double_value>,

"ipaddress":<String_value>,

"flags":<Double_value>

}]}
```

get (all)

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/vrid6_nsip6_binding

Query-parameters:

bulkbindings

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/vrid6_nsip6_binding?**bulkbindings=yes**

NITRO allows you to fetch bindings in bulk.

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "vrid6_nsip6_binding": [ {

"id":<Double_value>,

"ipaddress":<String_value>,

"flags":<Double_value>

}]}
```

count

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/vrid6_nsip6_binding/id_value<Double>?**count=yes**

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{"vrid6_nsip6_binding": [ { "__count": "#no"} ] }
```

vrid6_nsip_binding

Binding object showing the nsip that can be bound to vrid6.

Properties

(click to see Operations)

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
id	<Double>	Read-write	Integer value that uniquely identifies a VMAC6 address. Minimum value = 1 Maximum value = 255
ipaddress	<String>	Read-write	The IP address bound to the VRID6.
flags	<Double>	Read-only	Flags.
__count	<Double>	Read-only	count parameter

Operations

(click to see Properties)

GET| GET (ALL)| COUNT

Some options that you can use for each operations:

- **Getting warnings in response:**NITRO allows you to get warnings in an operation by specifying the "warning" query parameter as "yes". For example, to get warnings while connecting to the Citrix ADC appliance, the URL is as follows:

  http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/login?warning=yes

  If any, the warnings are displayed in the response payload with the HTTP code "209 X-NITRO-WARNING".

- **Authenticated access for individual NITRO operations:**NITRO allows you to logon to the Citrix ADC appliance to perform individual operations. You can use this option instead of creating a NITRO session (using the login object) and then using that session to perform all operations,

  To do this, you must specify the username and password in the request header of the NITRO request as follows:

  X-NITRO-USER:<username>

  X-NITRO-PASS:<password>

  Note: In such cases, make sure that the request header DOES not include the following:

  Cookie:NITRO_AUTH_TOKEN=<tokenvalue>

`*Note:`*

Mandatory parameters are marked in redand placeholder content is marked in <green>.

get

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/vrid6_nsip_binding/id_value<Double>

Query-parameters:

filter

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/vrid6_nsip_binding/id_value<Double>?**filter=property-name1:property-value1,property-name2:property-value2**

Use this query-parameter to get the filtered set of vrid6_nsip_binding resources configured on Citrix ADC. Filtering can be done on any of the properties of the resource.

pagination

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/vrid6_nsip_binding/id_value<Double>?**pagesize=#no;pageno=#no**

Use this query-parameter to get the vrid6_nsip_binding resources in chunks.

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "vrid6_nsip_binding": [ {

"id":<Double_value>,

"ipaddress":<String_value>,

"flags":<Double_value>

}]}
```

get (all)

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/vrid6_nsip_binding

Query-parameters:

bulkbindings

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/vrid6_nsip_binding?**bulkbindings=yes**

NITRO allows you to fetch bindings in bulk.

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "vrid6_nsip_binding": [ {

"id":<Double_value>,

"ipaddress":<String_value>,

"flags":<Double_value>

}]}
```

count

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/vrid6_nsip_binding/id_value<Double>?**count=yes**

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{"vrid6_nsip_binding": [ { "__count": "#no"} ] }
```

vrid6_trackinterface_binding

Binding object showing the trackinterface that can be bound to vrid6.

Properties

(click to see Operations)

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
trackifnum	<String>	Read-write	Interfaces which need to be tracked for this vrID.
id	<Double>	Read-write	Integer value that uniquely identifies a VMAC6 address. Minimum value = 1 Maximum value = 255
flags	<Double>	Read-only	Flags.
__count	<Double>	Read-only	count parameter

Operations

(click to see Properties)

ADD:| DELETE:| GET| GET (ALL)| COUNT

Some options that you can use for each operations:

- **Getting warnings in response:**NITRO allows you to get warnings in an operation by specifying the "warning" query parameter as "yes". For example, to get warnings while connecting to the Citrix ADC appliance, the URL is as follows:

  http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/login?warning=yes

  If any, the warnings are displayed in the response payload with the HTTP code "209 X-NITRO-WARNING".

- **Authenticated access for individual NITRO operations:**NITRO allows you to logon to the Citrix ADC appliance to perform individual operations. You can use this option instead of creating a NITRO session (using the login object) and then using that session to perform all operations,

  To do this, you must specify the username and password in the request header of the NITRO request as follows:

  X-NITRO-USER:<username>

  X-NITRO-PASS:<password>

  Note: In such cases, make sure that the request header DOES not include the following:

  Cookie:NITRO_AUTH_TOKEN=<tokenvalue>

`*Note:`*

Mandatory parameters are marked in redand placeholder content is marked in <green>.

add

URL:http://<Citrix-ADC-IP-address(NSIP)/nitro/v1/config/vrid6_trackinterface_binding

HTTP Method:PUT

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Content-Type:application/json

Request Payload:

```
{

"vrid6_trackinterface_binding":{

<b>"id":<Double_value>,

</b>"trackifnum":<String_value>

}}
```

Response:

HTTP Status Code on Success: 201 Created HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error

delete

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/vrid6_trackinterface_binding/id_value<Double>

Query-parameters:

args

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/vrid6_trackinterface_binding/id_value<Double>?**args=trackifnum:<String_value>**

HTTP Method:DELETE

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue>

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error

get

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/vrid6_trackinterface_binding/id_value<Double>

Query-parameters:

filter

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/vrid6_trackinterface_binding/id_value<Double>?**filter=property-name1:property-value1,property-name2:property-value2**

Use this query-parameter to get the filtered set of vrid6_trackinterface_binding resources configured on Citrix ADC. Filtering can be done on any of the properties of the resource.

pagination

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/vrid6_trackinterface_binding/id_value<Double>?**pagesize=#no;pageno=#no**

Use this query-parameter to get the vrid6_trackinterface_binding resources in chunks.

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "vrid6_trackinterface_binding": [ {

"trackifnum":<String_value>,

"id":<Double_value>,
```

```
"flags":<Double_value>

}]}
```

get (all)

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/vrid6_trackinterface_binding

Query-parameters:

bulkbindings

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/vrid6_trackinterface_binding?**bulkbindings=yes**

NITRO allows you to fetch bindings in bulk.

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "vrid6_trackinterface_binding": [ {

"trackifnum":<String_value>,

"id":<Double_value>,

"flags":<Double_value>
```

```
}]}
```

count

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/vrid6_trackinterface_binding/id_value<Double>?**count=yes**

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{"vrid6_trackinterface_binding": [ { "__count": "#no"} ] }
```

vridparam

Configuration for VR ID parameter resource.

Properties

(click to see Operations)

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
sendtomaster	<String>	Read-write	Forward packets to the master node, in an active-active mode configuration,

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
			if the virtual server is in the backup state and sharing is disabled. Default value: DISABLED Possible values = ENABLED, DISABLED
hellointerval	<Double>	Read-write	Interval, in milliseconds, between vrrp advertisement messages sent to the peer node in active-active mode. Default value: 1000 Minimum value = 200 Maximum value = 1000
deadinterval	<Double>	Read-write	Number of seconds after which a peer node in active-active mode is marked down if vrrp advertisements are not received from the peer node. Default value: 3 Minimum value = 1 Maximum value = 60

Operations

(click to see Properties)

UPDATE| UNSET| GET (ALL)

Some options that you can use for each operations:

- **Getting warnings in response:**NITRO allows you to get warnings in an operation by specifying the "warning" query parameter as "yes". For example, to get warnings while connecting to the Citrix ADC appliance, the URL is as follows:

  http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/login?warning=yes

  If any, the warnings are displayed in the response payload with the HTTP code "209 X-NITRO-WARNING".

- **Authenticated access for individual NITRO operations:**NITRO allows you to logon to the Citrix ADC appliance to perform individual operations. You can use this option instead of creating a NITRO session (using the login object) and then using that session to perform all operations,

  To do this, you must specify the username and password in the request header of the NITRO request as follows:

  X-NITRO-USER:<username>

  X-NITRO-PASS:<password>

  Note: In such cases, make sure that the request header DOES not include the following:

  Cookie:NITRO_AUTH_TOKEN=<tokenvalue>

`*Note:`*

Mandatory parameters are marked in redand placeholder content is marked in <green>.

update

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/vridparam

HTTP Method:PUT

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Content-Type:application/json

Request Payload:

```
{"vridparam":{

"sendtomaster":<String_value>,

"hellointerval":<Double_value>,

"deadinterval":<Double_value>

}}
```

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error

unset

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/vridparam?**action=unset**

HTTP Method:POST

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Content-Type:application/json

Request Payload:

```
{"vridparam":{

"sendtomaster":true,

"hellointerval":true,

"deadinterval":true

}}
```

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error

get (all)

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/vridparam

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "vridparam": [ {

"sendtomaster":<String_value>,

"hellointerval":<Double_value>,

"deadinterval":<Double_value>

}]}
```

vrid_binding

Binding object which returns the resources bound to vrid.

Properties

(click to see Operations)

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
id	<Double>	Read-write	Integer value that uniquely identifies the VMAC address. Minimum value = 1 Maximum value = 255
vrid_trackinterface_binding	<vrid_trackinterface_binding[]>	Read-only	trackinterface that can be bound to vrid.
vrid_nsip6_binding	<vrid_nsip6_binding[]>	Read-only	nsip6 that can be bound to vrid.
vrid_interface_binding	<vrid_interface_binding[]>	Read-only	interface that can be bound to vrid.
vrid_channel_binding	<vrid_channel_binding[]>	Read-only	channel that can be bound to vrid.
vrid_nsip_binding	<vrid_nsip_binding[]>	Read-only	nsip that can be bound to vrid.

Operations

(click to see Properties)

GET| GET (ALL)

Some options that you can use for each operations:

- **Getting warnings in response:**NITRO allows you to get warnings in an operation by specifying the "warning" query parameter as "yes". For example, to get warnings while connecting to the Citrix ADC appliance, the URL is as follows:

  http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/login?warning=yes

If any, the warnings are displayed in the response payload with the HTTP code "209 X-NITRO-WARNING".

- **Authenticated access for individual NITRO operations:**NITRO allows you to logon to the Citrix ADC appliance to perform individual operations. You can use this option instead of creating a NITRO session (using the login object) and then using that session to perform all operations,

  To do this, you must specify the username and password in the request header of the NITRO request as follows:

  X-NITRO-USER:<username>

  X-NITRO-PASS:<password>

  Note: In such cases, make sure that the request header DOES not include the following:

  Cookie:NITRO_AUTH_TOKEN=<tokenvalue>

`*Note:`*

Mandatory parameters are marked in redand placeholder content is marked in <green>.

get

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/vrid_binding/id_value<Double>

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "vrid_binding": [ {

"id":<Double_value>,

"vrid_nsip6_binding":<vrid_nsip6_binding[]_value>,

"vrid_nsip_binding":<vrid_nsip_binding[]_value>,

"vrid_interface_binding":<vrid_interface_binding[]_value>,

"vrid_channel_binding":<vrid_channel_binding[]_value>,

"vrid_trackinterface_binding":<vrid_trackinterface_binding[]_value>

}]}
```

get (all)

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/vrid_binding

Query-parameters:

bulkbindings

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/vrid_binding?**bulkbindings=yes**

NITRO allows you to fetch bindings in bulk.

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "vrid_binding": [ {

"id":<Double_value>,

"vrid_nsip6_binding":<vrid_nsip6_binding[]_value>,

"vrid_nsip_binding":<vrid_nsip_binding[]_value>,

"vrid_interface_binding":<vrid_interface_binding[]_value>,

"vrid_channel_binding":<vrid_channel_binding[]_value>,

"vrid_trackinterface_binding":<vrid_trackinterface_binding[]_value>

}]}
```

vrid_channel_binding

Binding object showing the channel that can be bound to vrid.

Properties

(click to see Operations)

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
id	<Double>	Read-write	Integer that uniquely identifies the VMAC address. The generic VMAC address is in the form of 00:00:5e:00:01:<VRID>. For example, if you add a VRID with a value of 60 and bind it to an interface, the resulting VMAC

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
			address is 00:00:5e:00:01:3c, where 3c is the hexadecimal representation of 60. Minimum value = 1 Maximum value = 255
ifnum	<String>	Read-write	Interfaces to bind to the VMAC, specified in (slot/port) notation (for example, 1/2).Use spaces to separate multiple entries.
flags	<Double>	Read-only	Flags.
vlan	<Double>	Read-only	The VLAN in which this VRID resides.
__count	<Double>	Read-only	count parameter

Operations

(click to see Properties)

ADD:| DELETE:| GET| GET (ALL)| COUNT

Some options that you can use for each operations:

- **Getting warnings in response:**NITRO allows you to get warnings in an operation by specifying the "warning" query parameter as "yes". For example, to get warnings while connecting to the Citrix ADC appliance, the URL is as follows:

  http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/login?warning=yes

  If any, the warnings are displayed in the response payload with the HTTP code "209 X-NITRO-WARNING".

- **Authenticated access for individual NITRO operations:**NITRO allows you to logon to the Citrix ADC appliance to perform individual operations. You can use this option instead of creating a NITRO session (using the login object) and then using that session to perform all operations,

  To do this, you must specify the username and password in the request header of the NITRO request as follows:

  X-NITRO-USER:<username>

  X-NITRO-PASS:<password>

  Note: In such cases, make sure that the request header DOES not include the following:

  Cookie:NITRO_AUTH_TOKEN=<tokenvalue>

`*Note:`*

Mandatory parameters are marked in redand placeholder content is marked in <green>.

add

URL:http://<Citrix-ADC-IP-address(NSIP)/nitro/v1/config/vrid_channel_binding

HTTP Method:PUT

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Content-Type:application/json

Request Payload:

```
{

"vrid_channel_binding":{

<b>"id":<Double_value>,

</b>"ifnum":<String_value>

}}
```

Response:

HTTP Status Code on Success: 201 Created HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error

delete

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/vrid_channel_binding/id_value<Double>

Query-parameters:

args

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/vrid_channel_binding/id_value<Double>?**args=ifnum:<String_value>**

HTTP Method:DELETE

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue>

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error

get

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/vrid_channel_binding/id_value<Double>

Query-parameters:

filter

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/vrid_channel_binding/id_value<Double>?**filter=property-name1:property-value1,property-name2:property-value2**

Use this query-parameter to get the filtered set of vrid_channel_binding resources configured on Citrix ADC. Filtering can be done on any of the properties of the resource.

pagination

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/vrid_channel_binding/id_value<Double>?**pagesize=#no;pageno=#no**

Use this query-parameter to get the vrid_channel_binding resources in chunks.

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "vrid_channel_binding": [ {

"id":<Double_value>,

"ifnum":<String_value>,

"flags":<Double_value>,

"vlan":<Double_value>

}]}
```

get (all)

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/vrid_channel_binding

Query-parameters:

bulkbindings

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/vrid_channel_binding?**bulkbindings=yes**

NITRO allows you to fetch bindings in bulk.

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "vrid_channel_binding": [ {

"id":<Double_value>,

"ifnum":<String_value>,

"flags":<Double_value>,

"vlan":<Double_value>

}]}
```

count

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/vrid_channel_binding/id_value<Double>?**count=yes**

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{"vrid_channel_binding": [ { "__count": "#no"} ] }
```

vrid_interface_binding

Binding object showing the interface that can be bound to vrid.

Properties

(click to see Operations)

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
id	<Double>	Read-write	Integer that uniquely identifies the VMAC address. The generic VMAC address is in the form of 00:00:5e:00:01:<VRID>. For example, if you add a VRID with a value of 60 and bind it to an interface, the resulting VMAC address is 00:00:5e:00:01:3c, where 3c is the hexadecimal representation of 60. Minimum value = 1

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
			Maximum value = 255
ifnum	<String>	Read-write	Interfaces to bind to the VMAC, specified in (slot/port) notation (for example, 1/2).Use spaces to separate multiple entries.
flags	<Double>	Read-only	Flags.
vlan	<Double>	Read-only	The VLAN in which this VRID resides.
__count	<Double>	Read-only	count parameter

Operations

(click to see Properties)

ADD:| DELETE:| GET| GET (ALL)| COUNT

Some options that you can use for each operations:

- **Getting warnings in response:**NITRO allows you to get warnings in an operation by specifying the "warning" query parameter as "yes". For example, to get warnings while connecting to the Citrix ADC appliance, the URL is as follows:

  http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/login?warning=yes

  If any, the warnings are displayed in the response payload with the HTTP code "209 X-NITRO-WARNING".

- **Authenticated access for individual NITRO operations:**NITRO allows you to logon to the Citrix ADC appliance to perform individual operations. You can use this option instead of creating a NITRO session (using the login object) and then using that session to perform all operations,

  To do this, you must specify the username and password in the request header of the NITRO request as follows:

X-NITRO-USER:<username>

X-NITRO-PASS:<password>

Note: In such cases, make sure that the request header DOES not include the following:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue>

`*Note:`*

Mandatory parameters are marked in redand placeholder content is marked in <green>.

add

URL:http://<Citrix-ADC-IP-address(NSIP)/nitro/v1/config/vrid_interface_binding

HTTP Method:PUT

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Content-Type:application/json

Request Payload:

```
{

"vrid_interface_binding":{

<b>"id":<Double_value>,

</b>"ifnum":<String_value>

}}
```

Response:

HTTP Status Code on Success: 201 Created HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error

delete

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/vrid_interface_binding/id_value<Double>

Query-parameters:

args

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/vrid_interface_binding/id_value<Double>?**args=ifnum:<String_value>**

HTTP Method:DELETE

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue>

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error

get

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/vrid_interface_binding/id_value<Double>

Query-parameters:

filter

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/vrid_interface_binding/id_value<Double>?**filter=property-name1:property-value1,property-name2:property-value2**

Use this query-parameter to get the filtered set of vrid_interface_binding resources configured on Citrix ADC. Filtering can be done on any of the properties of the resource.

pagination

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/vrid_interface_binding/id_value<Double>?**pagesize=#no;pageno=#no**

Use this query-parameter to get the vrid_interface_binding resources in chunks.

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "vrid_interface_binding": [ {

"id":<Double_value>,

"ifnum":<String_value>,

"flags":<Double_value>,

"vlan":<Double_value>

}]}
```

get (all)

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/vrid_interface_binding

Query-parameters:

bulkbindings

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/vrid_interface_binding?**bulkbindings=yes**

NITRO allows you to fetch bindings in bulk.

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "vrid_interface_binding": [ {

"id":<Double_value>,

"ifnum":<String_value>,

"flags":<Double_value>,

"vlan":<Double_value>

}]}
```

count

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/vrid_interface_binding/id_value<Double>?**count=yes**

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{"vrid_interface_binding": [ { "__count": "#no"} ] }
```

vrid_nsip6_binding

Binding object showing the nsip6 that can be bound to vrid.

Properties

(click to see Operations)

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
id	<Double>	Read-write	Integer that uniquely identifies the VMAC address. The generic VMAC address is in the form of 00:00:5e:00:01:<VRID>. For example, if you add a VRID with a value of 60 and bind it to an interface, the resulting VMAC address is 00:00:5e:00:01:3c, where 3c is the hexadecimal representation of 60. Minimum value = 1

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
			Maximum value = 255
ipaddress	<String>	Read-write	The IP address bound to the VRID.
flags	<Double>	Read-only	Flags.
__count	<Double>	Read-only	count parameter

Operations

(click to see Properties)

GET| GET (ALL)| COUNT

Some options that you can use for each operations:

- **Getting warnings in response:**NITRO allows you to get warnings in an operation by specifying the "warning" query parameter as "yes". For example, to get warnings while connecting to the Citrix ADC appliance, the URL is as follows:

  http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/login?warning=yes

  If any, the warnings are displayed in the response payload with the HTTP code "209 X-NITRO-WARNING".

- **Authenticated access for individual NITRO operations:**NITRO allows you to logon to the Citrix ADC appliance to perform individual operations. You can use this option instead of creating a NITRO session (using the login object) and then using that session to perform all operations,

  To do this, you must specify the username and password in the request header of the NITRO request as follows:

  X-NITRO-USER:<username>

  X-NITRO-PASS:<password>

  Note: In such cases, make sure that the request header DOES not include the following:

  Cookie:NITRO_AUTH_TOKEN=<tokenvalue>

`*Note:` *

Mandatory parameters are marked in redand placeholder content is marked in <green>.

get

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/vrid_nsip6_binding/id_value<Double>

Query-parameters:

filter

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/vrid_nsip6_binding/id_value<Double>?**filter=property-name1:property-value1,property-name2:property-value2**

Use this query-parameter to get the filtered set of vrid_nsip6_binding resources configured on Citrix ADC. Filtering can be done on any of the properties of the resource.

pagination

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/vrid_nsip6_binding/id_value<Double>?**pagesize=#no;pageno=#no**

Use this query-parameter to get the vrid_nsip6_binding resources in chunks.

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "vrid_nsip6_binding": [ {

"id":<Double_value>,
```

```
"ipaddress":<String_value>,

"flags":<Double_value>

}]}
```

get (all)

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/vrid_nsip6_binding

Query-parameters:

bulkbindings

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/vrid_nsip6_binding?**bulkbindings=yes**

NITRO allows you to fetch bindings in bulk.

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "vrid_nsip6_binding": [ {

"id":<Double_value>,

"ipaddress":<String_value>,

"flags":<Double_value>
```

```
}]}
```

count

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/vrid_nsip6_binding/id_value<Double>?**count=yes**

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{"vrid_nsip6_binding": [ { "__count": "#no"} ] }
```

vrid_nsip_binding

Binding object showing the nsip that can be bound to vrid.

Properties

(click to see Operations)

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
id	<Double>	Read-write	Integer that uniquely identifies the VMAC address. The generic VMAC address is in

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
			the form of 00:00:5e:00:01:<VRID>. For example, if you add a VRID with a value of 60 and bind it to an interface, the resulting VMAC address is 00:00:5e:00:01:3c, where 3c is the hexadecimal representation of 60. Minimum value = 1 Maximum value = 255
ipaddress	<String>	Read-write	The IP address bound to the VRID.
flags	<Double>	Read-only	Flags.
__count	<Double>	Read-only	count parameter

Operations

(click to see Properties)

GET| GET (ALL)| COUNT

Some options that you can use for each operations:

- **Getting warnings in response:**NITRO allows you to get warnings in an operation by specifying the "warning" query parameter as "yes". For example, to get warnings while connecting to the Citrix ADC appliance, the URL is as follows:

  http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/login?warning=yes

  If any, the warnings are displayed in the response payload with the HTTP code "209 X-NITRO-WARNING".

- **Authenticated access for individual NITRO operations:**NITRO allows you to logon to the Citrix ADC appliance to perform individual operations. You can use this option instead of creating a NITRO session (using the login object) and then using that session to perform all operations,

  To do this, you must specify the username and password in the request header of the NITRO request as follows:

  X-NITRO-USER:<username>

  X-NITRO-PASS:<password>

  Note: In such cases, make sure that the request header DOES not include the following:

  Cookie:NITRO_AUTH_TOKEN=<tokenvalue>

`*Note:`*

Mandatory parameters are marked in redand placeholder content is marked in <green>.

get

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/vrid_nsip_binding/id_value<Double>

Query-parameters:

filter

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/vrid_nsip_binding/id_value<Double>?**filter=property-name1:property-value1,property-name2:property-value2**

Use this query-parameter to get the filtered set of vrid_nsip_binding resources configured on Citrix ADC. Filtering can be done on any of the properties of the resource.

pagination

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/vrid_nsip_binding/id_value<Double>?**pagesize=#no;pageno=#no**

Use this query-parameter to get the vrid_nsip_binding resources in chunks.

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "vrid_nsip_binding": [ {

"id":<Double_value>,

"ipaddress":<String_value>,

"flags":<Double_value>

}]}
```

get (all)

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/vrid_nsip_binding

Query-parameters:

bulkbindings

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/vrid_nsip_binding?**bulkbindings=yes**

NITRO allows you to fetch bindings in bulk.

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "vrid_nsip_binding": [ {

"id":<Double_value>,

"ipaddress":<String_value>,

"flags":<Double_value>

}]}
```

count

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/vrid_nsip_binding/id_value<Double>?**count=yes**

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{"vrid_nsip_binding": [ { "__count": "#no"} ] }
```

vrid_trackinterface_binding

Binding object showing the trackinterface that can be bound to vrid.

Properties

(click to see Operations)

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
id	<Double>	Read-write	Integer that uniquely identifies the VMAC address. The generic VMAC address is in the form of 00:00:5e:00:01:<VRID>. For example, if you add a VRID with a value of 60 and bind it to an interface, the resulting VMAC address is 00:00:5e:00:01:3c, where 3c is the hexadecimal representation of 60. Minimum value = 1 Maximum value = 255
trackifnum	<String>	Read-write	Interfaces which need to be tracked for this vrID.
flags	<Double>	Read-only	Flags.

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
__count	<Double>	Read-only	count parameter

Operations

(click to see Properties)

ADD:| DELETE:| GET| GET (ALL)| COUNT

Some options that you can use for each operations:

- **Getting warnings in response:**NITRO allows you to get warnings in an operation by specifying the "warning" query parameter as "yes". For example, to get warnings while connecting to the Citrix ADC appliance, the URL is as follows:

  http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/login?warning=yes

  If any, the warnings are displayed in the response payload with the HTTP code "209 X-NITRO-WARNING".
- **Authenticated access for individual NITRO operations:**NITRO allows you to logon to the Citrix ADC appliance to perform individual operations. You can use this option instead of creating a NITRO session (using the login object) and then using that session to perform all operations,

  To do this, you must specify the username and password in the request header of the NITRO request as follows:

  X-NITRO-USER:<username>

  X-NITRO-PASS:<password>

  Note: In such cases, make sure that the request header DOES not include the following:

  Cookie:NITRO_AUTH_TOKEN=<tokenvalue>

`*Note:`*

Mandatory parameters are marked in redand placeholder content is marked in <green>.

add

URL:http://<Citrix-ADC-IP-address(NSIP)/nitro/v1/config/vrid_trackinterface_binding

HTTP Method:PUT

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Content-Type:application/json

Request Payload:

```
{

"vrid_trackinterface_binding":{

<b>"id":<Double_value>,

</b>"trackifnum":<String_value>

}}
```

Response:

HTTP Status Code on Success: 201 Created HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error

delete

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/vrid_trackinterface_binding/id_value<Double>

Query-parameters:

args

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/vrid_trackinterface_binding/id_value<Double>?**args=trackifnum:<String_value>**

HTTP Method:DELETE

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue>

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error

get

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/vrid_trackinterface_binding/id_value<Double>

Query-parameters:

filter

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/vrid_trackinterface_binding/id_value<Double>?**filter=property-name1:property-value1,property-name2:property-value2**

Use this query-parameter to get the filtered set of vrid_trackinterface_binding resources configured on Citrix ADC. Filtering can be done on any of the properties of the resource.

pagination

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/vrid_trackinterface_binding/id_value<Double>?**pagesize=#no;pageno=#no**

Use this query-parameter to get the vrid_trackinterface_binding resources in chunks.

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "vrid_trackinterface_binding": [ {
```

```
"id":<Double_value>,

"trackifnum":<String_value>,

"flags":<Double_value>

}]}
```

get (all)

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/vrid_trackinterface_binding

Query-parameters:

bulkbindings

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/vrid_trackinterface_binding?**bulkbindings=yes**

NITRO allows you to fetch bindings in bulk.

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "vrid_trackinterface_binding": [ {

"id":<Double_value>,
```

```
"trackifnum":<String_value>,

"flags":<Double_value>

}]}
```

count

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/vrid_trackinterface_binding/id_value<Double>?**count=yes**

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{"vrid_trackinterface_binding": [ { "__count": "#no"} ] }
```

vxlan

Configuration for "VXLAN" resource.

Properties

(click to see Operations)

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
id	<Double>	Read-write	A positive integer, which is also called VXLAN Network Identifier (VNI), that uniquely identifies a VXLAN. Minimum value = 1 Maximum value = 16777215
vlan	<Double>	Read-write	ID of VLANs whose traffic is allowed over this VXLAN. If you do not specify any VLAN IDs, the Citrix ADC allows traffic of all VLANs that are not part of any other VXLANs. Minimum value = 2 Maximum value = 4094
port	<Integer>	Read-write	Specifies UDP destination port for VXLAN packets. Default value: 4789 Minimum value = 1 Maximum value = 65534
dynamicrouting	<String>	Read-write	Enable dynamic routing on this VXLAN.

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
			Default value: DISABLED Possible values = ENABLED, DISABLED
ipv6dynamicrouting	<String>	Read-write	Enable all IPv6 dynamic routing protocols on this VXLAN. Note: For the ENABLED setting to work, you must configure IPv6 dynamic routing protocols from the VTYSH command line. Default value: DISABLED Possible values = ENABLED, DISABLED
type	<String>	Read-write	VXLAN encapsulation type. VXLAN, VXLANGPE. Default value: VXLAN Possible values = VXLAN, VXLANGPE
protocol	<String>	Read-write	VXLAN-GPE next protocol. RESERVED, IPv4,

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
			IPv6, ETHERNET, NSH. Default value: ETHERNET Possible values = IPv4, IPv6, ETHERNET, NSH
innervlantagging	<String>	Read-write	Specifies whether Citrix ADC should generate VXLAN packets with inner VLAN tag. Default value: DISABLED Possible values = ENABLED, DISABLED
td	<Double>	Read-only	Integer value that uniquely identifies the traffic domain in which you want to configure the entity. If you do not specify an ID, the entity becomes part of the default traffic domain, which has an ID of 0.
partitionname	<String>	Read-only	The Partition to which this vxlan is bound. Minimum length = 1
__count	<Double>	Read-only	count parameter

Operations

(click to see Properties)

ADD| DELETE| UPDATE| UNSET| GET (ALL)| GET| COUNT

Some options that you can use for each operations:

- **Getting warnings in response:**NITRO allows you to get warnings in an operation by specifying the "warning" query parameter as "yes". For example, to get warnings while connecting to the Citrix ADC appliance, the URL is as follows:

  http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/login?warning=yes

  If any, the warnings are displayed in the response payload with the HTTP code "209 X-NITRO-WARNING".

- **Authenticated access for individual NITRO operations:**NITRO allows you to logon to the Citrix ADC appliance to perform individual operations. You can use this option instead of creating a NITRO session (using the login object) and then using that session to perform all operations,

  To do this, you must specify the username and password in the request header of the NITRO request as follows:

  X-NITRO-USER:<username>

  X-NITRO-PASS:<password>

  Note: In such cases, make sure that the request header DOES not include the following:

  Cookie:NITRO_AUTH_TOKEN=<tokenvalue>

`*Note:`*

Mandatory parameters are marked in redand placeholder content is marked in <green>.

add

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/vxlan

HTTP Method:POST

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Content-Type:application/json

Request Payload:

```
{"vxlan":{

<b>"id":<Double_value>,

</b>"vlan":<Double_value>,

"port":<Integer_value>,

"dynamicrouting":<String_value>,

"ipv6dynamicrouting":<String_value>,

"type":<String_value>,

"protocol":<String_value>,

"innervlantagging":<String_value>

}}
```

Response:

HTTP Status Code on Success: 201 Created HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error

delete

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/vxlan/id_value<Double>

HTTP Method:DELETE

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue>

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error

update

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/vxlan

HTTP Method:PUT

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Content-Type:application/json

Request Payload:

```
{"vxlan":{

<b>"id":<Double_value>,

</b>"vlan":<Double_value>,

"port":<Integer_value>,

"dynamicrouting":<String_value>,

"ipv6dynamicrouting":<String_value>,

"innervlantagging":<String_value>

}}
```

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error

unset

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/vxlan?**action=unset**

HTTP Method:POST

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Content-Type:application/json

Request Payload:

```
{"vxlan":{

<b>"id":<Double_value>,

</b>"vlan":true,

"port":true,

"dynamicrouting":true,

"ipv6dynamicrouting":true,

"innervlantagging":true

}}
```

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error

get (all)

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/vxlan

Query-parameters:

attrs

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/vxlan?**attrs=property-name1,property-name2**

Use this query parameter to specify the resource details that you want to retrieve.

filter

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/vxlan?**filter=property-name1:property-val1,property-name2:property-val2**

Use this query-parameter to get the filtered set of vxlan resources configured on Citrix ADC. Filtering can be done on any of the properties of the resource.

view

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/vxlan?**view=summary**

Note: By default, the retrieved results are displayed in detail view (?view=detail).

pagination

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/vxlan?**pagesize=#no;pageno=#no**

Use this query-parameter to get the vxlan resources in chunks.

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "vxlan": [ {

"id":<Double_value>,

"vlan":<Double_value>,
```

```
"port":<Integer_value>,

"dynamicrouting":<String_value>,

"ipv6dynamicrouting":<String_value>,

"innervlantagging":<String_value>,

"td":<Double_value>,

"type":<String_value>,

"protocol":<String_value>,

"partitionname":<String_value>

}]}
```

get

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/vxlan/id_value<Double>

Query-parameters:

attrs

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/vxlan/id_value<Double>?**attrs=property-name1,property-name2**

Use this query parameter to specify the resource details that you want to retrieve.

view

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/vxlan/id_value<Double>?**view=summary**

Note: By default, the retrieved results are displayed in detail view (?view=detail).

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "vxlan": [ {

"id":<Double_value>,

"vlan":<Double_value>,

"port":<Integer_value>,

"dynamicrouting":<String_value>,

"ipv6dynamicrouting":<String_value>,

"innervlantagging":<String_value>,

"td":<Double_value>,

"type":<String_value>,

"protocol":<String_value>,

"partitionname":<String_value>

}]}
```

count

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/vxlan?**count=yes**

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "vxlan": [ { "__count": "#no"} ] }
```

vxlanvlanmap

Configuration for vxlan vlan mapping resource.

Properties

(click to see Operations)

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
name	<String>	Read-write	Name of the mapping table. Minimum length = 1
__count	<Double>	Read-only	count parameter

Operations

(click to see Properties)

ADD| DELETE| GET (ALL)| GET| COUNT

Some options that you can use for each operations:

- **Getting warnings in response:**NITRO allows you to get warnings in an operation by specifying the "warning" query parameter as "yes". For example, to get warnings while connecting to the Citrix ADC appliance, the URL is as follows:

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/login?warning=yes

If any, the warnings are displayed in the response payload with the HTTP code "209 X-NITRO-WARNING".

- **Authenticated access for individual NITRO operations:**NITRO allows you to logon to the Citrix ADC appliance to perform individual operations. You can use this option instead of creating a NITRO session (using the login object) and then using that session to perform all operations,

  To do this, you must specify the username and password in the request header of the NITRO request as follows:

  X-NITRO-USER:<username>

  X-NITRO-PASS:<password>

  Note: In such cases, make sure that the request header DOES not include the following:

  Cookie:NITRO_AUTH_TOKEN=<tokenvalue>

`*Note:`*

Mandatory parameters are marked in redand placeholder content is marked in <green>.

add

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/vxlanvlanmap

HTTP Method:POST

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Content-Type:application/json

Request Payload:

```
{"vxlanvlanmap":{

<b>"name":<String_value>

</b>}}
```

Response:

HTTP Status Code on Success: 201 Created HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error

delete

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/vxlanvlanmap/name_value<String>

HTTP Method:DELETE

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue>

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error

get (all)

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/vxlanvlanmap

Query-parameters:

attrs

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/vxlanvlanmap?**attrs=property-name1,property-name2**

Use this query parameter to specify the resource details that you want to retrieve.

filter

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/vxlanvlanmap?**filter=property-name1:property-val1,property-name2:property-val2**

Use this query-parameter to get the filtered set of vxlanvlanmap resources configured on Citrix ADC. Filtering can be done on any of the properties of the resource.

view

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/vxlanvlanmap?**view=summary**

Note: By default, the retrieved results are displayed in detail view (?view=detail).

pagination

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/vxlanvlanmap?**pagesize=#no;pageno=#no**

Use this query-parameter to get the vxlanvlanmap resources in chunks.

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "vxlanvlanmap": [ {

"name":<String_value>,



}]}
```

get

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/vxlanvlanmap/name_value<String>

Query-parameters:

attrs

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/vxlanvlanmap/name_value<String>?**attrs=property-name1,property-name2**

Use this query parameter to specify the resource details that you want to retrieve.

view

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/vxlanvlanmap/name_value<String>?**view=summary**

Note: By default, the retrieved results are displayed in detail view (?view=detail).

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "vxlanvlanmap": [ {

"name":<String_value>,



}]}
```

count

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/vxlanvlanmap?**count=yes**

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "vxlanvlanmap": [ { "__count": "#no"} ] }
```

vxlanvlanmap_binding

Binding object which returns the resources bound to vxlanvlanmap.

Properties

(click to see Operations)

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
name	<String>	Read-write	Name of the mapping table. Minimum length = 1
vxlanvlanmap_vxlan_binding	<vxlanvlanmap_vxlan_binding[]>	Read-only	vxlan that can be bound to vxlanvlanmap.

Operations

(click to see Properties)

GET| GET (ALL)

Some options that you can use for each operations:

- **Getting warnings in response:**NITRO allows you to get warnings in an operation by specifying the "warning" query parameter as "yes". For example, to get warnings while connecting to the Citrix ADC appliance, the URL is as follows:

  http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/login?warning=yes

  If any, the warnings are displayed in the response payload with the HTTP code "209 X-NITRO-WARNING".

- **Authenticated access for individual NITRO operations:**NITRO allows you to logon to the Citrix ADC appliance to perform individual operations. You can use this option instead of creating a NITRO session (using the login object) and then using that session to perform all operations,

  To do this, you must specify the username and password in the request header of the NITRO request as follows:

  X-NITRO-USER:<username>

  X-NITRO-PASS:<password>

  Note: In such cases, make sure that the request header DOES not include the following:

  Cookie:NITRO_AUTH_TOKEN=<tokenvalue>

`*Note:`*

Mandatory parameters are marked in redand placeholder content is marked in <green>.

get

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/vxlanvlanmap_binding/name_value<String>

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "vxlanvlanmap_binding": [ {

"name":<String_value>,

"vxlanvlanmap_vxlan_binding":<vxlanvlanmap_vxlan_binding[]_value>

}]}
```

get (all)

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/vxlanvlanmap_binding

Query-parameters:

bulkbindings

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/vxlanvlanmap_binding?**bulkbindings=yes**

NITRO allows you to fetch bindings in bulk.

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "vxlanvlanmap_binding": [ {

"name":<String_value>,

"vxlanvlanmap_vxlan_binding":<vxlanvlanmap_vxlan_binding[]_value>

}]}
```

vxlanvlanmap_vxlan_binding

Binding object showing the vxlan that can be bound to vxlanvlanmap.

Properties

(click to see Operations)

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
name	<String>	Read-write	Name of the mapping table. Minimum length = 1
vxlan	<Double>	Read-write	The VXLAN assigned to the vlan inside the cloud. Minimum value = 1 Maximum value = 16777215
vlan	<String[]>	Read-write	The vlan id or the range of vlan ids in the on-premise network. Minimum length = 1 Maximum length = 4094

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
__count	<Double>	Read-write	count parameter

Operations

(click to see Properties)

ADD:| DELETE:| GET| GET (ALL)| COUNT

Some options that you can use for each operations:

- **Getting warnings in response:**NITRO allows you to get warnings in an operation by specifying the "warning" query parameter as "yes". For example, to get warnings while connecting to the Citrix ADC appliance, the URL is as follows:

  http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/login?warning=yes

  If any, the warnings are displayed in the response payload with the HTTP code "209 X-NITRO-WARNING".
- **Authenticated access for individual NITRO operations:**NITRO allows you to logon to the Citrix ADC appliance to perform individual operations. You can use this option instead of creating a NITRO session (using the login object) and then using that session to perform all operations,

  To do this, you must specify the username and password in the request header of the NITRO request as follows:

  X-NITRO-USER:<username>

  X-NITRO-PASS:<password>

  Note: In such cases, make sure that the request header DOES not include the following:

  Cookie:NITRO_AUTH_TOKEN=<tokenvalue>

`*Note:`*

Mandatory parameters are marked in redand placeholder content is marked in <green>.

add

URL:http://<Citrix-ADC-IP-address(NSIP)/nitro/v1/config/vxlanvlanmap_vxlan_binding

HTTP Method:PUT

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Content-Type:application/json

Request Payload:

```
{

"vxlanvlanmap_vxlan_binding":{

<b>"name":<String_value>,

</b>"vxlan":<Double_value>,

"vlan":<String[]_value>

}}
```

Response:

HTTP Status Code on Success: 201 Created HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error

delete

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/vxlanvlanmap_vxlan_binding/name_value<String>

Query-parameters:

args

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/vxlanvlanmap_vxlan_binding/name_value<String>?**args=vxlan:<Double_value>**

HTTP Method:DELETE

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue>

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error

get

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/vxlanvlanmap_vxlan_binding/name_value<String>

Query-parameters:

filter

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/vxlanvlanmap_vxlan_binding/name_value<String>?**filter=property-name1:property-value1,property-name2:property-value2**

Use this query-parameter to get the filtered set of vxlanvlanmap_vxlan_binding resources configured on Citrix ADC. Filtering can be done on any of the properties of the resource.

pagination

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/vxlanvlanmap_vxlan_binding/name_value<String>?**pagesize=#no;pageno=#no**

Use this query-parameter to get the vxlanvlanmap_vxlan_binding resources in chunks.

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "vxlanvlanmap_vxlan_binding": [ {

"name":<String_value>,

"vxlan":<Double_value>,

"vlan":<String[]_value>

}]}
```

get (all)

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/vxlanvlanmap_vxlan_binding

Query-parameters:

bulkbindings

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/vxlanvlanmap_vxlan_binding?**bulkbindings=yes**

NITRO allows you to fetch bindings in bulk.

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "vxlanvlanmap_vxlan_binding": [ {

"name":<String_value>,

"vxlan":<Double_value>,

"vlan":<String[]_value>

}]}
```

count

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/vxlanvlanmap_vxlan_binding/name_value<String>?**count=yes**

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{"vxlanvlanmap_vxlan_binding": [ { "__count": "#no"} ] }
```

vxlan_binding

Binding object which returns the resources bound to vxlan.

Properties

(click to see Operations)

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
id	<Double>	Read-write	A positive integer, which is also called VXLAN Network Identifier (VNI), that uniquely identifies a VXLAN. Minimum value = 1 Maximum value = 16777215
vxlan_iptunnel_binding	<vxlan_iptunnel_binding[]>	Read-only	iptunnel that can be bound to vxlan.
vxlan_nsip_binding	<vxlan_nsip_binding[]>	Read-only	nsip that can be bound to vxlan.
vxlan_srcip_binding	<vxlan_srcip_binding[]>	Read-only	srcip that can be bound to vxlan.
vxlan_nsip6_binding	<vxlan_nsip6_binding[]>	Read-only	nsip6 that can be bound to vxlan.

Operations

(click to see Properties)

GET| GET (ALL)

Some options that you can use for each operations:

- **Getting warnings in response:**NITRO allows you to get warnings in an operation by specifying the "warning" query parameter as "yes". For example, to get warnings while connecting to the Citrix ADC appliance, the URL is as follows:

  http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/login?warning=yes

If any, the warnings are displayed in the response payload with the HTTP code "209 X-NITRO-WARNING".

- **Authenticated access for individual NITRO operations:**NITRO allows you to logon to the Citrix ADC appliance to perform individual operations. You can use this option instead of creating a NITRO session (using the login object) and then using that session to perform all operations,

  To do this, you must specify the username and password in the request header of the NITRO request as follows:

  X-NITRO-USER:<username>

  X-NITRO-PASS:<password>

  Note: In such cases, make sure that the request header DOES not include the following:

  Cookie:NITRO_AUTH_TOKEN=<tokenvalue>

`*Note:`*

Mandatory parameters are marked in redand placeholder content is marked in <green>.

get

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/vxlan_binding/id_value<Double>

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "vxlan_binding": [ {

"id":<Double_value>,

"vxlan_nsip6_binding":<vxlan_nsip6_binding[]_value>,

"vxlan_nsip_binding":<vxlan_nsip_binding[]_value>,

"vxlan_iptunnel_binding":<vxlan_iptunnel_binding[]_value>,

"vxlan_srcip_binding":<vxlan_srcip_binding[]_value>

}]}
```

get (all)

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/vxlan_binding

Query-parameters:

bulkbindings

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/vxlan_binding?**bulkbindings=yes**

NITRO allows you to fetch bindings in bulk.

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "vxlan_binding": [ {

"id":<Double_value>,

"vxlan_nsip6_binding":<vxlan_nsip6_binding[]_value>,

"vxlan_nsip_binding":<vxlan_nsip_binding[]_value>,

"vxlan_iptunnel_binding":<vxlan_iptunnel_binding[]_value>,

"vxlan_srcip_binding":<vxlan_srcip_binding[]_value>

}]}
```

vxlan_iptunnel_binding

Binding object showing the iptunnel that can be bound to vxlan.

Properties

(click to see Operations)

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
id	<Double>	Read-write	A positive integer, which is also called VXLAN Network Identifier (VNI), that uniquely identifies a VXLAN. Minimum value = 1 Maximum value = 16777215
tunnel	<String>	Read-write	Specifies the name of the configured tunnel to be

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
			associated with this VXLAN.
__count	<Double>	Read-write	count parameter

Operations

(click to see Properties)

GET| GET (ALL)| COUNT

Some options that you can use for each operations:

- **Getting warnings in response:**NITRO allows you to get warnings in an operation by specifying the "warning" query parameter as "yes". For example, to get warnings while connecting to the Citrix ADC appliance, the URL is as follows:

  http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/login?warning=yes

  If any, the warnings are displayed in the response payload with the HTTP code "209 X-NITRO-WARNING".
- **Authenticated access for individual NITRO operations:**NITRO allows you to logon to the Citrix ADC appliance to perform individual operations. You can use this option instead of creating a NITRO session (using the login object) and then using that session to perform all operations,

  To do this, you must specify the username and password in the request header of the NITRO request as follows:

  X-NITRO-USER:<username>

  X-NITRO-PASS:<password>

  Note: In such cases, make sure that the request header DOES not include the following:

  Cookie:NITRO_AUTH_TOKEN=<tokenvalue>

`*Note:`*

Mandatory parameters are marked in redand placeholder content is marked in <green>.

get

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/vxlan_iptunnel_binding/id_value<Double>

Query-parameters:

filter

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/vxlan_iptunnel_binding/id_value<Double>?**filter=property-name1:property-value1,property-name2:property-value2**

Use this query-parameter to get the filtered set of vxlan_iptunnel_binding resources configured on Citrix ADC. Filtering can be done on any of the properties of the resource.

pagination

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/vxlan_iptunnel_binding/id_value<Double>?**pagesize=#no;pageno=#no**

Use this query-parameter to get the vxlan_iptunnel_binding resources in chunks.

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "vxlan_iptunnel_binding": [ {

"id":<Double_value>,

"tunnel":<String_value>
```

```
}]}
```

get (all)

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/vxlan_iptunnel_binding

Query-parameters:

bulkbindings

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/vxlan_iptunnel_binding?**bulkbindings=yes**

NITRO allows you to fetch bindings in bulk.

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "vxlan_iptunnel_binding": [ {

"id":<Double_value>,

"tunnel":<String_value>

}]}
```

count

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/vxlan_iptunnel_binding/id_value<Double>?**count=yes**

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{"vxlan_iptunnel_binding": [ { "__count": "#no"} ] }
```

vxlan_nsip6_binding

Binding object showing the nsip6 that can be bound to vxlan.

Properties

(click to see Operations)

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
id	<Double>	Read-write	A positive integer, which is also called VXLAN Network Identifier (VNI), that uniquely identifies a VXLAN. Minimum value = 1

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
			Maximum value = 16777215
netmask	<String>	Read-write	Subnet mask for the network address defined for this VXLAN. Minimum length = 1
ipaddress	<String>	Read-write	The IP address assigned to the VXLAN.
__count	<Double>	Read-write	count parameter

Operations

(click to see Properties)

ADD:| DELETE:| GET| GET (ALL)| COUNT

Some options that you can use for each operations:

- **Getting warnings in response:**NITRO allows you to get warnings in an operation by specifying the "warning" query parameter as "yes". For example, to get warnings while connecting to the Citrix ADC appliance, the URL is as follows:

  http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/login?warning=yes

  If any, the warnings are displayed in the response payload with the HTTP code "209 X-NITRO-WARNING".

- **Authenticated access for individual NITRO operations:**NITRO allows you to logon to the Citrix ADC appliance to perform individual operations. You can use this option instead of creating a NITRO session (using the login object) and then using that session to perform all operations,

  To do this, you must specify the username and password in the request header of the NITRO request as follows:

  X-NITRO-USER:<username>

X-NITRO-PASS:<password>

Note: In such cases, make sure that the request header DOES not include the following:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue>

`*Note:`*

Mandatory parameters are marked in redand placeholder content is marked in <green>.

add

URL:http://<Citrix-ADC-IP-address(NSIP)/nitro/v1/config/vxlan_nsip6_binding

HTTP Method:PUT

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Content-Type:application/json

Request Payload:

```
{

"vxlan_nsip6_binding":{

<b>"id":<Double_value>,

</b>"ipaddress":<String_value>,

"netmask":<String_value>

}}
```

Response:

HTTP Status Code on Success: 201 Created HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error

delete

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/vxlan_nsip6_binding/id_value<Double>

Query-parameters:

args

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/vxlan_nsip6_binding/id_value<Double>?**args=ipaddress:<String_value>,netmask:<String_value>**

HTTP Method:DELETE

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue>

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error

get

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/vxlan_nsip6_binding/id_value<Double>

Query-parameters:

filter

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/vxlan_nsip6_binding/id_value<Double>?**filter=property-name1:property-value1,property-name2:property-value2**

Use this query-parameter to get the filtered set of vxlan_nsip6_binding resources configured on Citrix ADC. Filtering can be done on any of the properties of the resource.

pagination

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/vxlan_nsip6_binding/id_value<Double>?**pagesize=#no;pageno=#no**

Use this query-parameter to get the vxlan_nsip6_binding resources in chunks.

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "vxlan_nsip6_binding": [ {

"id":<Double_value>,

"netmask":<String_value>,

"ipaddress":<String_value>

}]}
```

get (all)

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/vxlan_nsip6_binding

Query-parameters:

bulkbindings

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/vxlan_nsip6_binding?**bulkbindings=yes**

NITRO allows you to fetch bindings in bulk.

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "vxlan_nsip6_binding": [ {

"id":<Double_value>,

"netmask":<String_value>,

"ipaddress":<String_value>

}]}
```

count

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/vxlan_nsip6_binding/id_value<Double>?**count=yes**

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{"vxlan_nsip6_binding": [ { "__count": "#no"} ] }
```

vxlan_nsip_binding

Binding object showing the nsip that can be bound to vxlan.

Properties

(click to see Operations)

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
id	<Double>	Read-write	A positive integer, which is also called VXLAN Network Identifier (VNI), that uniquely identifies a VXLAN. Minimum value = 1 Maximum value = 16777215
netmask	<String>	Read-write	Subnet mask for the network address defined for this VXLAN.
ipaddress	<String>	Read-write	The IP address assigned to the VXLAN.
__count	<Double>	Read-write	count parameter

Operations

(click to see Properties)

ADD:| DELETE:| GET| GET (ALL)| COUNT

Some options that you can use for each operations:

- **Getting warnings in response:**NITRO allows you to get warnings in an operation by specifying the "warning" query parameter as "yes". For example, to get warnings while connecting to the Citrix ADC appliance, the URL is as follows:

  http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/login?warning=yes

  If any, the warnings are displayed in the response payload with the HTTP code "209 X-NITRO-WARNING".

- **Authenticated access for individual NITRO operations:**NITRO allows you to logon to the Citrix ADC appliance to perform individual operations. You can use this option instead of creating a NITRO session (using the login object) and then using that session to perform all operations,

  To do this, you must specify the username and password in the request header of the NITRO request as follows:

  X-NITRO-USER:<username>

  X-NITRO-PASS:<password>

  Note: In such cases, make sure that the request header DOES not include the following:

  Cookie:NITRO_AUTH_TOKEN=<tokenvalue>

`*Note:`*

Mandatory parameters are marked in redand placeholder content is marked in <green>.

add

URL:http://<Citrix-ADC-IP-address(NSIP)/nitro/v1/config/vxlan_nsip_binding

HTTP Method:PUT

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Content-Type:application/json

Request Payload:

```
{

"vxlan_nsip_binding":{

<b>"id":<Double_value>,

</b>"ipaddress":<String_value>,

"netmask":<String_value>

}}
```

Response:

HTTP Status Code on Success: 201 Created HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error

delete

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/vxlan_nsip_binding/id_value<Double>

Query-parameters:

args

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/vxlan_nsip_binding/id_value<Double>?**args=ipaddress:<String_value>,netmask:<String_value>**

HTTP Method:DELETE

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue>

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error

get

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/vxlan_nsip_binding/id_value<Double>

Query-parameters:

filter

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/vxlan_nsip_binding/id_value<Double>?**filter=property-name1:property-value1,property-name2:property-value2**

Use this query-parameter to get the filtered set of vxlan_nsip_binding resources configured on Citrix ADC. Filtering can be done on any of the properties of the resource.

pagination

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/vxlan_nsip_binding/id_value<Double>?**pagesize=#no;pageno=#no**

Use this query-parameter to get the vxlan_nsip_binding resources in chunks.

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "vxlan_nsip_binding": [ {

"id":<Double_value>,

"netmask":<String_value>,
```

```
"ipaddress":<String_value>

}]}
```

get (all)

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/vxlan_nsip_binding

Query-parameters:

bulkbindings

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/vxlan_nsip_binding?**bulkbindings=yes**

NITRO allows you to fetch bindings in bulk.

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "vxlan_nsip_binding": [ {

"id":<Double_value>,

"netmask":<String_value>,

"ipaddress":<String_value>

}]}
```

count

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/vxlan_nsip_binding/id_value<Double>?**count=yes**

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{"vxlan_nsip_binding": [ { "__count": "#no"} ] }
```

vxlan_srcip_binding

Binding object showing the srcip that can be bound to vxlan.

Properties

(click to see Operations)

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
id	<Double>	Read-write	A positive integer, which is also called VXLAN Network Identifier (VNI), that uniquely identifies a VXLAN.

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
			Minimum value = 1 Maximum value = 16777215
srcip	<String>	Read-write	The source IP address to use in outgoing vxlan packets. Minimum length = 1
__count	<Double>	Read-write	count parameter

Operations

(click to see Properties)

ADD:| DELETE:| GET| GET (ALL)| COUNT

Some options that you can use for each operations:

- **Getting warnings in response:**NITRO allows you to get warnings in an operation by specifying the "warning" query parameter as "yes". For example, to get warnings while connecting to the Citrix ADC appliance, the URL is as follows:

  http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/login?warning=yes

  If any, the warnings are displayed in the response payload with the HTTP code "209 X-NITRO-WARNING".

- **Authenticated access for individual NITRO operations:**NITRO allows you to logon to the Citrix ADC appliance to perform individual operations. You can use this option instead of creating a NITRO session (using the login object) and then using that session to perform all operations,

  To do this, you must specify the username and password in the request header of the NITRO request as follows:

  X-NITRO-USER:<username>

  X-NITRO-PASS:<password>

Note: In such cases, make sure that the request header DOES not include the following:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue>

`*Note:` *

Mandatory parameters are marked in redand placeholder content is marked in <green>.

add

URL:http://<Citrix-ADC-IP-address(NSIP)/nitro/v1/config/vxlan_srcip_binding

HTTP Method:PUT

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Content-Type:application/json

Request Payload:

```
{

"vxlan_srcip_binding":{

<b>"id":<Double_value>,

</b>"srcip":<String_value>

}}
```

Response:

HTTP Status Code on Success: 201 Created HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error

delete

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/vxlan_srcip_binding/id_value<Double>

Query-parameters:

args

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/vxlan_srcip_binding/id_value<Double>?**args=srcip:<String_value>**

HTTP Method:DELETE

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue>

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error

get

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/vxlan_srcip_binding/id_value<Double>

Query-parameters:

filter

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/vxlan_srcip_binding/id_value<Double>?**filter=property-name1:property-value1,property-name2:property-value2**

Use this query-parameter to get the filtered set of vxlan_srcip_binding resources configured on Citrix ADC. Filtering can be done on any of the properties of the resource.

pagination

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/vxlan_srcip_binding/id_value<Double>?**pagesize=#no;pageno=#no**

Use this query-parameter to get the vxlan_srcip_binding resources in chunks.

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "vxlan_srcip_binding": [ {

"id":<Double_value>,

"srcip":<String_value>

}]}
```

get (all)

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/vxlan_srcip_binding

Query-parameters:

bulkbindings

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/vxlan_srcip_binding?**bulkbindings=yes**

NITRO allows you to fetch bindings in bulk.

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "vxlan_srcip_binding": [ {

"id":<Double_value>,

"srcip":<String_value>

}]}
```

count

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/vxlan_srcip_binding/id_value<Double>?**count=yes**

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{"vxlan_srcip_binding": [ { "__count": "#no"} ] }
```

NS

NS

System/Global level configuration.

NAME	DESCRIPTION
migrate	
nsacl	Configuration for ACL entry
nsacl6	Configuration for ACL6 entry
nsacls	Configuration for ACL entry
nsacls6	Configuration for ACL6 entry
nsappflowcollector	Configuration for appflowCollector
nsappflowparam	Configuration for appflowParam
nsaptlicense	Configuration for aptlicense
nsassignment	Configuration for assignment
nscapacity	Configuration for capacity
nscentralmanagementserver	Configuration for centralmanagementserver
nsconfig	Configuration for system config
nsconnectiontable	Configuration for TCP/IP connection table
nsconsoleloginprompt	Configuration for console prompt
nscqaparam	Configuration for cqaparam
nsdhcpip	Configuration for dhcp IP
nsdhcpparams	Configuration for DHCP parameters
nsdiameter	Configuration for Diameter Parameters

NAME	DESCRIPTION
nsencryptionkey	Configuration for encryption key
nsencryptionparams	Configuration for default encryption parameters
nsevents	Configuration for events
nsextension	Configuration for Extension
nsextension_binding	Binding object showing the resources that can be bound to nsextension
nsextension_extensionfunction_binding	Binding object showing the extensionfunction that can be bound to nsextension
nsfeature	Configuration for feature
nshardware	Configuration for hardware
nshmackey	Configuration for HMAC key
nshostname	Configuration for host name
nshttpparam	Configuration for HTTP parameter
nshttpprofile	Configuration for HTTP profile
nsicapprofile	Configuration for ICAP profile
nsip	Configuration for ip
nsip6	Configuration for ip6
nslicense	Configuration for license
nslicenseproxyserver	Configuration for licenseproxyserver
nslicenseserver	Configuration for licenseserver
nslicenseserverpool	Configuration for licenseserver
nslimitidentifier	Configuration for limit Indetifier

NAME	DESCRIPTION
nslimitidentifier_binding	Binding object showing the resources that can be bound to nslimitidentifier
nslimitidentifier_nslimitsessions_binding	Binding object showing the nslimitsessions that can be bound to nslimitidentifier
nslimitselector	Configuration for limit selector
nslimitsessions	Configuration for limit sessions
nsmode	Configuration for ns mode
nsparam	Configuration for Citrix ADC parameters
nspartition	Configuration for admin partition
nspartition_binding	Binding object showing the resources that can be bound to nspartition
nspartition_bridgegroup_binding	Binding object showing the bridgegroup that can be bound to nspartition
nspartition_vlan_binding	Binding object showing the vlan that can be bound to nspartition
nspartition_vxlan_binding	Binding object showing the vxlan that can be bound to nspartition
nspartitionmac	Configuration for partition MACs
nspbr	Configuration for Policy Based Routing(PBR) entry
nspbr6	Configuration for PBR6 entry
nspbrs	Configuration for Policy based routing
nsratecontrol	Configuration for rate control
nsrollbackcmd	Configuration for Generate rollback commands
nsrpcnode	Configuration for rpc node

NAME	DESCRIPTION
nsrunningconfig	Configuration for running configuration
nssavedconfig	Configuration for last saved configuration
nsservicefunction	Configuration for service Function
nsservicepath	Configuration for service Chain
nsservicepath_binding	Binding object showing the resources that can be bound to nsservicepath
nsservicepath_nsservicefunction_binding	Binding object showing the nsservicefunction that can be bound to nsservicepath
nssimpleacl	Configuration for simple ACL
nssimpleacl6	Configuration for simple ACL6
nssourceroutecachetable	Configuration for Source IP Mac Cache Table.
nsspparams	Configuration for Surge Protection parameter
nsstats	Configuration for stats
nssurgeq	Configuration for surge queue
nstcpbufparam	Configuration for tcp buffer parameter
nstcpparam	Configuration for tcp parameters
nstcpprofile	Configuration for TCP profile
nstimeout	Configuration for timeout
nstimer	Configuration for Timer
nstimer_autoscalepolicy_binding	Binding object showing the autoscalepolicy that can be bound to nstimer
nstimer_binding	Binding object showing the resources that can be bound to nstimer

NAME	DESCRIPTION
nstimezone	
nstrafficdomain	Configuration for Traffic Domain
nstrafficdomain_binding	Binding object showing the resources that can be bound to nstrafficdomain
nstrafficdomain_bridgegroup_binding	Binding object showing the bridgegroup that can be bound to nstrafficdomain
nstrafficdomain_vlan_binding	Binding object showing the vlan that can be bound to nstrafficdomain
nstrafficdomain_vxlan_binding	Binding object showing the vxlan that can be bound to nstrafficdomain
nsvariable	Configuration for variable
nsversion	Configuration for version
nsvpxparam	Configuration for "VPX"
nsweblogparam	Configuration for Web log parameters
nsxmlnamespace	Configuration for XML namespace
reboot	
shutdown	

nsacl

Configuration for ACL entry resource.

Properties

(click to see Operations)

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
aclname	<String>	Read-write	Name for the extended ACL rule.

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
			Must begin with an ASCII alphabetic or underscore (_) character, and must contain only ASCII alphanumeric, underscore, hash (#), period (.), space, colon (:), at (@), equals (=), and hyphen (-) characters. Minimum length = 1
aclaction	<String>	Read-write	Action to perform on incoming IPv4 packets that match the extended ACL rule. Available settings function as follows: * ALLOW - The Citrix ADC processes the packet. * BRIDGE - The Citrix ADC bridges the packet to the destination without processing it. * DENY - The Citrix ADC drops the packet.

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
			Possible values = BRIDGE, DENY, ALLOW
td	<Double>	Read-write	Integer value that uniquely identifies the traffic domain in which you want to configure the entity. If you do not specify an ID, the entity becomes part of the default traffic domain, which has an ID of 0. Minimum value = 0 Maximum value = 4094
srcip	<Boolean>	Read-write	IP address or range of IP addresses to match against the source IP address of an incoming IPv4 packet. In the command line interface, separate the range with a hyphen. For example: 10.102.29.30-10.102.29.189.
srcipop	<String>	Read-write	Either the equals (=) or does not equal (!=) logical operator. Possible values = =, !=, EQ, NEQ

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
srcipval	<String>	Read-write	IP address or range of IP addresses to match against the source IP address of an incoming IPv4 packet. In the command line interface, separate the range with a hyphen. For example:10.102.29.30-10.102.29.189.
srcport	<Boolean>	Read-write	Port number or range of port numbers to match against the source port number of an incoming IPv4 packet. In the command line interface, separate the range with a hyphen. For example: 40-90.
srcportop	<String>	Read-write	Either the equals (=) or does not equal (!=) logical operator. Possible values = =, !=, EQ, NEQ
srcportval	<String>	Read-write	Port number or range of port numbers to match against the source port number of an incoming IPv4 packet. In the command line interface, separate

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
			the range with a hyphen. For example: 40-90. Maximum length = 65535
destip	<Boolean>	Read-write	IP address or range of IP addresses to match against the destination IP address of an incoming IPv4 packet. In the command line interface, separate the range with a hyphen. For example: 10.102.29.30-10.102.29.189.
destipop	<String>	Read-write	Either the equals (=) or does not equal (!=) logical operator. Possible values = =, !=, EQ, NEQ
destipval	<String>	Read-write	IP address or range of IP addresses to match against the destination IP address of an incoming IPv4 packet. In the command line interface, separate the range with a hyphen. For example:

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
			10.102.29.30-10.102.29.189.
destport	<Boolean>	Read-write	Port number or range of port numbers to match against the destination port number of an incoming IPv4 packet. In the command line interface, separate the range with a hyphen. For example: 40-90. Note: The destination port can be specified only for TCP and UDP protocols.
destportop	<String>	Read-write	Either the equals (=) or does not equal (!=) logical operator. Possible values = =, !=, EQ, NEQ
destportval	<String>	Read-write	Port number or range of port numbers to match against the destination port number of an incoming IPv4 packet. In the command line interface, separate the range with a

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
			hyphen. For example: 40-90. Note: The destination port can be specified only for TCP and UDP protocols. Maximum length = 65535
ttl	<Double>	Read-write	Number of seconds, in multiples of four, after which the extended ACL rule expires. If you do not want the extended ACL rule to expire, do not specify a TTL value. Minimum value = 1 Maximum value = 2147483647
srcmac	<String>	Read-write	MAC address to match against the source MAC address of an incoming IPv4 packet.
srcmacmask	<String>	Read-write	Used to define range of Source MAC address. It takes string of 0 and 1, 0s are for exact match and 1s for wildcard. For matching first 3 bytes of MAC address,

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
			srcMacMask value "000000111111". . Default value: "000000000000"
protocol	<String>	Read-write	Protocol to match against the protocol of an incoming IPv4 packet. Possible values = ICMP, IGMP, TCP, EGP, IGP, ARGUS, UDP, RDP, RSVP, EIGRP, L2TP, ISIS, GGP, IPoverIP, ST, CBT, BBN-RCC-M, NVP-II, PUP, EMCON, XNET, CHAOS, MUX, DCN-MEAS, HMP, PRM, XNS-IDP, TRUNK-1, TRUNK-2, LEAF-1, LEAF-2, IRTP, ISO-TP4, NETBLT, MFE-NSP, MERIT-INP, SEP, 3PC, IDPR, XTP, DDP, IDPR-CMTP, TP++, IL, IPv6, SDRP, IPv6-Route, IPv6-Frag, IDRP, GRE, MHRP, BNA, ESP, AH, I-NLSP, SWIPE, NARP, MOBILE, TLSP, SKIP, ICMPV6,

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
			IPv6-NoNx, IPv6-Opts, Any-Host-Internal-Protocol, CFTP, Any-Local-Network, SAT-EXPAK, KRYPTOLAN, RVD, IPPC, Any-Distributed-File-System, TFTP, VISA, IPCV, CPNX, CPHB, WSN, PVP, BR-SAT-MO, SUN-ND, WB-MON, WB-EXPAK, ISO-IP, VMTP, SECURE-VM, VINES, TTP, NSFNET-IG, DGP, TCF, OSPFIGP, Sprite-RP, LARP, MTP, AX.25, IPIP, MICP, SCC-SP, ETHERIP, Any-Private-Encryption-Scheme, GMTP, IFMP, PNNI, PIM, ARIS, SCPS, QNX, A/N, IPComp, SNP, Compaq-Pe, IPX-in-IP, VRRP, PGM, Any-0-Hop-Protocol, ENCAP, DDX, IATP, STP, SRP, UTI, SMP, SM, PTP, FIRE, CRTP, CRUDP,

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
			SSCOPMCE, IPLT, SPS, PIPE, SCTP, FC, RSVP-E2E-IGNORE, Mobility-Header, UDPLite
protocolnumber	<Double>	Read-write	Protocol to match against the protocol of an incoming IPv4 packet. Minimum value = 1 Maximum value = 255
vlan	<Double>	Read-write	ID of the VLAN. The Citrix ADC applies the ACL rule only to the incoming packets of the specified VLAN. If you do not specify a VLAN ID, the appliance applies the ACL rule to the incoming packets on all VLANs. Minimum value = 1 Maximum value = 4094
vxlan	<Double>	Read-write	ID of the VXLAN. The Citrix ADC applies the ACL rule only to the incoming packets of the specified VXLAN. If you do not specify a VXLAN ID, the appliance applies the

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
			ACL rule to the incoming packets on all VXLANs. Minimum value = 1 Maximum value = 16777215
Interface	<String>	Read-write	ID of an interface. The Citrix ADC applies the ACL rule only to the incoming packets from the specified interface. If you do not specify any value, the appliance applies the ACL rule to the incoming packets of all interfaces.
established	<Boolean>	Read-write	Allow only incoming TCP packets that have the ACK or RST bit set, if the action set for the ACL rule is ALLOW and these packets match the other conditions in the ACL rule.
icmptype	<Double>	Read-write	ICMP Message type to match against the message type of an incoming ICMP packet. For example, to block DESTINATION UNREACHABLE messages, you must

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
			specify 3 as the ICMP type. Note: This parameter can be specified only for the ICMP protocol. Minimum value = 0 Maximum value = 65536
icmpcode	<Double>	Read-write	Code of a particular ICMP message type to match against the ICMP code of an incoming ICMP packet. For example, to block DESTINATION HOST UNREACHABLE messages, specify 3 as the ICMP type and 1 as the ICMP code. If you set this parameter, you must set the ICMP Type parameter. Minimum value = 0 Maximum value = 65536
priority	<Double>	Read-write	Priority for the extended ACL rule that determines the order in which it is evaluated relative to

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
			the other extended ACL rules. If you do not specify priorities while creating extended ACL rules, the ACL rules are evaluated in the order in which they are created. Minimum value = 1 Maximum value = 100000
state	<String>	Read-write	Enable or disable the extended ACL rule. After you apply the extended ACL rules, the Citrix ADC compares incoming packets against the enabled extended ACL rules. Default value: ENABLED Possible values = ENABLED, DISABLED
logstate	<String>	Read-write	Enable or disable logging of events related to the extended ACL rule. The log messages are stored in the configured syslog or auditlog server.

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
			Default value: DISABLED Possible values = ENABLED, DISABLED
ratelimit	<Double>	Read-write	Maximum number of log messages to be generated per second. If you set this parameter, you must enable the Log State parameter. Default value: 100 Minimum value = 1 Maximum value = 10000
type	<String>	Read-write	Type of the acl ,default will be CLASSIC. Available options as follows: * CLASSIC - specifies the regular extended acls. * DFD - cluster specific acls,specifies hashmethod for steering of the packet in cluster .

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
			Default value: CLASSIC Possible values = CLASSIC, DFD
dfdhash	<String>	Read-write	Specifies the type hashmethod to be applied, to steer the packet to the FP of the packet. Possible values = SIP-SPORT-DIP-DPORT, SIP, DIP, SIP-DIP, SIP-SPORT, DIP-DPORT
stateful	<String>	Read-write	If stateful option is enabled, transparent sessions are created for the traffic hitting this ACL and not hitting any other features like LB, INAT etc. . Default value: NO Possible values = YES, NO
newname	<String>	Read-write	New name for the extended ACL rule. Must begin with an ASCII alphabetic or underscore (_) character, and must contain only ASCII alphanumeric, underscore, hash

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
			(#), period (.), space, colon (:), at (@), equals (=), and hyphen (-) characters. Minimum length = 1
hits	<Double>	Read-only	The hits of this ACL.
kernelstate	<String>	Read-only	The commit status of the ACL. Default value: NOTAPPLIED Possible values = APPLIED, NOTAPPLIED, RE-APPLY, SFAPPLIED, SFNOTAPPLIED, SFAPPLIED61, SFNOTAPPLIED61
aclassociate	<String[]>	Read-only	ACL linked. Possible values = NAT, FORWARDINGSESSION, NAT64, LSN
__count	<Double>	Read-only	count parameter

Operations

(click to see Properties)

ADD| DELETE| UPDATE| UNSET| ENABLE| DISABLE| RENAME| GET (ALL)| GET| COUNT

Some options that you can use for each operations:

- **Getting warnings in response:**NITRO allows you to get warnings in an operation by specifying the "warning" query parameter as "yes". For example, to get warnings while connecting to the Citrix ADC appliance, the URL is as follows:

  http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/login?warning=yes

  If any, the warnings are displayed in the response payload with the HTTP code "209 X-NITRO-WARNING".

- **Authenticated access for individual NITRO operations:**NITRO allows you to logon to the Citrix ADC appliance to perform individual operations. You can use this option instead of creating a NITRO session (using the login object) and then using that session to perform all operations,

  To do this, you must specify the username and password in the request header of the NITRO request as follows:

  X-NITRO-USER:<username>

  X-NITRO-PASS:<password>

  Note: In such cases, make sure that the request header DOES not include the following:

  Cookie:NITRO_AUTH_TOKEN=<tokenvalue>

`*Note:`*

Mandatory parameters are marked in redand placeholder content is marked in <green>.

add

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/nsacl

HTTP Method:POST

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Content-Type:application/json

Request Payload:

```
{"nsacl":{
```

```
<b>"aclname":<String_value>,

</b><b>"aclaction":<String_value>,

</b>"td":<Double_value>,

"srcip":<Boolean_value>,

"srcipop":<String_value>,

"srcipval":<String_value>,

"srcport":<Boolean_value>,

"srcportop":<String_value>,

"srcportval":<String_value>,

"destip":<Boolean_value>,

"destipop":<String_value>,

"destipval":<String_value>,

"destport":<Boolean_value>,

"destportop":<String_value>,

"destportval":<String_value>,

"ttl":<Double_value>,

"srcmac":<String_value>,

"srcmacmask":<String_value>,

"protocol":<String_value>,

"protocolnumber":<Double_value>,

"vlan":<Double_value>,

"vxlan":<Double_value>,
```

```
"Interface":<String_value>,

"established":<Boolean_value>,

"icmptype":<Double_value>,

"icmpcode":<Double_value>,

"priority":<Double_value>,

"state":<String_value>,

"logstate":<String_value>,

"ratelimit":<Double_value>,

"type":<String_value>,

"dfdhash":<String_value>,

"stateful":<String_value>

}}
```

Response:

HTTP Status Code on Success: 201 Created HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error

delete

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/nsacl/aclname_value<String>

HTTP Method:DELETE

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue>

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error

update

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/nsacl

HTTP Method:PUT

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Content-Type:application/json

Request Payload:

```
{"nsacl":{

<b>"aclname":<String_value>,

</b>"aclaction":<String_value>,

"srcip":<Boolean_value>,

"srcipop":<String_value>,

"srcipval":<String_value>,

"srcport":<Boolean_value>,

"srcportop":<String_value>,

"srcportval":<String_value>,

"destip":<Boolean_value>,

"destipop":<String_value>,

"destipval":<String_value>,

"destport":<Boolean_value>,
```

```
"destportop":<String_value>,

"destportval":<String_value>,

"srcmac":<String_value>,

"srcmacmask":<String_value>,

"protocol":<String_value>,

"protocolnumber":<Double_value>,

"icmptype":<Double_value>,

"icmpcode":<Double_value>,

"vlan":<Double_value>,

"vxlan":<Double_value>,

"Interface":<String_value>,

"priority":<Double_value>,

"logstate":<String_value>,

"ratelimit":<Double_value>,

"established":<Boolean_value>,

"dfdhash":<String_value>,

"stateful":<String_value>

}}
```

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error

unset

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/nsacl?**action=unset**

HTTP Method:POST

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Content-Type:application/json

Request Payload:

```
{"nsacl":{

<b>"aclname":<String_value>,

</b>"srcip":true,

"srcport":true,

"destip":true,

"destport":true,

"srcmac":true,

"srcmacmask":true,

"protocol":true,

"icmptype":true,

"icmpcode":true,

"vlan":true,

"vxlan":true,

"Interface":true,

"logstate":true,
```

```
"ratelimit":true,

"established":true,

"stateful":true,

"dfdhash":true

}}
```

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error

enable

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/nsacl?**action=enable**

HTTP Method:POST

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Content-Type:application/json

Request Payload:

```
{"nsacl":{

<b>"aclname":<String_value>

</b>}}
```

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error

disable

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/nsacl?**action=disable**

HTTP Method:POST

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Content-Type:application/json

Request Payload:

```
{"nsacl":{

<b>"aclname":<String_value>

</b>}}
```

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error

rename

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/nsacl?**action=rename**

HTTP Method:POST

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Content-Type:application/json

Request Payload:

```
{"nsacl":{

<b>"aclname":<String_value>,
```

```
</b><b>"newname":<String_value>

</b>}}
```

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error

get (all)

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/nsacl

Query-parameters:

args

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/nsacl?**args=aclname:<String_value>,type:<String_value>**

Use this query-parameter to get nsacl resources based on additional properties.

attrs

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/nsacl?**attrs=property-name1,property-name2**

Use this query parameter to specify the resource details that you want to retrieve.

filter

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/nsacl?**filter=property-name1:property-val1,property-name2:property-val2**

Use this query-parameter to get the filtered set of nsacl resources configured on Citrix ADC. Filtering can be done on any of the properties of the resource.

view

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/nsacl?**view=summary**

Note: By default, the retrieved results are displayed in detail view (?view=detail).

pagination

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/nsacl?**pagesize=#no;pageno=#no**

Use this query-parameter to get the nsacl resources in chunks.

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "nsacl": [ {

aclname:<String_value>,type:<String_value>"td":<Double_value>,

"aclaction":<String_value>,

"srcmac":<String_value>,

"srcmacmask":<String_value>,

"protocol":<String_value>,

"protocolnumber":<Double_value>,

"srcportval":<String_value>,

"srcportop":<String_value>,

"destportval":<String_value>,
```

```
"destportop":<String_value>,

"srcipval":<String_value>,

"srcipop":<String_value>,

"destipval":<String_value>,

"destipop":<String_value>,

"vlan":<Double_value>,

"vxlan":<Double_value>,

"state":<String_value>,

"ttl":<Double_value>,

"icmptype":<Double_value>,

"icmpcode":<Double_value>,

"Interface":<String_value>,

"hits":<Double_value>,

"established":<Boolean_value>,

"priority":<Double_value>,

"kernelstate":<String_value>,

"logstate":<String_value>,

"ratelimit":<Double_value>,

"aclassociate":<String[]_value>,

"dfdhash":<String_value>,

"stateful":<String_value>

}]}
```

get

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/nsacl/aclname_value<String>

Query-parameters:

attrs

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/nsacl/aclname_value<String>?**attrs=property-name1,property-name2**

Use this query parameter to specify the resource details that you want to retrieve.

view

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/nsacl/aclname_value<String>?**view=summary**

Note: By default, the retrieved results are displayed in detail view (?view=detail).

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "nsacl": [ {

aclname:<String_value>,type:<String_value>"td":<Double_value>,

"aclaction":<String_value>,
```

```
"srcmac":<String_value>,

"srcmacmask":<String_value>,

"protocol":<String_value>,

"protocolnumber":<Double_value>,

"srcportval":<String_value>,

"srcportop":<String_value>,

"destportval":<String_value>,

"destportop":<String_value>,

"srcipval":<String_value>,

"srcipop":<String_value>,

"destipval":<String_value>,

"destipop":<String_value>,

"vlan":<Double_value>,

"vxlan":<Double_value>,

"state":<String_value>,

"ttl":<Double_value>,

"icmptype":<Double_value>,

"icmpcode":<Double_value>,

"Interface":<String_value>,

"hits":<Double_value>,

"established":<Boolean_value>,

"priority":<Double_value>,
```

```
"kernelstate":<String_value>,

"logstate":<String_value>,

"ratelimit":<Double_value>,

"aclassociate":<String[]_value>,

"dfdhash":<String_value>,

"stateful":<String_value>

}]}
```

count

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/nsacl?**count=yes**

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "nsacl": [ { "__count": "#no"} ] }
```

nsacl6

Configuration for ACL6 entry resource.

Properties

(click to see Operations)

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
acl6name	<String>	Read-write	Name for the ACL6 rule. Must begin with an ASCII alphabetic or underscore (_) character, and must contain only ASCII alphanumeric, underscore, hash (#), period (.), space, colon (:), at (@), equals (=), and hyphen (-) characters. Minimum length = 1
acl6action	<String>	Read-write	Action to perform on the incoming IPv6 packets that match the ACL6 rule. Available settings function as follows: * ALLOW - The Citrix ADC processes the packet. * BRIDGE - The Citrix ADC bridges the packet to the destination without processing it.

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
			* DENY - The Citrix ADC drops the packet. Possible values = BRIDGE, DENY, ALLOW
td	<Double>	Read-write	Integer value that uniquely identifies the traffic domain in which you want to configure the entity. If you do not specify an ID, the entity becomes part of the default traffic domain, which has an ID of 0. Minimum value = 0 Maximum value = 4094
srcipv6	<Boolean>	Read-write	IP address or range of IP addresses to match against the source IP address of an incoming IPv6 packet. In the command line interface, separate the range with a hyphen.
srcipop	<String>	Read-write	Either the equals (=) or does not equal (!=) logical operator. Possible values = =, !=, EQ, NEQ

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
srcipv6val	<String>	Read-write	Source IPv6 address (range).
srcport	<Boolean>	Read-write	Port number or range of port numbers to match against the source port number of an incoming IPv6 packet. In the command line interface, separate the range with a hyphen. For example: 40-90. Note: The destination port can be specified only for TCP and UDP protocols.
srcportop	<String>	Read-write	Either the equals (=) or does not equal (!=) logical operator. Possible values = =, !=, EQ, NEQ
srcportval	<String>	Read-write	Source port (range). Maximum length = 65535
destipv6	<Boolean>	Read-write	IP address or range of IP addresses to match against the destination IP address of an incoming IPv6 packet. In the command line interface, separate

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
			the range with a hyphen.
destipop	<String>	Read-write	Either the equals (=) or does not equal (!=) logical operator. Possible values = =, !=, EQ, NEQ
destipv6val	<String>	Read-write	Destination IPv6 address (range).
destport	<Boolean>	Read-write	Port number or range of port numbers to match against the destination port number of an incoming IPv6 packet. In the command line interface, separate the range with a hyphen. For example: 40-90. Note: The destination port can be specified only for TCP and UDP protocols.
destportop	<String>	Read-write	Either the equals (=) or does not equal (!=) logical operator. Possible values = =, !=, EQ, NEQ
destportval	<String>	Read-write	Destination port (range).

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
			Maximum length = 65535
ttl	<Double>	Read-write	Time to expire this ACL6 (in seconds). Minimum value = 1 Maximum value = 2147483647
srcmac	<String>	Read-write	MAC address to match against the source MAC address of an incoming IPv6 packet.
srcmacmask	<String>	Read-write	Used to define range of Source MAC address. It takes string of 0 and 1, 0s are for exact match and 1s for wildcard. For matching first 3 bytes of MAC address, srcMacMask value "000000111111". . Default value: "000000000000"
protocol	<String>	Read-write	Protocol, identified by protocol name, to match against the protocol of an incoming IPv6 packet. Possible values = ICMPV6, TCP, UDP, ICMP, IGMP, EGP, IGP, ARGUS, RDP,

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
			RSVP, EIGRP, L2TP, ISIS, GGP, IPoverIP, ST, CBT, BBN-RCC-M, NVP-II, PUP, EMCON, XNET, CHAOS, MUX, DCN-MEAS, HMP, PRM, XNS-IDP, TRUNK-1, TRUNK-2, LEAF-1, LEAF-2, IRTP, ISO-TP4, NETBLT, MFE-NSP, MERIT-INP, SEP, 3PC, IDPR, XTP, DDP, IDPR-CMTP, TP++, IL, IPv6, SDRP, IPv6-Route, IPv6-Frag, IDRP, GRE, MHRP, BNA, ESP, AH, I-NLSP, SWIPE, NARP, MOBILE, TLSP, SKIP, IPv6-NoNx, IPv6-Opts, Any-Host-Internal-Protocol, CFTP, Any-Local-Network, SAT-EXPAK, KRYPTOLAN, RVD, IPPC, Any-Distributed-File-System, TFTP, VISA, IPCV, CPNX, CPHB, WSN, PVP, BR-SAT-MO, SUN-

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
			ND, WB-MON, WB-EXPAK, ISO-IP, VMTP, SECURE-VM, VINES, TTP, NSFNET-IG, DGP, TCF, OSPFIGP, Sprite-RP, LARP, MTP, AX.25, IPIP, MICP, SCC-SP, ETHERIP, Any-Private-Encryption-Scheme, GMTP, IFMP, PNNI, PIM, ARIS, SCPS, QNX, A/N, IPComp, SNP, Compaq-Pe, IPX-in-IP, VRRP, PGM, Any-0-Hop-Protocol, ENCAP, DDX, IATP, STP, SRP, UTI, SMP, SM, PTP, FIRE, CRTP, CRUDP, SSCOPMCE, IPLT, SPS, PIPE, SCTP, FC, RSVP-E2E-IGNORE, Mobility-Header, UDPLite
protocolnumber	<Double>	Read-write	Protocol, identified by protocol number, to match against the protocol of an incoming IPv6 packet. Minimum value = 1

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
			Maximum value = 255
vlan	<Double>	Read-write	ID of the VLAN. The Citrix ADC applies the ACL6 rule only to the incoming packets on the specified VLAN. If you do not specify a VLAN ID, the appliance applies the ACL6 rule to the incoming packets on all VLANs. Minimum value = 1 Maximum value = 4094
vxlan	<Double>	Read-write	ID of the VXLAN. The Citrix ADC applies the ACL6 rule only to the incoming packets on the specified VXLAN. If you do not specify a VXLAN ID, the appliance applies the ACL6 rule to the incoming packets on all VXLANs. Minimum value = 1 Maximum value = 16777215
Interface	<String>	Read-write	ID of an interface. The Citrix ADC applies the ACL6 rule only to the incoming packets

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
			from the specified interface. If you do not specify any value, the appliance applies the ACL6 rule to the incoming packets from all interfaces.
established	<Boolean>	Read-write	Allow only incoming TCP packets that have the ACK or RST bit set if the action set for the ACL6 rule is ALLOW and these packets match the other conditions in the ACL6 rule.
icmptype	<Double>	Read-write	ICMP Message type to match against the message type of an incoming IPv6 ICMP packet. For example, to block DESTINATION UNREACHABLE messages, you must specify 3 as the ICMP type. Note: This parameter can be specified only for the ICMP protocol. Minimum value = 0 Maximum value = 65536

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
icmpcode	<Double>	Read-write	Code of a particular ICMP message type to match against the ICMP code of an incoming IPv6 ICMP packet. For example, to block DESTINATION HOST UNREACHABLE messages, specify 3 as the ICMP type and 1 as the ICMP code. If you set this parameter, you must set the ICMP Type parameter. Minimum value = 0 Maximum value = 65536
priority	<Double>	Read-write	Priority for the ACL6 rule, which determines the order in which it is evaluated relative to the other ACL6 rules. If you do not specify priorities while creating ACL6 rules, the ACL6 rules are evaluated in the order in which they are created. Minimum value = 1

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
			Maximum value = 81920
state	<String>	Read-write	State of the ACL6. Default value: ENABLED Possible values = ENABLED, DISABLED
type	<String>	Read-write	Type of the acl6 ,default will be CLASSIC. Available options as follows: * CLASSIC - specifies the regular extended acls. * DFD - cluster specific acls,specifies hashmethod for steering of the packet in cluster . Default value: CLASSIC Possible values = CLASSIC, DFD
dfdhash	<String>	Read-write	Specifies the type of hashmethod to be applied, to steer the packet to the FP of the packet.

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
			Possible values = SIP-SPORT-DIP-DPORT, SIP, DIP, SIP-DIP, SIP-SPORT, DIP-DPORT
dfdprefix	<Double>	Read-write	hashprefix to be applied to SIP/DIP to generate rsshash FP.eg 128 => hash calculated on the complete IP. Default value: 128 Minimum value = 1 Maximum value = 128
stateful	<String>	Read-write	If stateful option is enabled, transparent sessions are created for the traffic hitting this ACL6 and not hitting any other features like LB, INAT etc. . Default value: NO Possible values = YES, NO
logstate	<String>	Read-write	Enable or disable logging of events related to the ACL6 rule. The log messages are stored in the configured

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
			syslog or auditlog server. Default value: DISABLED Possible values = ENABLED, DISABLED
ratelimit	<Double>	Read-write	Maximum number of log messages to be generated per second. If you set this parameter, you must enable the Log State parameter. Default value: 100 Minimum value = 1 Maximum value = 10000
aclaction	<String>	Read-write	Action associated with the ACL6. Possible values = BRIDGE, DENY, ALLOW
newname	<String>	Read-write	New name for the ACL6 rule. Must begin with an ASCII alphabetic or underscore \(_\) character, and must contain only ASCII alphanumeric, underscore, hash \(#\), period \(.\), space, colon \(:\), at \(@\), equals \(=\),

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
			and hyphen \(-\) characters. Minimum length = 1
kernelstate	<String>	Read-only	Commit status of the ACL6. Default value: NOTAPPLIED Possible values = APPLIED, NOTAPPLIED, RE-APPLY, SFAPPLIED, SFNOTAPPLIED
hits	<Double>	Read-only	Number of hits of this ACL6.
aclassociate	<String[]>	Read-only	ACL6 linked. Possible values = NAT, FORWARDINGSESSION, NAT64, LSN
__count	<Double>	Read-only	count parameter

Operations

(click to see Properties)

ADD| DELETE| UPDATE| UNSET| ENABLE| DISABLE| RENAME| GET (ALL)| GET| COUNT

Some options that you can use for each operations:

- **Getting warnings in response:**NITRO allows you to get warnings in an operation by specifying the "warning" query parameter as "yes". For example, to get warnings while connecting to the Citrix ADC appliance, the URL is as follows:

  http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/login?warning=yes

If any, the warnings are displayed in the response payload with the HTTP code "209 X-NITRO-WARNING".

- **Authenticated access for individual NITRO operations:**NITRO allows you to logon to the Citrix ADC appliance to perform individual operations. You can use this option instead of creating a NITRO session (using the login object) and then using that session to perform all operations,

  To do this, you must specify the username and password in the request header of the NITRO request as follows:

  X-NITRO-USER:<username>

  X-NITRO-PASS:<password>

  Note: In such cases, make sure that the request header DOES not include the following:

  Cookie:NITRO_AUTH_TOKEN=<tokenvalue>

`*Note:`*

Mandatory parameters are marked in redand placeholder content is marked in <green>.

add

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/nsacl6

HTTP Method:POST

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Content-Type:application/json

Request Payload:

```
{"nsacl6":{

<b>"acl6name":<String_value>,

</b><b>"acl6action":<String_value>,

</b>"td":<Double_value>,
```

```
"srcipv6":<Boolean_value>,

"srcipop":<String_value>,

"srcipv6val":<String_value>,

"srcport":<Boolean_value>,

"srcportop":<String_value>,

"srcportval":<String_value>,

"destipv6":<Boolean_value>,

"destipop":<String_value>,

"destipv6val":<String_value>,

"destport":<Boolean_value>,

"destportop":<String_value>,

"destportval":<String_value>,

"ttl":<Double_value>,

"srcmac":<String_value>,

"srcmacmask":<String_value>,

"protocol":<String_value>,

"protocolnumber":<Double_value>,

"vlan":<Double_value>,

"vxlan":<Double_value>,

"Interface":<String_value>,

"established":<Boolean_value>,

"icmptype":<Double_value>,
```

```
"icmpcode":<Double_value>,

"priority":<Double_value>,

"state":<String_value>,

"type":<String_value>,

"dfdhash":<String_value>,

"dfdprefix":<Double_value>,

"stateful":<String_value>,

"logstate":<String_value>,

"ratelimit":<Double_value>

}}
```

Response:

HTTP Status Code on Success: 201 Created HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error

delete

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/nsacl6/acl6name_value<String>

HTTP Method:DELETE

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue>

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error

update

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/nsacl6

HTTP Method:PUT

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Content-Type:application/json

Request Payload:

```
{"nsacl6":{

<b>"acl6name":<String_value>,

</b>"aclaction":<String_value>,

"srcipv6":<Boolean_value>,

"srcipop":<String_value>,

"srcipv6val":<String_value>,

"srcport":<Boolean_value>,

"srcportop":<String_value>,

"srcportval":<String_value>,

"destipv6":<Boolean_value>,

"destipop":<String_value>,

"destipv6val":<String_value>,

"destport":<Boolean_value>,

"destportop":<String_value>,

"destportval":<String_value>,
```

```
"srcmac":<String_value>,

"srcmacmask":<String_value>,

"protocol":<String_value>,

"protocolnumber":<Double_value>,

"icmptype":<Double_value>,

"icmpcode":<Double_value>,

"vlan":<Double_value>,

"vxlan":<Double_value>,

"Interface":<String_value>,

"priority":<Double_value>,

"logstate":<String_value>,

"ratelimit":<Double_value>,

"established":<Boolean_value>,

"dfdhash":<String_value>,

"dfdprefix":<Double_value>,

"stateful":<String_value>

}}
```

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error

unset

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/nsacl6?**action=unset**

HTTP Method:POST

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Content-Type:application/json

Request Payload:

```
{"nsacl6":{

<b>"acl6name":<String_value>,

</b>"srcipv6":true,

"srcport":true,

"destipv6":true,

"destport":true,

"srcmac":true,

"srcmacmask":true,

"protocol":true,

"icmptype":true,

"icmpcode":true,

"vlan":true,

"vxlan":true,

"Interface":true,

"logstate":true,
```

```
"ratelimit":true,

"established":true,

"stateful":true,

"dfdhash":true,

"dfdprefix":true

}}
```

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error

enable

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/nsacl6?**action=enable**

HTTP Method:POST

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Content-Type:application/json

Request Payload:

```
{"nsacl6":{

<b>"acl6name":<String_value>

</b>}}
```

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error

disable

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/nsacl6?**action=disable**

HTTP Method:POST

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Content-Type:application/json

Request Payload:

```
{"nsacl6":{

<b>"acl6name":<String_value>

</b>}}
```

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error

rename

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/nsacl6?**action=rename**

HTTP Method:POST

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Content-Type:application/json

Request Payload:

```
{"nsacl6":{

<b>"acl6name":<String_value>,

</b><b>"newname":<String_value>

</b>}}
```

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error

get (all)

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/nsacl6

Query-parameters:

args

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/nsacl6?**args=acl6name:<String_value>,type:<String_value>**

Use this query-parameter to get nsacl6 resources based on additional properties.

attrs

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/nsacl6?**attrs=property-name1,property-name2**

Use this query parameter to specify the resource details that you want to retrieve.

filter

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/nsacl6?**filter=property-name1:property-val1,property-name2:property-val2**

Use this query-parameter to get the filtered set of nsacl6 resources configured on Citrix ADC. Filtering can be done on any of the properties of the resource.

view

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/nsacl6?**view=summary**

Note: By default, the retrieved results are displayed in detail view (?view=detail).

pagination

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/nsacl6?**pagesize=#no;pageno=#no**

Use this query-parameter to get the nsacl6 resources in chunks.

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "nsacl6": [ {

acl6name:<String_value>,type:<String_value>"acl6action":<String_value>,

"srcmac":<String_value>,

"srcmacmask":<String_value>,

"protocol":<String_value>,

"protocolnumber":<Double_value>,

"srcportval":<String_value>,

"srcportop":<String_value>,
```

```
"destportval":<String_value>,

"destportop":<String_value>,

"srcipv6val":<String_value>,

"srcipop":<String_value>,

"destipv6val":<String_value>,

"destipop":<String_value>,

"vlan":<Double_value>,

"vxlan":<Double_value>,

"state":<String_value>,

"kernelstate":<String_value>,

"ttl":<Double_value>,

"icmptype":<Double_value>,

"icmpcode":<Double_value>,

"Interface":<String_value>,

"hits":<Double_value>,

"established":<Boolean_value>,

"priority":<Double_value>,

"logstate":<String_value>,

"ratelimit":<Double_value>,

"td":<Double_value>,

"aclassociate":<String[]_value>,

"dfdhash":<String_value>,
```

```
"dfdprefix":<Double_value>,

"stateful":<String_value>

}]}
```

get

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/nsacl6/acl6name_value<String>

Query-parameters:

attrs

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/nsacl6/acl6name_value<String>?**attrs=property-name1,property-name2**

Use this query parameter to specify the resource details that you want to retrieve.

view

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/nsacl6/acl6name_value<String>?**view=summary**

Note: By default, the retrieved results are displayed in detail view (?view=detail).

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "nsacl6": [ {

acl6name:<String_value>,type:<String_value>"acl6action":<String_value>,

"srcmac":<String_value>,

"srcmacmask":<String_value>,

"protocol":<String_value>,

"protocolnumber":<Double_value>,

"srcportval":<String_value>,

"srcportop":<String_value>,

"destportval":<String_value>,

"destportop":<String_value>,

"srcipv6val":<String_value>,

"srcipop":<String_value>,

"destipv6val":<String_value>,

"destipop":<String_value>,

"vlan":<Double_value>,

"vxlan":<Double_value>,

"state":<String_value>,

"kernelstate":<String_value>,

"ttl":<Double_value>,

"icmptype":<Double_value>,

"icmpcode":<Double_value>,
```

```
"Interface":<String_value>,

"hits":<Double_value>,

"established":<Boolean_value>,

"priority":<Double_value>,

"logstate":<String_value>,

"ratelimit":<Double_value>,

"td":<Double_value>,

"aclassociate":<String[]_value>,

"dfdhash":<String_value>,

"dfdprefix":<Double_value>,

"stateful":<String_value>

}]}
```

count

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/nsacl6?**count=yes**

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "nsacl6": [ { "__count": "#no"} ] }
```

nsacls

Configuration for ACL entry resource.

Properties

(click to see Operations)

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
type	<String>	Read-write	Type of the acl ,default will be CLASSIC. Available options as follows: * CLASSIC - specifies the regular extended acls. * DFD - cluster specific acls,specifies hashmethod for steering of the packet in cluster . Default value: CLASSIC Possible values = CLASSIC, DFD

Operations

(click to see Properties)

RENUMBER| CLEAR| APPLY

Some options that you can use for each operations:

- **Getting warnings in response:**NITRO allows you to get warnings in an operation by specifying the "warning" query parameter as "yes". For example, to get warnings while connecting to the Citrix ADC appliance, the URL is as follows:

  http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/login?warning=yes

  If any, the warnings are displayed in the response payload with the HTTP code "209 X-NITRO-WARNING".

- **Authenticated access for individual NITRO operations:**NITRO allows you to logon to the Citrix ADC appliance to perform individual operations. You can use this option instead of creating a NITRO session (using the login object) and then using that session to perform all operations,

  To do this, you must specify the username and password in the request header of the NITRO request as follows:

  X-NITRO-USER:<username>

  X-NITRO-PASS:<password>

  Note: In such cases, make sure that the request header DOES not include the following:

  Cookie:NITRO_AUTH_TOKEN=<tokenvalue>

`*Note:`*

Mandatory parameters are marked in redand placeholder content is marked in <green>.

renumber

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/nsacls?**action=renumber**

HTTP Method:POST

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Content-Type:application/json

Request Payload:

```
{"nsacls":{

"type":<String_value>

}}
```

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error

clear

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/nsacls?**action=clear**

HTTP Method:POST

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Content-Type:application/json

Request Payload:

```
{"nsacls":{

"type":<String_value>

}}
```

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error

apply

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/nsacls?**action=apply**

HTTP Method:POST

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Content-Type:application/json

Request Payload:

```
{"nsacls":{

"type":<String_value>

}}
```

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error

nsacls6

Configuration for ACL6 entry resource.

Properties

(click to see Operations)

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
type	<String>	Read-write	Type of the acl ,default will be CLASSIC. Available options as follows:

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
			* CLASSIC - specifies the regular extended acls. * DFD - cluster specific acls,specifies hashmethod for steering of the packet in cluster . Default value: CLASSIC Possible values = CLASSIC, DFD

Operations

(click to see Properties)

CLEAR| APPLY| RENUMBER

Some options that you can use for each operations:

- **Getting warnings in response:**NITRO allows you to get warnings in an operation by specifying the "warning" query parameter as "yes". For example, to get warnings while connecting to the Citrix ADC appliance, the URL is as follows:

  http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/login?warning=yes

  If any, the warnings are displayed in the response payload with the HTTP code "209 X-NITRO-WARNING".

- **Authenticated access for individual NITRO operations:**NITRO allows you to logon to the Citrix ADC appliance to perform individual operations. You can use this option instead of creating a NITRO session (using the login object) and then using that session to perform all operations,

  To do this, you must specify the username and password in the request header of the NITRO request as follows:

X-NITRO-USER:<username>

X-NITRO-PASS:<password>

Note: In such cases, make sure that the request header DOES not include the following:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue>

`*Note:` *

Mandatory parameters are marked in redand placeholder content is marked in <green>.

clear

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/nsacls6?**action=clear**

HTTP Method:POST

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Content-Type:application/json

Request Payload:

```
{"nsacls6":{

"type":<String_value>

}}
```

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error

apply

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/nsacls6?**action=apply**

HTTP Method:POST

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Content-Type:application/json

Request Payload:

```
{"nsacls6":{

"type":<String_value>

}}
```

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error

renumber

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/nsacls6?**action=renumber**

HTTP Method:POST

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Content-Type:application/json

Request Payload:

```
{"nsacls6":{

"type":<String_value>

}}
```

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error

nsappflowcollector

Configuration for appflowCollector resource.

Properties

(click to see Operations)

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
name	<String>	Read-write	Name of the AppFlow collector. Minimum length = 1 Maximum length = 127
ipaddress	<String>	Read-write	The IPv4 address of the AppFlow collector.
port	<Integer>	Read-write	The UDP port on which the AppFlow collector is listening. Default value: 4739
__count	<Double>	Read-only	count parameter

Operations

(click to see Properties)

ADD| DELETE| GET (ALL)| GET| COUNT

Some options that you can use for each operations:

- **Getting warnings in response:**NITRO allows you to get warnings in an operation by specifying the "warning" query parameter as "yes". For example, to get warnings while connecting to the Citrix ADC appliance, the URL is as follows:

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/login?warning=yes

If any, the warnings are displayed in the response payload with the HTTP code "209 X-NITRO-WARNING".

- **Authenticated access for individual NITRO operations:**NITRO allows you to logon to the Citrix ADC appliance to perform individual operations. You can use this option instead of creating a NITRO session (using the login object) and then using that session to perform all operations,

  To do this, you must specify the username and password in the request header of the NITRO request as follows:

  X-NITRO-USER:<username>

  X-NITRO-PASS:<password>

  Note: In such cases, make sure that the request header DOES not include the following:

  Cookie:NITRO_AUTH_TOKEN=<tokenvalue>

`*Note:`*

Mandatory parameters are marked in redand placeholder content is marked in <green>.

add

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/nsappflowcollector

HTTP Method:POST

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Content-Type:application/json

Request Payload:

```
{"nsappflowcollector":{

<b>"name":<String_value>,

</b><b>"ipaddress":<String_value>,

</b>"port":<Integer_value>
```

```
}}
```

Response:

HTTP Status Code on Success: 201 Created HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error

delete

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/nsappflowcollector/name_value<String>

HTTP Method:DELETE

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue>

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error

get (all)

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/nsappflowcollector

Query-parameters:

attrs

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/nsappflowcollector?**attrs=property-name1,property-name2**

Use this query parameter to specify the resource details that you want to retrieve.

filter

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/nsappflowcollector?**filter=property-name1:property-val1,property-name2:property-val2**

Use this query-parameter to get the filtered set of nsappflowcollector resources configured on Citrix ADC. Filtering can be done on any of the properties of the resource.

view

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/nsappflowcollector?**view=summary**

Note: By default, the retrieved results are displayed in detail view (?view=detail).

pagination

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/nsappflowcollector?**pagesize=#no;pageno=#no**

Use this query-parameter to get the nsappflowcollector resources in chunks.

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "nsappflowcollector": [ {

"name":<String_value>,

"ipaddress":<String_value>,

"port":<Integer_value>

}]}
```

get

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/nsappflowcollector/name_value<String>

Query-parameters:

attrs

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/nsappflowcollector/name_value<String>?**attrs=property-name1,property-name2**

Use this query parameter to specify the resource details that you want to retrieve.

view

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/nsappflowcollector/name_value<String>?**view=summary**

Note: By default, the retrieved results are displayed in detail view (?view=detail).

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "nsappflowcollector": [ {

"name":<String_value>,

"ipaddress":<String_value>,

"port":<Integer_value>

}]}
```

count

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/nsappflowcollector?**count=yes**

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "nsappflowcollector": [ { "__count": "#no"} ] }
```

nsappflowparam

Configuration for appflowParam resource.

Properties

(click to see Operations)

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
templaterefresh	<Double>	Read-write	IPFIX template refresh interval (in seconds). Default value: 600 Minimum value = 60

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
			Maximum value = 3600
udppmtu	<Double>	Read-write	MTU to be used for IPFIX UDP packets. Default value: 1472 Minimum value = 128 Maximum value = 1472
httpurl	<String>	Read-write	Enable AppFlow HTTP URL logging. Default value: DISABLED Possible values = ENABLED, DISABLED
httpcookie	<String>	Read-write	Enable AppFlow HTTP cookie logging. Default value: DISABLED Possible values = ENABLED, DISABLED
httpreferer	<String>	Read-write	Enable AppFlow HTTP referer logging. Default value: DISABLED

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
			Possible values = ENABLED, DISABLED
httpmethod	<String>	Read-write	Enable AppFlow HTTP method logging. Default value: DISABLED Possible values = ENABLED, DISABLED
httphost	<String>	Read-write	Enable AppFlow HTTP host logging. Default value: DISABLED Possible values = ENABLED, DISABLED
httpuseragent	<String>	Read-write	Enable AppFlow HTTP user-agent logging. Default value: DISABLED Possible values = ENABLED, DISABLED
clienttrafficonly	<String>	Read-write	Control whether AppFlow records should be generated only for client-side traffic. Default value: NO

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
			Possible values = YES, NO

Operations

(click to see Properties)

UPDATE| UNSET| GET (ALL)

Some options that you can use for each operations:

- **Getting warnings in response:**NITRO allows you to get warnings in an operation by specifying the "warning" query parameter as "yes". For example, to get warnings while connecting to the Citrix ADC appliance, the URL is as follows:

  http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/login?warning=yes

  If any, the warnings are displayed in the response payload with the HTTP code "209 X-NITRO-WARNING".

- **Authenticated access for individual NITRO operations:**NITRO allows you to logon to the Citrix ADC appliance to perform individual operations. You can use this option instead of creating a NITRO session (using the login object) and then using that session to perform all operations,

  To do this, you must specify the username and password in the request header of the NITRO request as follows:

  X-NITRO-USER:<username>

  X-NITRO-PASS:<password>

  Note: In such cases, make sure that the request header DOES not include the following:

  Cookie:NITRO_AUTH_TOKEN=<tokenvalue>

`*Note:`*

Mandatory parameters are marked in redand placeholder content is marked in <green>.

update

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/nsappflowparam

HTTP Method:PUT

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Content-Type:application/json

Request Payload:

```
{"nsappflowparam":{

"templaterefresh":<Double_value>,

"udppmtu":<Double_value>,

"httpurl":<String_value>,

"httpcookie":<String_value>,

"httpreferer":<String_value>,

"httpmethod":<String_value>,

"httphost":<String_value>,

"httpuseragent":<String_value>,

"clienttrafficonly":<String_value>

}}
```

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error

unset

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/nsappflowparam?**action=unset**

HTTP Method:POST

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Content-Type:application/json

Request Payload:

```
{"nsappflowparam":{

"templaterefresh":true,

"udppmtu":true,

"httpurl":true,

"httpcookie":true,

"httpreferer":true,

"httpmethod":true,

"httphost":true,

"httpuseragent":true,

"clienttrafficonly":true

}}
```

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error

get (all)

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/nsappflowparam

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "nsappflowparam": [ {

"templaterefresh":<Double_value>,

"udppmtu":<Double_value>,

"httpurl":<String_value>,

"httpcookie":<String_value>,

"httpreferer":<String_value>,

"httpmethod":<String_value>,

"httphost":<String_value>,

"httpuseragent":<String_value>,

"clienttrafficonly":<String_value>

}]}
```

nsaptlicense

Configuration for aptlicense resource.

Properties

(click to see Operations)

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
serialno	<String>	Read-write	Hardware Serial Number/License Activation Code(LAC).
useproxy	<String>	Read-write	Specifies whether to use the licenseproxyserver to reach the internet. Make sure to configure licenseproxyserver to use this option. Default value: NO Possible values = YES, NO
id	<String>	Read-write	License ID.
sessionid	<String>	Read-write	Session ID.
bindtype	<String>	Read-write	Bind type.
countavailable	<String>	Read-write	The user can allocate one or more licenses. Ensure the value is less than (for partial allocation) or equal to the total number of available licenses.
licensedir	<String>	Read-write	License Directory.
response	<String>	Read-only	Response data as text blob.

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
counttotal	<String>	Read-only	Count.
name	<String>	Read-only	License name.
relevance	<String>	Read-only	License relevance.
datepurchased	<String>	Read-only	License purchase date.
datesa	<String>	Read-only	License SA date.
dateexp	<String>	Read-only	License expiry date.
features	<String[]>	Read-only	Features.
__count	<Double>	Read-only	count parameter

Operations

(click to see Properties)

CHANGE| GET (ALL)| COUNT

Some options that you can use for each operations:

- **Getting warnings in response:**NITRO allows you to get warnings in an operation by specifying the "warning" query parameter as "yes". For example, to get warnings while connecting to the Citrix ADC appliance, the URL is as follows:

  http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/login?warning=yes

  If any, the warnings are displayed in the response payload with the HTTP code "209 X-NITRO-WARNING".

- **Authenticated access for individual NITRO operations:**NITRO allows you to logon to the Citrix ADC appliance to perform individual operations. You can use this option instead of creating a NITRO session (using the login object) and then using that session to perform all operations,

  To do this, you must specify the username and password in the request header of the NITRO request as follows:

X-NITRO-USER:<username>

X-NITRO-PASS:<password>

Note: In such cases, make sure that the request header DOES not include the following:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue>

`*Note:`*

Mandatory parameters are marked in redand placeholder content is marked in <green>.

change

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/nsaptlicense?**action=update**

HTTP Method:POST

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Content-Type:application/json

Request Payload:

```
{"nsaptlicense":{

<b>"id":<String_value>,

</b><b>"sessionid":<String_value>,

</b><b>"bindtype":<String_value>,

</b><b>"countavailable":<String_value>,

</b>"licensedir":<String_value>,

"useproxy":<String_value>

}}
```

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error

get (all)

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/nsaptlicense

Query-parameters:

args

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/nsaptlicense?
args=serialno:<String_value>,useproxy:<String_value>

Use this query-parameter to get nsaptlicense resources based on additional properties.

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "nsaptlicense": [ {

<b>serialno:<String_value>,</
b>useproxy:<String_value>"response":<String_value>,

"id":<String_value>,

"sessionid":<String_value>,

"bindtype":<String_value>,
```

```
"countavailable":<String_value>,

"counttotal":<String_value>,

"name":<String_value>,

"relevance":<String_value>,

"datepurchased":<String_value>,

"datesa":<String_value>,

"dateexp":<String_value>,

"features":<String[]_value>

}]}
```

count

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/nsaptlicense?**args=serialno:<String_value>,useproxy:<String_value>;count=yes**

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "nsaptlicense": [ { "__count": "#no"} ] }
```

nsassignment

Configuration for assignment resource.

Properties

(click to see Operations)

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
name	<String>	Read-write	Name for the assignment. Must begin with a letter, number, or the underscore character (_), and must contain only letters, numbers, and the hyphen (-), period (.) hash (#), space ( ), at (@), equals (=), colon (:), and underscore characters. Can be changed after the assignment is added. The following requirement applies only to the Citrix ADC CLI: If the name includes one or more spaces, enclose the name in double or single quotation marks (for example, "my assignment" or my assignment).

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
variable	<String>	Read-write	Left hand side of the assigment, of the form $variable-name (for a singleton variabled) or $variable-name[key-expression], where key-expression is an expression that evaluates to a text string and provides the key to select a map entry.
set	<String>	Read-write	Right hand side of the assignment. The expression is evaluated and assigned to the left hand variable.
Add	<String>	Read-write	Right hand side of the assignment. The expression is evaluated and added to the left hand variable.
sub	<String>	Read-write	Right hand side of the assignment. The expression is evaluated and subtracted from the left hand variable.
append	<String>	Read-write	Right hand side of the assignment. The expression is evaluated and appended to the left hand variable.

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
clear	<Boolean>	Read-write	Clear the variable value. Deallocates a text value, and for a map, the text key.
comment	<String>	Read-write	Comment. Can be used to preserve information about this rewrite action.
newname	<String>	Read-write	New name for the assignment. Must begin with a letter, number, or the underscore character (_), and must contain only letters, numbers, and the hyphen (-), period (.) hash (#), space ( ), at (@), equals (=), colon (:), and underscore characters. Can be changed after the rewrite policy is added. The following requirement applies only to the Citrix ADC CLI: If the name includes one or more spaces, enclose the name in double or single quotation

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
			marks (for example, "my assignment" or my assignment). Minimum length = 1
hits	<Double>	Read-only	The number of times the action has been taken.
undefhits	<Double>	Read-only	The number of times the action resulted in UNDEF.
referencecount	<Double>	Read-only	The number of references to the action.
__count	<Double>	Read-only	count parameter

Operations

(click to see Properties)

ADD| UPDATE| UNSET| DELETE| RENAME| GET (ALL)| GET| COUNT

Some options that you can use for each operations:

- **Getting warnings in response:**NITRO allows you to get warnings in an operation by specifying the "warning" query parameter as "yes". For example, to get warnings while connecting to the Citrix ADC appliance, the URL is as follows:

  http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/login?warning=yes

  If any, the warnings are displayed in the response payload with the HTTP code "209 X-NITRO-WARNING".

- **Authenticated access for individual NITRO operations:**NITRO allows you to logon to the Citrix ADC appliance to perform individual operations. You can use this option instead of creating a NITRO session (using the login object) and then using that session to perform all operations,

To do this, you must specify the username and password in the request header of the NITRO request as follows:

X-NITRO-USER:<username>

X-NITRO-PASS:<password>

Note: In such cases, make sure that the request header DOES not include the following:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue>

`*Note:`*

Mandatory parameters are marked in redand placeholder content is marked in <green>.

add

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/nsassignment

HTTP Method:POST

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Content-Type:application/json

Request Payload:

```
{"nsassignment":{

<b>"name":<String_value>,

</b><b>"variable":<String_value>,

</b>"set":<String_value>,

"Add":<String_value>,

"sub":<String_value>,

"append":<String_value>,

"clear":<Boolean_value>,
```

```
"comment":<String_value>

}}
```

Response:

HTTP Status Code on Success: 201 Created HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error

update

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/nsassignment

HTTP Method:PUT

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Content-Type:application/json

Request Payload:

```
{"nsassignment":{

<b>"name":<String_value>,

</b>"variable":<String_value>,

"set":<String_value>,

"Add":<String_value>,

"sub":<String_value>,

"append":<String_value>,

"clear":<Boolean_value>,

"comment":<String_value>
```

```
}}
```

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error

unset

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/nsassignment?**action=unset**

HTTP Method:POST

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Content-Type:application/json

Request Payload:

```
{"nsassignment":{

<b>"name":<String_value>,

</b>"comment":true

}}
```

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error

delete

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/nsassignment/name_value<String>

HTTP Method:DELETE

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue>

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error

rename

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/nsassignment?**action=rename**

HTTP Method:POST

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Content-Type:application/json

Request Payload:

```
{"nsassignment":{

<b>"name":<String_value>,

</b><b>"newname":<String_value>

</b>}}
```

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error

get (all)

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/nsassignment

Query-parameters:

attrs

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/nsassignment?**attrs=property-name1,property-name2**

Use this query parameter to specify the resource details that you want to retrieve.

filter

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/nsassignment?**filter=property-name1:property-val1,property-name2:property-val2**

Use this query-parameter to get the filtered set of nsassignment resources configured on Citrix ADC. Filtering can be done on any of the properties of the resource.

view

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/nsassignment?**view=summary**

Note: By default, the retrieved results are displayed in detail view (?view=detail).

pagination

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/nsassignment?**pagesize=#no;pageno=#no**

Use this query-parameter to get the nsassignment resources in chunks.

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "nsassignment": [ {

"name":<String_value>,

"variable":<String_value>,

"set":<String_value>,

"Add":<String_value>,

"sub":<String_value>,

"append":<String_value>,

"clear":<Boolean_value>,

"hits":<Double_value>,

"undefhits":<Double_value>,

"referencecount":<Double_value>,

"comment":<String_value>

}]}
```

get

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/nsassignment/name_value<String>

Query-parameters:

attrs

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/nsassignment/name_value<String>?**attrs=property-name1,property-name2**

Use this query parameter to specify the resource details that you want to retrieve.

view

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/nsassignment/name_value<String>?**view=summary**

Note: By default, the retrieved results are displayed in detail view (?view=detail).

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "nsassignment": [ {

"name":<String_value>,

"variable":<String_value>,

"set":<String_value>,

"Add":<String_value>,

"sub":<String_value>,

"append":<String_value>,

"clear":<Boolean_value>,

"hits":<Double_value>,

"undefhits":<Double_value>,

"referencecount":<Double_value>,
```

```
"comment":<String_value>

}]}
```

count

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/nsassignment?**count=yes**

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "nsassignment": [ { "__count": "#no"} ] }
```

nscapacity

Configuration for capacity resource.

Properties

(click to see Operations)

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
bandwidth	<Double>	Read-write	System bandwidth limit.

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
platform	<String>	Read-write	appliance platform type. Possible values = VS10, VE10, VP10, VS25, VE25, VP25, VS200, VE200, VP200, VS1000, VE1000, VP1000, VS3000, VE3000, VP3000, VS5000, VE5000, VP5000, VS8000, VE8000, VP8000, VS10000, VE10000, VP10000, VS15000, VE15000, VP15000, VS25000, VE25000, VP25000, VS40000, VE40000, VP40000, VS100000, VE100000, VP100000, CP1000
vcpu	<Boolean>	Read-write	licensed using vcpu pool.
edition	<String>	Read-write	Product edition. Possible values = Standard, Enterprise, Platinum

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
unit	<String>	Read-write	Bandwidth unit. Possible values = Gbps, Mbps
nodeid	<Double>	Read-write	Unique number that identifies the cluster node. Minimum value = 0 Maximum value = 31
actualbandwidth	<Double>	Read-only	Bandwith in MBPS.
vcpucount	<Double>	Read-only	number of vCPUs licensed. Minimum value = 1 Maximum value = 20
maxvcpucount	<Double>	Read-only	number of max vCPUs. Minimum value = 1 Maximum value = 20
maxbandwidth	<Double>	Read-only	Maximum Bandwidth.
minbandwidth	<Double>	Read-only	Minimum Bandwidth.
instancecount	<Double>	Read-only	VPX will consume one instance and MPX will consume zero instance.

Operations

(click to see Properties)

UPDATE| UNSET| GET (ALL)

Some options that you can use for each operations:

- **Getting warnings in response:**NITRO allows you to get warnings in an operation by specifying the "warning" query parameter as "yes". For example, to get warnings while connecting to the Citrix ADC appliance, the URL is as follows:

  http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/login?warning=yes

  If any, the warnings are displayed in the response payload with the HTTP code "209 X-NITRO-WARNING".

- **Authenticated access for individual NITRO operations:**NITRO allows you to logon to the Citrix ADC appliance to perform individual operations. You can use this option instead of creating a NITRO session (using the login object) and then using that session to perform all operations,

  To do this, you must specify the username and password in the request header of the NITRO request as follows:

  X-NITRO-USER:<username>

  X-NITRO-PASS:<password>

  Note: In such cases, make sure that the request header DOES not include the following:

  Cookie:NITRO_AUTH_TOKEN=<tokenvalue>

`*Note:`*

Mandatory parameters are marked in redand placeholder content is marked in <green>.

update

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/nscapacity

HTTP Method:PUT

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Content-Type:application/json

Request Payload:

```
{"nscapacity":{

"bandwidth":<Double_value>,

"platform":<String_value>,

"vcpu":<Boolean_value>,

"edition":<String_value>,

"unit":<String_value>

}}
```

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error

unset

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/nscapacity?**action=unset**

HTTP Method:POST

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Content-Type:application/json

Request Payload:

```
{"nscapacity":{

"bandwidth":true,

"platform":true,
```

```
"vcpu":true

}}
```

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error

get (all)

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/nscapacity

Query-parameters:

args

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/nscapacity?**args=nodeid:<Double_value>**

Use this query-parameter to get nscapacity resources based on additional properties.

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "nscapacity": [ {

nodeid:<Double_value>"actualbandwidth":<Double_value>,
```

```
"platform":<String_value>,

"vcpucount":<Double_value>,

"maxvcpucount":<Double_value>,

"edition":<String_value>,

"unit":<String_value>,

"bandwidth":<Double_value>,

"maxbandwidth":<Double_value>,

"minbandwidth":<Double_value>,

"instancecount":<Double_value>

}]}
```

nscentralmanagementserver

Configuration for centralmanagementserver resource.

Properties

(click to see Operations)

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
type	<String>	Read-write	Type of the central management server. Must be either CLOUD or ONPREM depending on whether the server is on the cloud or on premise. Possible values = CLOUD, ONPREM

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
username	<String>	Read-write	Username for access to central management server. Must begin with a letter, number, or the underscore character (_), and must contain only letters, numbers, and the hyphen (-), period (.) pound (#), space ( ), at (@), equals (=), colon (:), and underscore characters. The following requirement applies only to the Citrix ADC CLI: If the name includes one or more spaces, enclose the name in double or single quotation marks (for example, "my ns centralmgmtserver" or "my ns centralmgmtserver"). Minimum length = 1 Maximum length = 127

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
password	<String>	Read-write	Password for access to central management server. Required for any user account. Minimum length = 1 Maximum length = 127
ipaddress	<String>	Read-write	Ip Address of central management server. Minimum length = 1
servername	<String>	Read-write	Fully qualified domain name of the central management server.
__count	<Double>	Read-only	count parameter

Operations

(click to see Properties)

ADD| DELETE| GET (ALL)| GET| COUNT

Some options that you can use for each operations:

- **Getting warnings in response:**NITRO allows you to get warnings in an operation by specifying the "warning" query parameter as "yes". For example, to get warnings while connecting to the Citrix ADC appliance, the URL is as follows:

  http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/login?warning=yes

  If any, the warnings are displayed in the response payload with the HTTP code "209 X-NITRO-WARNING".

- **Authenticated access for individual NITRO operations:**NITRO allows you to logon to the Citrix ADC appliance to perform individual operations. You can use this option instead of

creating a NITRO session (using the login object) and then using that session to perform all operations,

To do this, you must specify the username and password in the request header of the NITRO request as follows:

X-NITRO-USER:<username>

X-NITRO-PASS:<password>

Note: In such cases, make sure that the request header DOES not include the following:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue>

`*Note:`*

Mandatory parameters are marked in redand placeholder content is marked in <green>.

add

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/nscentralmanagementserver

HTTP Method:POST

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Content-Type:application/json

Request Payload:

```
{"nscentralmanagementserver":{

<b>"type":<String_value>,

</b><b>"username":<String_value>,

</b><b>"password":<String_value>,

</b>"ipaddress":<String_value>,

"servername":<String_value>

}}
```

Response:

HTTP Status Code on Success: 201 Created HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error

delete

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/nscentralmanagementserver/type_value<String>

HTTP Method:DELETE

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue>

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error

get (all)

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/nscentralmanagementserver

Query-parameters:

attrs

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/nscentralmanagementserver?**attrs=property-name1,property-name2**

Use this query parameter to specify the resource details that you want to retrieve.

filter

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/nscentralmanagementserver?**filter=property-name1:property-val1,property-name2:property-val2**

Use this query-parameter to get the filtered set of nscentralmanagementserver resources configured on Citrix ADC. Filtering can be done on any of the properties of the resource.

view

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/nscentralmanagementserver?**view=summary**

Note: By default, the retrieved results are displayed in detail view (?view=detail).

pagination

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/nscentralmanagementserver?**pagesize=#no;pageno=#no**

Use this query-parameter to get the nscentralmanagementserver resources in chunks.

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "nscentralmanagementserver": [ {

"type":<String_value>,

"username":<String_value>,

"ipaddress":<String_value>,

"servername":<String_value>,

"password":<String_value>

}]}
```

get

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/nscentralmanagementserver/type_value<String>

Query-parameters:

attrs

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/nscentralmanagementserver/type_value<String>?**attrs=property-name1,property-name2**

Use this query parameter to specify the resource details that you want to retrieve.

view

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/nscentralmanagementserver/type_value<String>?**view=summary**

Note: By default, the retrieved results are displayed in detail view (?view=detail).

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "nscentralmanagementserver": [ {

"type":<String_value>,
```

```
"username":<String_value>,

"ipaddress":<String_value>,

"servername":<String_value>,

"password":<String_value>

}]}
```

count

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/nscentralmanagementserver?**count=yes**

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "nscentralmanagementserver": [ { "__count": "#no"} ] }
```

nsconfig

Configuration for system config resource.

Properties

(click to see Operations)

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
force	<Boolean>	Read-write	Configurations will be cleared without prompting for confirmation.
level	<String>	Read-write	Types of configurations to be cleared. * basic: Clears all configurations except the following: - NSIP, default route (gateway), MIPs, and SNIPs - Network settings (DG, VLAN, RHI and DNS settings) - Cluster settings - HA node definitions - Feature and mode settings - nsroot password * extended: Clears the same configurations as the 'basic' option. In addition, it clears

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
			the feature and mode settings. * full: Clears all configurations except NSIP, default route, and interface settings. Note: When you clear the configurations through the cluster IP address, by specifying the level as 'full', the cluster is deleted and all cluster nodes become standalone appliances. The 'basic' and 'extended' levels are propagated to the cluster nodes. Possible values = basic, extended, full
rbaconfig	<String>	Read-write	RBA configurations and TACACS policies bound to system global will not be cleared if RBA is set to NO.This option is applicable only for BASIC level of clear configuration.Default is YES, which will

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
			clear rba configurations. Default value: YES Possible values = YES, NO
ipaddress	<String>	Read-write	IP address of the Citrix ADC. Commonly referred to as NSIP address. This parameter is mandatory to bring up the appliance. Minimum length = 1
netmask	<String>	Read-write	Netmask corresponding to the IP address. This parameter is mandatory to bring up the appliance.
nsvlan	<Double>	Read-write	VLAN (NSVLAN) for the subnet on which the IP address resides. Minimum value = 2 Maximum value = 4094
ifnum	<String[]>	Read-write	Interfaces of the appliances that must be bound to the NSVLAN. Minimum length = 1
tagged	<String>	Read-write	Specifies that the interfaces will be added as 802.1q

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
			tagged interfaces. Packets sent on these interface on this VLAN will have an additional 4-byte 802.1q tag which identifies the VLAN. To use 802.1q tagging, the switch connected to the appliance's interfaces must also be configured for tagging. Default value: YES Possible values = YES, NO
httpport	<Integer[]>	Read-write	The HTTP ports on the Web server. This allows the system to perform connection off-load for any client request that has a destination port matching one of these configured ports. Minimum value = 1
maxconn	<Double>	Read-write	The maximum number of connections that will be made from the system to the web server(s) attached to it. The value entered here is applied

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
			globally to all attached servers. Minimum value = 0 Maximum value = 4294967294
maxreq	<Double>	Read-write	The maximum number of requests that the system can pass on a particular connection between the system and a server attached to it. Setting this value to 0 allows an unlimited number of requests to be passed. Minimum value = 0 Maximum value = 65535
cip	<String>	Read-write	The option to control (enable or disable) the insertion of the actual client IP address into the HTTP header request passed from the client to one, some, or all servers attached to the system. The passed address can then be accessed through a minor

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
			modification to the server. l If cipHeader is specified, it will be used as the client IP header. l If it is not specified, then the value that has been set by the set ns config CLI command will be used as the client IP header. Possible values = ENABLED, DISABLED
cipheader	<String>	Read-write	The text that will be used as the client IP header. Minimum length = 1
cookieversion	<String>	Read-write	The version of the cookie inserted by system. Possible values = 0, 1
securecookie	<String>	Read-write	enable/disable secure flag for persistence cookie. Default value: ENABLED

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
			Possible values = ENABLED, DISABLED
pmtumin	<Double>	Read-write	The minimum Path MTU. Default value: 576 Minimum value = 168 Maximum value = 1500
pmtutimeout	<Double>	Read-write	The timeout value in minutes. Default value: 10 Minimum value = 1 Maximum value = 1440
ftpportrange	<String>	Read-write	Port range configured for FTP services. Minimum length = 1024 Maximum length = 64000
crportrange	<String>	Read-write	Port range for cache redirection services. Minimum length = 1 Maximum length = 65535

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
timezone	<String>	Read-write	Name of the timezone. Default value: CoordinatedUniversalTime Minimum length = 1 Maximum length = 64
grantquotamaxclient	<Double>	Read-write	The percentage of shared quota to be granted at a time for maxClient. Default value: 10 Minimum value = 0 Maximum value = 100
exclusivequotamaxclient	<Double>	Read-write	The percentage of maxClient to be given to PEs. Default value: 80 Minimum value = 0 Maximum value = 100
grantquotaspillover	<Double>	Read-write	The percentage of shared quota to be granted at a time for spillover. Default value: 10 Minimum value = 0

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
			Maximum value = 100
exclusivequotaspillover	<Double>	Read-write	The percentage of max limit to be given to PEs. Default value: 80 Minimum value = 0 Maximum value = 100
all	<Boolean>	Read-write	Use this option to do saveconfig for all partitions.
config1	<String>	Read-write	Location of the configurations.
config2	<String>	Read-write	Location of the configurations.
outtype	<String>	Read-write	Format to display the difference in configurations. Possible values = cli, xml
template	<Boolean>	Read-write	File that contains the commands to be compared.
ignoredevicespecific	<Boolean>	Read-write	Suppress device specific differences.
weakpassword	<Boolean>	Read-write	Option to list all weak passwords (not adhering to strong password requirements). Takes

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
			config file as input, if no input specified, running configuration is considered. Command => query ns config -weakpassword / query ns config -weakpassword / nsconfig/ns.conf.
config	<String>	Read-write	configuration File to be used to find weak passwords, if not specified, running config is taken as input.
message	<String>	Read-only	.
mappedip	<String>	Read-only	Mapped IP Address of the System. Minimum length = 1
range	<Double>	Read-only	The range of Mapped IP addresses to be configured. Minimum value = 1
systemtype	<String>	Read-only	The type of the System. Possible Values: Standalone, HA, Cluster. Possible values = Stand-alone, HA, Cluster
primaryip	<String>	Read-only	HA Master Node IP address.

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
primaryip6	<String>	Read-only	.
flags	<Double>	Read-only	The flags for this entry.
lastconfigchangedtime	<String>	Read-only	Time when the configuration was last modified.
lastconfigsavetime	<String>	Read-only	Time when the configuration was last saved through savensconfig.
currentsytemtime	<String>	Read-only	current system time in date format.
systemtime	<Double>	Read-only	current system time.
configchanged	<Boolean>	Read-only	returns True if configuration has changed since last saved config.
response	<String>	Read-only	.

Operations

(click to see Properties)

CLEAR| UPDATE| UNSET| SAVE| DIFF| GET (ALL)

Some options that you can use for each operations:

- **Getting warnings in response:**NITRO allows you to get warnings in an operation by specifying the "warning" query parameter as "yes". For example, to get warnings while connecting to the Citrix ADC appliance, the URL is as follows:

  http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/login?warning=yes

  If any, the warnings are displayed in the response payload with the HTTP code "209 X-NITRO-WARNING".

- **Authenticated access for individual NITRO operations:**NITRO allows you to logon to the Citrix ADC appliance to perform individual operations. You can use this option instead of creating a NITRO session (using the login object) and then using that session to perform all operations,

  To do this, you must specify the username and password in the request header of the NITRO request as follows:

  X-NITRO-USER:<username>

  X-NITRO-PASS:<password>

  Note: In such cases, make sure that the request header DOES not include the following:

  Cookie:NITRO_AUTH_TOKEN=<tokenvalue>

`*Note:`*

Mandatory parameters are marked in redand placeholder content is marked in <green>.

clear

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/nsconfig?**action=clear**

HTTP Method:POST

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Content-Type:application/json

Request Payload:

```
{"nsconfig":{

"force":<Boolean_value>,

<b>"level":<String_value>,

</b>"rbaconfig":<String_value>

}}
```

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error

update

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/nsconfig

HTTP Method:PUT

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Content-Type:application/json

Request Payload:

```
{"nsconfig":{

"ipaddress":<String_value>,

"netmask":<String_value>,

"nsvlan":<Double_value>,

"ifnum":<String[]_value>,

"tagged":<String_value>,

"httpport":<Integer[]_value>,

"maxconn":<Double_value>,

"maxreq":<Double_value>,

"cip":<String_value>,

"cipheader":<String_value>,

"cookieversion":<String_value>,
```

```
"securecookie":<String_value>,

"pmtumin":<Double_value>,

"pmtutimeout":<Double_value>,

"ftpportrange":<String_value>,

"crportrange":<String_value>,

"timezone":<String_value>,

"grantquotamaxclient":<Double_value>,

"exclusivequotamaxclient":<Double_value>,

"grantquotaspillover":<Double_value>,

"exclusivequotaspillover":<Double_value>

}}
```

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error

unset

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/nsconfig?**action=unset**

HTTP Method:POST

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Content-Type:application/json

Request Payload:

```
{"nsconfig":{
"nsvlan":true,
"ftpportrange":true,
"crportrange":true,
"timezone":true,
"ipaddress":true,
"netmask":true,
"ifnum":true,
"tagged":true,
"httpport":true,
"maxconn":true,
"maxreq":true,
"cip":true,
"cipheader":true,
"cookieversion":true,
"securecookie":true,
"pmtumin":true,
"pmtutimeout":true,
"grantquotamaxclient":true,
"exclusivequotamaxclient":true,
"grantquotaspillover":true,
```

```
"exclusivequotaspillover":true

}}
```

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error

save

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/nsconfig?**action=save**

HTTP Method:POST

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Content-Type:application/json

Request Payload:

```
{"nsconfig":{

"all":<Boolean_value>

}}
```

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error

diff

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/nsconfig?**action=diff**

HTTP Method:POST

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Content-Type:application/json

Request Payload:

```
{"nsconfig":{

"config1":<String_value>,

"config2":<String_value>,

"outtype":<String_value>,

"template":<Boolean_value>,

"ignoredevicespecific":<Boolean_value>

}}
```

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Payload:**

```
{ "nsconfig": [ {

"response":<String_value>



}]}
```

get (all)

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/nsconfig

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "nsconfig": [ {

"ipaddress":<String_value>,

"netmask":<String_value>,

"mappedip":<String_value>,

"range":<Double_value>,

"nsvlan":<Double_value>,

"ifnum":<String[]_value>,

"tagged":<String_value>,

"httpport":<Integer[]_value>,

"maxconn":<Double_value>,

"maxreq":<Double_value>,

"cip":<String_value>,

"cipheader":<String_value>,
```

```
"cookieversion":<String_value>,

"securecookie":<String_value>,

"failover":<String_value>,

"systemtype":<String_value>,

"primaryip":<String_value>,

"primaryip6":<String_value>,

"pmtumin":<Double_value>,

"pmtutimeout":<Double_value>,

"ftpportrange":<String_value>,

"crportrange":<String_value>,

"flags":<Double_value>,

"timezone":<String_value>,

"lastconfigchangedtime":<String_value>,

"lastconfigsavetime":<String_value>,

"currentsytemtime":<String_value>,

"systemtime":<Double_value>,

"grantquotamaxclient":<Double_value>,

"exclusivequotamaxclient":<Double_value>,

"grantquotaspillover":<Double_value>,

"exclusivequotaspillover":<Double_value>,

"configchanged":<Boolean_value>

}]}
```

nsconnectiontable

Configuration for TCP/IP connection table resource.

Properties

(click to see Operations)

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
filterexpression	<String>	Read-write	The maximum length of filter expression is 255 and it can be of following format: <expression> [<relop> <expression>] <relop> = (;; | ||) <expression> =: CONNECTION.<qualifier>.<qualifier-method>.(<qualifier-value>) <qualifier> = SRCIP <qualifier-method> = [EQ | NE] <qualifier-value> = A valid IPv4 address example = CONNECTION.SRCIP.EQ(127.0.0.1)

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
			<qualifier> = DSTIP <qualifier-method> = [EQ | NE] <qualifier-value> = A valid IPv4 address. example = CONNECTION.DSTIP.EQ(127.0.0.1) <qualifier> = IP <qualifier-method> = [EQ | NE] <qualifier-value> = A valid IPv4 address. example = CONNECTION.IP.EQ(127.0.0.1) <qualifier> = SRCIPv6 <qualifier-method> = [EQ | NE] <qualifier-value> = A valid IPv6 address. example = CONNECTION.SR

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
			CIPv6.EQ(2001:db8:0:0:1::1) <qualifier> = DSTIPv6 <qualifier-method> = [EQ | NE] <qualifier-value> = A valid IPv6 address. example = CONNECTION.DSTIPv6.EQ(2001:db8:0:0:1::1) <qualifier> = IPv6 <qualifier-method> = [EQ | NE] <qualifier-value> = A valid IPv6 address. example = CONNECTION.IPv6.EQ(2001:db8:0:0:1::1) <qualifier> = SRCPORT <qualifier-method> = [EQ | NE | GT | GE | LT | LE

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
			| BETWEEN] <qualifier-value> = A valid port number. example = CONNECTION.SRCPORT.EQ(80) <qualifier> = DSTPORT <qualifier-method> = [EQ | NE | GT | GE | LT | LE | BETWEEN] <qualifier-value> = A valid port number. example = CONNECTION.DSTPORT.EQ(80) <qualifier> = PORT <qualifier-method> = [EQ | NE | GT | GE | LT | LE | BETWEEN] <qualifier-value> = A valid port number. example = CONNECTION.PORT.EQ(80)

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
			<qualifier> = SVCNAME <qualifier-method> = [EQ | NE | CONTAINS | STARTSWITH | ENDSWITH] <qualifier-value> = service name. example = CONNECTION.SVCNAME.EQ("name") <qualifier> = LB_VSERVER.NAME <qualifier-method> = [EQ | NE | CONTAINS | STARTSWITH | ENDSWITH] <qualifier-value> = LB vserver name. example = CONNECTION.LB_VSERVER.NAME.EQ("name")

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
			<qualifier> = CS_VSERVER.NAME <qualifier-method> = [EQ | NE | CONTAINS | STARTSWITH | ENDSWITH] <qualifier-value> = CS vserver name. example = CONNECTION.CS_VSERVER.NAME.EQ("name") <qualifier> = INTF <qualifier-method> = [EQ | NE] <qualifier-value> = A valid interface id in the form of x/y (n/x/y in case of cluster interface). examle = CONNECTION.INTF.EQ("0/1/1") <qualifier> = VLANID

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
			<qualifier-method> = [EQ | NE | GT | GE | LT | LE | BETWEEN] <qualifier-value> = A valid VLAN ID. example = CONNECTION.VLANID.EQ(0) <qualifier> = CONNID <qualifier-method> = [EQ | NE | GT | GE | LT | LE | BETWEEN] <qualifier-value> = A valid PCB dev number. example = CONNECTION.CONNID.EQ(0) <qualifier> = PPEID <qualifier-method> = [EQ | NE | GT | GE | LT | LE | BETWEEN] <qualifier-value> = A valid core ID.

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
			example = CONNECTION.PPEID.EQ(0) <qualifier> = IDLETIME <qualifier-method> = [EQ | NE | GT | GE | LT | LE | BETWEEN] <qualifier-value> = A positive integer indicating the idletime. example = CONNECTION.IDLETIME.LT(100) <qualifier> = TCPSTATE <qualifier-method> = [EQ | NE] <qualifier-value> = (CLOSE_WAIT | CLOSED | CLOSING | ESTABLISHED | FIN_WAIT_1 | FIN_WAIT_2 | LAST_ACK |

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
			LISTEN | SYN_RECEIVED | SYN_SENT | TIME_WAIT | NOT_APPLICABLE) example = CONNECTION.TCPSTATE.EQ(LISTEN) <qualifier> = SERVICE_TYPE <qualifier-method> = [EQ | NE] <qualifier-value> = (SVC_HTTP | FTP | TCP | UDP | SSL | SSL_BRIDGE | SSL_TCP | NNTP | RPCSVR | RPCSVRS | RPCCLNT | SVC_DNS | ADNS | SNMP | RTSP | DHCPRA | NAT | ANY | MONITOR | MONITOR_UDP | MONITOR_PING | SIP_UDP |

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
			SVC_MYSQL | SVC_MSSQL | SERVICE_UNKNOWN) example = CONNECTION.SERVICE_TYPE.EQ(ANY) <qualifier> = TRAFFIC_DOMAIN_ID <qualifier-method> = [EQ | NE | GT | GE | LT | LE | BETWEEN] <qualifier-value> = A valid traffic domain ID. example = CONNECTION.TRAFFIC_DOMAIN_ID.EQ(0) common usecases: Filtering out loopback connections and view present connections through netsclaer

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
			show connectiontable "CONNECTION.IP.NEQ(127.0.0.1) ;; CONNECTION.TCPSTATE.EQ(ESTABLISHED)" -detail full show connections from a particular sourceip and targeted to port 80 show connectiontable "CONNECTION.SRCIP.EQ(10.102.1.91) ;; CONNECTION.DSTPORT.EQ(80)" show connection particular to a service and its linked client connections show connectiontable "CONNECTION.SVCNAME.EQ("S1")" -detail link

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
			show connections for a particular servicetype(e.g.http) show connectiontable "CONNECTION.SERVICE_TYPE.EQ(TCP)" viewing connections that have been idle for a long time show connectiontable "CONNECTION.IDLETIME.GT(100)" show connections for a particular interface and vlan show connectiontable "CONNECTION.INTF.EQ("1/1") ;; CONNECTION.VLANID.EQ(1)" .
link	<Boolean>	Read-write	Display link information if available.

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
filtername	<Boolean>	Read-write	Display name instead of IP for local entities.
detail	<String[]>	Read-write	Specify display options for the connection table. * LINK - Displays the linked PCB (Protocol Control Block). * NAME - Displays along with the service name. * CONNFAILOVER - Displays PCB with connection failover. * FULL - Displays all available details. Possible values = LINK, NAME, CONNFAILOVER, FULL, NNM, FULL+
listen	<Boolean>	Read-write	Display listening services only.
nodeid	<Double>	Read-write	Unique number that identifies the cluster node. Minimum value = 0 Maximum value = 31

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
sourceip	<String>	Read-only	Source IP of the connection.
sourceport	<Integer>	Read-only	Source port of the connection. Range 1 - 65535 * in CLI is represented as 65535 in NITRO API
destip	<String>	Read-only	Destination IP of the connection.
destport	<Integer>	Read-only	Destination port of the connection. Range 1 - 65535 * in CLI is represented as 65535 in NITRO API
svctype	<String>	Read-only	Protocol supported by the connection. Possible values = HTTP, FTP, TCP, UDP, SSL, SSL_BRIDGE, SSL_TCP, NNTP, RPCSVR, DNS, ADNS, SNMP, RTSP, DHCPRA, ANY, SIP_UDP, SIP_TCP, SIP_SSL, DNS_TCP, ADNS_TCP,

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
			HTTPSVR, HTTPCLIENT, NAT, HA, AAA, SINCTCP, VPN AFTP, MONITORS, SSLVPN UDP, SINCUDP, RIP, UDP CLNT, SASP, RPCCLNTS, ROUTE, AUDIT, STA HTTP, STA SSL, DNS RESOLVE, RDP, MYSQL, MSSQL, DIAMETER, SSL_DIAMETER, TFTP, ORACLE, ICA, RADIUS, RADIUSListener, SMPP, PPTP, GRE, SYSLOGTCP, SYSLOGUDP, FIX, SSL_FIX, QUIC, TFTP_DATA, USER_TCP, USER_SSL_TCP
idletime	<Integer>	Read-only	Time since last activity was detected on the connection.
state	<String>	Read-only	Current TCP/IP state of the connection. Possible values = CLOSED, LISTEN, SYN_SENT, SYN_RECEIVED,

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
			ESTABLISHED, CLOSE_WAIT, FIN_WAIT_1, CLOSING, LAST_ACK, FIN_WAIT_2, TIME_WAIT, NA
linksourceip	<String>	Read-only	Source IP of the link connection.
linksourceport	<Integer>	Read-only	Source port of the link connection. Range 1 - 65535 * in CLI is represented as 65535 in NITRO API
linkdestip	<String>	Read-only	Destination IP of the link connection.
linkdestport	<Integer>	Read-only	Destination port of the link connection. Range 1 - 65535 * in CLI is represented as 65535 in NITRO API
linkservicetype	<String>	Read-only	Protocol supported by the link connection. Possible values = HTTP, FTP, TCP, UDP, SSL, SSL_BRIDGE,

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
			SSL_TCP, NNTP, RPCSVR, DNS, ADNS, SNMP, RTSP, DHCPRA, ANY, SIP_UDP, SIP_TCP, SIP_SSL, DNS_TCP, ADNS_TCP, HTTPSVR, HTTPCLIENT, NAT, HA, AAA, SINCTCP, VPN AFTP, MONITORS, SSLVPN UDP, SINCUDP, RIP, UDP CLNT, SASP, RPCCLNTS, ROUTE, AUDIT, STA HTTP, STA SSL, DNS RESOLVE, RDP, MYSQL, MSSQL, DIAMETER, SSL_DIAMETER, TFTP, ORACLE, ICA, RADIUS, RADIUSListener, SMPP, PPTP, GRE, SYSLOGTCP, SYSLOGUDP, FIX, SSL_FIX, QUIC, TFTP_DATA, USER_TCP, USER_SSL_TCP

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
linkidletime	<Integer>	Read-only	Time since last activity was detected on link connection.
linkstate	<String>	Read-only	TCP/IP current state of link connection. Possible values = CLOSED, LISTEN, SYN_SENT, SYN_RECEIVED, ESTABLISHED, CLOSE_WAIT, FIN_WAIT_1, CLOSING, LAST_ACK, FIN_WAIT_2, TIME_WAIT, NA
entityname	<String>	Read-only	Citrix ADC entity name for the connection.
linkentityname	<String>	Read-only	Citrix ADC entity name for link connection.
connid	<Double>	Read-only	Unique transaction number for the connection.
linkconnid	<Double>	Read-only	Unique transaction number for the peer connection.
connproperties	<String[]>	Read-only	flags used to store connection properties like client, server etc.

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
			Possible values = LINK, HASLINK, CLIENT, SERVER, NNM, MPTCP, SUBFLOW
optionflags	<String[]>	Read-only	flags used to store TCP options like Sack, WS. Possible values = sack, timstmp, ws
nswsvalue	<Double>	Read-only	Citrix ADC window scaling value.
peerwsvalue	<Double>	Read-only	peer window scaling value.
mss	<Integer>	Read-only	Client side MSS for the connection - used in server SYN.
retxretrycnt	<Integer>	Read-only	Retransmission retry count for the connection.
rcvwnd	<Integer>	Read-only	Received Advertised Window for the connection.
advwnd	<Integer>	Read-only	Sent advertised window for the connection.
sndcwnd	<Integer>	Read-only	sent congestion window for the connection.
iss	<Double>	Read-only	Initial send sequence number for the connection.

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
irs	<Double>	Read-only	Initial receive sequence number for the connection.
rcvnxt	<Integer>	Read-only	next expecting seq number for the connection.
maxack	<Integer>	Read-only	current running max ack sent for the connection.
sndnxt	<Integer>	Read-only	next bytes seq number for the connection.
sndunack	<Integer>	Read-only	Most recently received ACK for the connection.
httpendseq	<Integer>	Read-only	HTTP parsing tracking seq number for the connection.
httpstate	<String>	Read-only	HTTP Protocol state for the connection. Possible values = INITIALIZED, CONTENT_LENGTH, CHUNKED, WAIT_FINAL_ACK, UNKNOWN, DONE, WAIT_FIN
trcount	<Integer>	Read-only	Max reuests allowed per connection.
priority	<String>	Read-only	priority of the connection.

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
			Possible values = SC Priority, Priority queue1, priority queue2, priority queue3, DoS Priority, Surge Priority
httpreqver	<String>	Read-only	current HTTP request version on the connection. Possible values = HTTP_1_0, HTTP_1_1, HTTP_0_9, RTSP_1_0, SIP_2_0, VPN_ICA_SOCKS, VPN_ICA_CGP, HTTP_2_0
httprequest	<String>	Read-only	current HTTP request type on the connection. Possible values = OPTIONS, GET, HEAD, POST, PUT, DELETE, TRACE, TRACK, CONNECT, RPCCONNECT
httprspcode	<Integer>	Read-only	current response type on the connection.
rttsmoothed	<Integer>	Read-only	smoothed RTT value of the connection.

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
rttvariance	<Integer>	Read-only	RTT variance for the connection.
outoforderpkts	<Integer>	Read-only	held packets on the connection.
linkoptionflag	<String[]>	Read-only	Link connection's TCP option flag for Sack and WS. Possible values = sack, timstmp, ws
linknswsvalue	<Double>	Read-only	Link connection-s Citrix ADC window scaling value.
linkpeerwsvalue	<Double>	Read-only	Link connection-s peer Citrix ADC window scaling value.
targetnodeidnnm	<Integer>	Read-only	NNM connection target node ID.
sourcenodeidnnm	<Integer>	Read-only	NNM connection source node ID.
channelidnnm	<Double>	Read-only	NNM connection channel ID.
msgversionnnm	<Double>	Read-only	nnm message version.
td	<Double>	Read-only	Traffic Domain Id.
maxrcvbuf	<Integer>	Read-only	Maximum receive window that application advertizes to peer.

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
linkmaxrcvbuf	<Integer>	Read-only	Maximum receive window that application advertizes to peer in linked connection.
rxqsize	<Integer>	Read-only	Total number of bytes in Citrix ADC receive buffer. This includes bytes being processed / policy related data / stored in application buffer.
linkrxqsize	<Integer>	Read-only	Total number of bytes in Citrix ADC receive buffer for linked connection. This includes bytes being processed / policy related data / stored in application buffer.
maxsndbuf	<Integer>	Read-only	Maximum send window that application can process and send.
linkmaxsndbuf	<Integer>	Read-only	Maximum send window that application can process and send in linked connection.
txqsize	<Integer>	Read-only	Total number of bytes in Citrix ADC send buffer. This includes both inflight and queued bytes in Citrix ADC.

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
linktxqsize	<Integer>	Read-only	Total number of bytes in Citrix ADC send buffer for linked connection. This includes both inflight and queued bytes in Citrix ADC.
flavor	<String>	Read-only	TCP congestion control algorithm. Possible values = Default, Westwood, BIC, CUBIC, Nile
linkflavor	<String>	Read-only	TCP congestion control algorithm for a linked connection. Possible values = Default, Westwood, BIC, CUBIC, Nile
bwestimate	<Integer>	Read-only	TCP Bandwidth Estimate.
linkbwestimate	<Integer>	Read-only	TCP Bandwidth Estimate for a linked connection.
rttmin	<Integer>	Read-only	Minimum Round Trip Time for the connection.
linkrttmin	<Integer>	Read-only	Minimum Round Trip Time for linked connection.
name	<String>	Read-only	Name of TCP profile attached to the connection.

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
			Minimum length = 1 Maximum length = 127
linkname	<String>	Read-only	Name of TCP profile attached to the connection. Minimum length = 1 Maximum length = 127
tcpmode	<String>	Read-only	TCP Optimization modes TRANSPARENT / ENDPOINT. Possible values = TRANSPARENT, ENDPOINT
linktcpmode	<String>	Read-only	TCP Optimization modes TRANSPARENT / ENDPOINT for linked connection. Possible values = TRANSPARENT, ENDPOINT
realtimertt	<Integer>	Read-only	Real Time / Instantaneous round trip time.
linkrealtimertt	<Integer>	Read-only	Real Time / Instantaneous round trip time for linked connection.
sndbuf	<Integer>	Read-only	send buffer size.

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
linksndbuf	<Integer>	Read-only	Send buffer size for linked connection.
nsbtcpwaitq	<Integer>	Read-only	Number of packets in TCP wait queue.
linknsbtcpwaitq	<Integer>	Read-only	Number of packets in wait queue for linked connection.
nsbretxq	<Integer>	Read-only	Number of packets in retransmission queue.
linknsbretxq	<Integer>	Read-only	Number of packets in retransmission queue for linked connection.
sackblocks	<Integer>	Read-only	Number of sack blocks attached to the connection.
linksackblocks	<Integer>	Read-only	Number of sack blocks attached in linked connection.
congstate	<String>	Read-only	TCP congestion state. Possible values = open, recovery, loss, reneg, partial_ACK, retx_lost
linkcongstate	<String>	Read-only	TCP congestion state for a linked connection. Possible values = open, recovery,

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
			loss, reneg, partial_ACK, retx_lost
sndrecoverle	<Double>	Read-only	Sequence Number denoting end of fast recovery.
linksndrecoverle	<Double>	Read-only	Sequence Number denoting end of fast recovery for linked connection.
creditsinbytes	<Integer>	Read-only	Connections current credits in Bytes/ms.
linkcredits	<Integer>	Read-only	Link connections current credits in Bytes/ms.
rateinbytes	<Double>	Read-only	Connections current rate in Bytes/ms.
linkrateinbytes	<Double>	Read-only	Link connections current rate in Bytes/ms.
rateschedulerqueue	<Double>	Read-only	Bytes that are queued in the rate scheduler for this connection.
linkrateschedulerqueue	<Double>	Read-only	Bytes that are queued in the rate scheduler for link connection.
burstratecontrol	<String>	Read-only	TCP Burst Rate Control DISABLED/FIXED/DYNAMIC.

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
			Possible values = DISABLED, FIXED, DYNAMIC
linkburstratecontrol	<String>	Read-only	TCP Burst Rate Control DISABLED/ FIXED/DYNAMIC. Possible values = DISABLED, FIXED, DYNAMIC
cqabifavg	<Double>	Read-only	Average bytes in flight over maxburst.
cqathruputavg	<Double>	Read-only	Average instantaneous throughput over maxburst.
cqarcvwndavg	<Double>	Read-only	Average receive window over maxburst.
cqaiai1mspct	<Double>	Read-only	Percentage of inter arriaval samples for 1ms over maxburst.
cqaiai2mspct	<Double>	Read-only	Percentage of inter arriaval samples for 2ms over maxburst.
cqasamples	<Double>	Read-only	Number of ACK samples over maxburst.
cqaiaisamples	<Double>	Read-only	Number of samples considered for calculating inter arrival interval over maxburst.

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
cqanetclass	<String>	Read-only	Network type name. Maximum length = 49
cqaccl	<Double>	Read-only	Connection congestion level. Minimum value = 0 Maximum value = 100
cqacsq	<Double>	Read-only	Connectiontion signal quality level. Minimum value = 0 Maximum value = 100
cqaiaiavg	<Double>	Read-only	Average inter arriaval interval over maxburst.
cqaisiavg	<Double>	Read-only	Average inter send interval over maxburst.
cqarcvwndmin	<Double>	Read-only	Minimum receive window over maxburst.
cqaretxcorr	<Double>	Read-only	Number of retransmission due to noise over maxburst.
cqaretxcong	<Double>	Read-only	Number of retransmission due to congestion over maxburst.

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
cqaretxpackets	<Double>	Read-only	Number of retransmission over maxburst.
cqaloaddelayavg	<Double>	Read-only	Average load delay over maxburst.
cqanoisedelayavg	<Double>	Read-only	Average noise delay over maxburst.
cqarttmax	<Double>	Read-only	Maximum RTT sample over maxburst in ms.
cqarttmin	<Double>	Read-only	Minimum RTT sample over maxburst in ms.
cqarttavg	<Double>	Read-only	Minimum RTT sample over maxburst in ms.
adaptivetcpprofname	<String>	Read-only	Name of the Adaptive-TCP profile currently assigned to the connection. Minimum length = 1 Maximum length = 127
outoforderblocks	<Integer>	Read-only	held packets on the connection.
outoforderflushedcount	<Integer>	Read-only	held packets on the connection.
outoforderbytes	<Integer>	Read-only	held packets on the connection.
__count	<Double>	Read-only	count parameter

Operations

(click to see Properties)

GET (ALL)| COUNT

Some options that you can use for each operations:

- **Getting warnings in response:**NITRO allows you to get warnings in an operation by specifying the "warning" query parameter as "yes". For example, to get warnings while connecting to the Citrix ADC appliance, the URL is as follows:

  http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/login?warning=yes

  If any, the warnings are displayed in the response payload with the HTTP code "209 X-NITRO-WARNING".

- **Authenticated access for individual NITRO operations:**NITRO allows you to logon to the Citrix ADC appliance to perform individual operations. You can use this option instead of creating a NITRO session (using the login object) and then using that session to perform all operations,

  To do this, you must specify the username and password in the request header of the NITRO request as follows:

  X-NITRO-USER:<username>

  X-NITRO-PASS:<password>

  Note: In such cases, make sure that the request header DOES not include the following:

  Cookie:NITRO_AUTH_TOKEN=<tokenvalue>

`*Note:`*

Mandatory parameters are marked in redand placeholder content is marked in <green>.

get (all)

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/nsconnectiontable

Query-parameters:

args

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/nsconnectiontable?**args=filterexpression:<String_value>,link:<Boolean_value>,filtername:<Boolean_value>,detail:<String_value>,listen:<Boolean_value>,nodeid:<Double_value>**

Use this query-parameter to get nsconnectiontable resources based on additional properties.

attrs

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/nsconnectiontable?**attrs=property-name1,property-name2**

Use this query parameter to specify the resource details that you want to retrieve.

filter

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/nsconnectiontable?**filter=property-name1:property-val1,property-name2:property-val2**

Use this query-parameter to get the filtered set of nsconnectiontable resources configured on Citrix ADC. Filtering can be done on any of the properties of the resource.

view

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/nsconnectiontable?**view=summary**

Note: By default, the retrieved results are displayed in detail view (?view=detail).

pagination

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/nsconnectiontable?**pagesize=#no;pageno=#no**

Use this query-parameter to get the nsconnectiontable resources in chunks.

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "nsconnectiontable": [ {

filterexpression:<String_value>,link:<Boolean_value>,filtername:<Boolean_v
alue>,detail:<String[]_value>,listen:<Boolean_value>,nodeid:<Double_value>
"sourceip":<String_value>,

"sourceport":<Integer_value>,

"destip":<String_value>,

"destport":<Integer_value>,

"svctype":<String_value>,

"idletime":<Integer_value>,

"state":<String_value>,

"linksourceip":<String_value>,

"linksourceport":<Integer_value>,

"linkdestip":<String_value>,

"linkdestport":<Integer_value>,

"linkservicetype":<String_value>,

"linkidletime":<Integer_value>,

"linkstate":<String_value>,

"entityname":<String_value>,

"linkentityname":<String_value>,

"connectionnumber":<Integer_value>,
```

```
"linkconnectionnumber":<Integer_value>,

"connid":<Double_value>,

"linkconnid":<Double_value>,

"connproperties":<String[]_value>,

"optionflags":<String[]_value>,

"nswsvalue":<Double_value>,

"peerwsvalue":<Double_value>,

"mss":<Integer_value>,

"retxretrycnt":<Integer_value>,

"rcvwnd":<Integer_value>,

"advwnd":<Integer_value>,

"sndcwnd":<Integer_value>,

"iss":<Double_value>,

"irs":<Double_value>,

"rcvnxt":<Integer_value>,

"maxack":<Integer_value>,

"sndnxt":<Integer_value>,

"sndunack":<Integer_value>,

"httpendseq":<Integer_value>,

"httpstate":<String_value>,

"trcount":<Integer_value>,

"priority":<String_value>,
```

```
"httpreqver":<String_value>,

"httprequest":<String_value>,

"httprspcode":<Integer_value>,

"rttsmoothed":<Integer_value>,

"rttvariance":<Integer_value>,

"outoforderpkts":<Integer_value>,

"linkoptionflag":<String[]_value>,

"linknswsvalue":<Double_value>,

"linkpeerwsvalue":<Double_value>,

"targetnodeidnnm":<Integer_value>,

"sourcenodeidnnm":<Integer_value>,

"channelidnnm":<Double_value>,

"msgversionnnm":<Double_value>,

"td":<Double_value>,

"maxrcvbuf":<Integer_value>,

"linkmaxrcvbuf":<Integer_value>,

"rxqsize":<Integer_value>,

"linkrxqsize":<Integer_value>,

"maxsndbuf":<Integer_value>,

"linkmaxsndbuf":<Integer_value>,

"txqsize":<Integer_value>,

"linktxqsize":<Integer_value>,
```

```
"flavor":<String_value>,

"linkflavor":<String_value>,

"bwestimate":<Integer_value>,

"linkbwestimate":<Integer_value>,

"rttmin":<Integer_value>,

"linkrttmin":<Integer_value>,

"name":<String_value>,

"linkname":<String_value>,

"tcpmode":<String_value>,

"linktcpmode":<String_value>,

"realtimertt":<Integer_value>,

"linkrealtimertt":<Integer_value>,

"sndbuf":<Integer_value>,

"linksndbuf":<Integer_value>,

"nsbtcpwaitq":<Integer_value>,

"linknsbtcpwaitq":<Integer_value>,

"nsbretxq":<Integer_value>,

"linknsbretxq":<Integer_value>,

"sackblocks":<Integer_value>,

"linksackblocks":<Integer_value>,

"congstate":<String_value>,

"linkcongstate":<String_value>,
```

```
"sndrecoverle":<Double_value>,

"linksndrecoverle":<Double_value>,

"creditsinbytes":<Integer_value>,

"linkcredits":<Integer_value>,

"rateinbytes":<Double_value>,

"linkrateinbytes":<Double_value>,

"rateschedulerqueue":<Double_value>,

"linkrateschedulerqueue":<Double_value>,

"burstratecontrol":<String_value>,

"linkburstratecontrol":<String_value>,

"cqabifavg":<Double_value>,

"cqathruputavg":<Double_value>,

"cqarcvwndavg":<Double_value>,

"cqaiai1mspct":<Double_value>,

"cqaiai2mspct":<Double_value>,

"cqasamples":<Double_value>,

"cqaiaisamples":<Double_value>,

"cqanetclass":<String_value>,

"cqaccl":<Double_value>,

"cqacsq":<Double_value>,

"cqaiaiavg":<Double_value>,

"cqaisiavg":<Double_value>,
```

```
"cqarcvwndmin":<Double_value>,

"cqaretxcorr":<Double_value>,

"cqaretxcong":<Double_value>,

"cqaretxpackets":<Double_value>,

"cqaloaddelayavg":<Double_value>,

"cqanoisedelayavg":<Double_value>,

"cqarttmax":<Double_value>,

"cqarttmin":<Double_value>,

"cqarttavg":<Double_value>,

"adaptivetcpprofname":<String_value>,

"outoforderblocks":<Integer_value>,

"outoforderflushedcount":<Integer_value>,

"outoforderbytes":<Integer_value>

}]}
```

count

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/nsconnectiontable?**count=yes**

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "nsconnectiontable": [ { "__count": "#no"} ] }
```

nsconsoleloginprompt

Configuration for console prompt resource.

Properties

(click to see Operations)

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
promptstring	<String>	Read-write	Console login prompt string.

Operations

(click to see Properties)

UPDATE| UNSET| GET (ALL)

Some options that you can use for each operations:

- **Getting warnings in response:**NITRO allows you to get warnings in an operation by specifying the "warning" query parameter as "yes". For example, to get warnings while connecting to the Citrix ADC appliance, the URL is as follows:

  http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/login?warning=yes

  If any, the warnings are displayed in the response payload with the HTTP code "209 X-NITRO-WARNING".

- **Authenticated access for individual NITRO operations:**NITRO allows you to logon to the Citrix ADC appliance to perform individual operations. You can use this option instead of creating a NITRO session (using the login object) and then using that session to perform all operations,

  To do this, you must specify the username and password in the request header of the NITRO request as follows:

  X-NITRO-USER:<username>

  X-NITRO-PASS:<password>

  Note: In such cases, make sure that the request header DOES not include the following:

  Cookie:NITRO_AUTH_TOKEN=<tokenvalue>

`*Note:`*

Mandatory parameters are marked in redand placeholder content is marked in <green>.

update

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/nsconsoleloginprompt

HTTP Method:PUT

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Content-Type:application/json

Request Payload:

```
{"nsconsoleloginprompt":{

"promptstring":<String_value>

}}
```

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error

unset

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/nsconsoleloginprompt?**action=unset**

HTTP Method:POST

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Content-Type:application/json

Request Payload:

```
{"nsconsoleloginprompt":{

"promptstring":true

}}
```

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error

get (all)

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/nsconsoleloginprompt

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "nsconsoleloginprompt": [ {

"promptstring":<String_value>

}]}
```

nscqaparam

Configuration for cqaparam resource.

Properties

(click to see Operations)

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
harqretxdelay	<Double>	Read-write	HARQ retransmission delay (in ms). Default value: 0 Minimum value = 1 Maximum value = 64000
net1label	<String>	Read-write	Name of the network label. Maximum length = 15

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
minrttnet1	<Double>	Read-write	MIN RTT (in ms) for the first network. Default value: 0 Minimum value = 0 Maximum value = 6400
lr1coeflist	<String>	Read-write	coefficients values for Label1.
lr1probthresh	<Double>	Read-write	Probability threshold values for LR model to differentiate between NET1 and reset(NET2 and NET3). Default value: 0 Minimum value = 0 Maximum value = 1
net1cclscale	<String>	Read-write	Three congestion level scores limits corresponding to None, Low, Medium.
net1csqscale	<String>	Read-write	Three signal quality level scores limits corresponding to Excellent, Good, Fair.
net1logcoef	<String>	Read-write	Connection quality ranking Log coefficients of network 1.

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
net2label	<String>	Read-write	Name of the network label 2. Maximum length = 15
minrttnet2	<Double>	Read-write	MIN RTT (in ms) for the second network. Default value: 0 Minimum value = 0 Maximum value = 6400
lr2coeflist	<String>	Read-write	coefficients values for Label 2.
lr2probthresh	<Double>	Read-write	Probability threshold values for LR model to differentiate between NET2 and NET3. Default value: 0 Minimum value = 0 Maximum value = 1
net2cclscale	<String>	Read-write	Three congestion level scores limits corresponding to None, Low, Medium.
net2csqscale	<String>	Read-write	Three signal quality level scores limits corresponding to Excellent, Good, Fair.
net2logcoef	<String>	Read-write	Connnection quality ranking Log

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
			coefficients of network 2.
net3label	<String>	Read-write	Name of the network label 3. Maximum length = 15
minrttnet3	<Double>	Read-write	MIN RTT (in ms) for the third network. Default value: 0 Minimum value = 0 Maximum value = 6400
net3cclscale	<String>	Read-write	Three congestion level scores limits corresponding to None, Low, Medium.
net3csqscale	<String>	Read-write	Three signal quality level scores limits corresponding to Excellent, Good, Fair.
net3logcoef	<String>	Read-write	Connection quality ranking Log coefficients of network 3.

Operations

(click to see Properties)

UPDATE| UNSET| GET (ALL)

Some options that you can use for each operations:

- **Getting warnings in response:**NITRO allows you to get warnings in an operation by specifying the "warning" query parameter as "yes". For example, to get warnings while connecting to the Citrix ADC appliance, the URL is as follows:

  http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/login?warning=yes

  If any, the warnings are displayed in the response payload with the HTTP code "209 X-NITRO-WARNING".

- **Authenticated access for individual NITRO operations:**NITRO allows you to logon to the Citrix ADC appliance to perform individual operations. You can use this option instead of creating a NITRO session (using the login object) and then using that session to perform all operations,

  To do this, you must specify the username and password in the request header of the NITRO request as follows:

  X-NITRO-USER:<username>

  X-NITRO-PASS:<password>

  Note: In such cases, make sure that the request header DOES not include the following:

  Cookie:NITRO_AUTH_TOKEN=<tokenvalue>

`*Note:`*

Mandatory parameters are marked in redand placeholder content is marked in <green>.

update

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/nscqaparam

HTTP Method:PUT

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Content-Type:application/json

Request Payload:

```
{"nscqaparam":{
```

```
"harqretxdelay":<Double_value>,

"net1label":<String_value>,

"minrttnet1":<Double_value>,

"lr1coeflist":<String_value>,

"lr1probthresh":<Double_value>,

"net1cclscale":<String_value>,

"net1csqscale":<String_value>,

"net1logcoef":<String_value>,

"net2label":<String_value>,

"minrttnet2":<Double_value>,

"lr2coeflist":<String_value>,

"lr2probthresh":<Double_value>,

"net2cclscale":<String_value>,

"net2csqscale":<String_value>,

"net2logcoef":<String_value>,

"net3label":<String_value>,

"minrttnet3":<Double_value>,

"net3cclscale":<String_value>,

"net3csqscale":<String_value>,

"net3logcoef":<String_value>

}}
```

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error

unset

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/nscqaparam?**action=unset**

HTTP Method:POST

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Content-Type:application/json

Request Payload:

```
{"nscqaparam":{

"harqretxdelay":true,

"net1label":true,

"minrttnet1":true,

"lr1coeflist":true,

"lr1probthresh":true,

"net1cclscale":true,

"net1csqscale":true,

"net1logcoef":true,

"net2label":true,

"minrttnet2":true,
```

```
"net2cclscale":true,

"net2csqscale":true,

"net2logcoef":true,

"net3label":true,

"minrttnet3":true,

"net3cclscale":true,

"net3csqscale":true,

"net3logcoef":true

}}
```

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error

get (all)

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/nscqaparam

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "nscqaparam": [ {
"harqretxdelay":<Double_value>,
"net1label":<String_value>,
"net2label":<String_value>,
"minrttnet1":<Double_value>,
"lr1coeflist":<String_value>,
"lr1probthresh":<Double_value>,
"net1cclscale":<String_value>,
"net1csqscale":<String_value>,
"net1logcoef":<String_value>,
"minrttnet2":<Double_value>,
"lr2coeflist":<String_value>,
"lr2probthresh":<Double_value>,
"net2cclscale":<String_value>,
"net2csqscale":<String_value>,
"net2logcoef":<String_value>,
"net3label":<String_value>,
"minrttnet3":<Double_value>,
"net3cclscale":<String_value>,
"net3csqscale":<String_value>,
"net3logcoef":<String_value>
```

```
}]}
```

nsdhcpip

Configuration for dhcp IP resource.

Operations

(click to see Properties)

RELEASE

Some options that you can use for each operations:

- **Getting warnings in response:**NITRO allows you to get warnings in an operation by specifying the "warning" query parameter as "yes". For example, to get warnings while connecting to the Citrix ADC appliance, the URL is as follows:

  http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/login?warning=yes

  If any, the warnings are displayed in the response payload with the HTTP code "209 X-NITRO-WARNING".
- **Authenticated access for individual NITRO operations:**NITRO allows you to logon to the Citrix ADC appliance to perform individual operations. You can use this option instead of creating a NITRO session (using the login object) and then using that session to perform all operations,

  To do this, you must specify the username and password in the request header of the NITRO request as follows:

  X-NITRO-USER:<username>

  X-NITRO-PASS:<password>

  Note: In such cases, make sure that the request header DOES not include the following:

  Cookie:NITRO_AUTH_TOKEN=<tokenvalue>

`*Note:` *

Mandatory parameters are marked in redand placeholder content is marked in <green>.

release

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/nsdhcpip?**action=release**

HTTP Method:POST

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Content-Type:application/json

Request Payload:

```
{"nsdhcpip":{

}}
```

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error

nsdhcpparams

Configuration for DHCP parameters resource.

Properties

(click to see Operations)

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
dhcpclient	<String>	Read-write	Enables DHCP client to acquire IP address from the DHCP server in the next boot. When set to OFF, disables the DHCP client in the next boot.

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
			Default value: OFF Possible values = ON, OFF
saveroute	<String>	Read-write	DHCP acquired routes are saved on the Citrix ADC. Default value: OFF Possible values = ON, OFF
ipaddress	<String>	Read-only	DHCP acquired IP.
netmask	<String>	Read-only	DHCP acquired Netmask.
hostrtgw	<String>	Read-only	DHCP acquired Gateway.
running	<Boolean>	Read-only	DHCP mode.

Operations

(click to see Properties)

UPDATE| UNSET| GET (ALL)

Some options that you can use for each operations:

- **Getting warnings in response:**NITRO allows you to get warnings in an operation by specifying the "warning" query parameter as "yes". For example, to get warnings while connecting to the Citrix ADC appliance, the URL is as follows:

  http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/login?warning=yes

  If any, the warnings are displayed in the response payload with the HTTP code "209 X-NITRO-WARNING".

- **Authenticated access for individual NITRO operations:**NITRO allows you to logon to the Citrix ADC appliance to perform individual operations. You can use this option instead of

creating a NITRO session (using the login object) and then using that session to perform all operations,

To do this, you must specify the username and password in the request header of the NITRO request as follows:

X-NITRO-USER:<username>

X-NITRO-PASS:<password>

Note: In such cases, make sure that the request header DOES not include the following:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue>

`*Note:`*

Mandatory parameters are marked in redand placeholder content is marked in <green>.

update

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/nsdhcpparams

HTTP Method:PUT

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Content-Type:application/json

Request Payload:

```
{"nsdhcpparams":{

"dhcpclient":<String_value>,

"saveroute":<String_value>

}}
```

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error

unset

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/nsdhcpparams?**action=unset**

HTTP Method:POST

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Content-Type:application/json

Request Payload:

```
{"nsdhcpparams":{

"dhcpclient":true,

"saveroute":true

}}
```

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error

get (all)

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/nsdhcpparams

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "nsdhcpparams": [ {

"dhcpclient":<String_value>,

"ipaddress":<String_value>,

"netmask":<String_value>,

"hostrtgw":<String_value>,

"running":<Boolean_value>,

"saveroute":<String_value>

}]}
```

nsdiameter

Configuration for Diameter Parameters resource.

Properties

(click to see Operations)

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
identity	<String>	Read-write	DiameterIdentity to be used by NS. DiameterIdentity is used to identify a Diameter node uniquely. Before

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
			setting up diameter configuration, Citrix ADC (as a Diameter node) MUST be assigned a unique DiameterIdentity. example => set ns diameter -identity netscaler.com Now whenever Citrix ADC needs to use identity in diameter messages. It will use 'netscaler.com' as Origin-Host AVP as defined in RFC3588 . Minimum length = 1
realm	<String>	Read-write	Diameter Realm to be used by NS. example => set ns diameter -realm com Now whenever Citrix ADC system needs to use realm in diameter messages. It will use 'com' as Origin-

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
			Realm AVP as defined in RFC3588 . Minimum length = 1
servercloseproposagation	<String>	Read-write	when a Server connection goes down, whether to close the corresponding client connection if there were requests pending on the server. Default value: NO Possible values = YES, NO
ownernode	<Double>	Read-write	ID of the cluster node for which the diameter id is set, can be configured only through CLIP. Default value: -1 Minimum value = 0 Maximum value = 31
__count	<Double>	Read-only	count parameter

Operations

(click to see Properties)

UPDATE| UNSET| GET (ALL)| GET| COUNT

Some options that you can use for each operations:

- **Getting warnings in response:**NITRO allows you to get warnings in an operation by specifying the "warning" query parameter as "yes". For example, to get warnings while connecting to the Citrix ADC appliance, the URL is as follows:

  http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/login?warning=yes

  If any, the warnings are displayed in the response payload with the HTTP code "209 X-NITRO-WARNING".

- **Authenticated access for individual NITRO operations:**NITRO allows you to logon to the Citrix ADC appliance to perform individual operations. You can use this option instead of creating a NITRO session (using the login object) and then using that session to perform all operations,

  To do this, you must specify the username and password in the request header of the NITRO request as follows:

  X-NITRO-USER:<username>

  X-NITRO-PASS:<password>

  Note: In such cases, make sure that the request header DOES not include the following:

  Cookie:NITRO_AUTH_TOKEN=<tokenvalue>

`*Note:`*

Mandatory parameters are marked in redand placeholder content is marked in <green>.

update

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/nsdiameter

HTTP Method:PUT

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Content-Type:application/json

Request Payload:

```
{"nsdiameter":{
```

```
"identity":<String_value>,

"realm":<String_value>,

"serverclosepropagation":<String_value>,

"ownernode":<Double_value>

}}
```

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error

unset

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/nsdiameter?**action=unset**

HTTP Method:POST

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Content-Type:application/json

Request Payload:

```
{"nsdiameter":{

"serverclosepropagation":true,

"ownernode":<Double_value>

}}
```

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error

get (all)

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/nsdiameter

Query-parameters:

attrs

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/nsdiameter?**attrs=property-name1,property-name2**

Use this query parameter to specify the resource details that you want to retrieve.

filter

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/nsdiameter?**filter=property-name1:property-val1,property-name2:property-val2**

Use this query-parameter to get the filtered set of nsdiameter resources configured on Citrix ADC. Filtering can be done on any of the properties of the resource.

view

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/nsdiameter?**view=summary**

Note: By default, the retrieved results are displayed in detail view (?view=detail).

pagination

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/nsdiameter?**pagesize=#no;pageno=#no**

Use this query-parameter to get the nsdiameter resources in chunks.

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "nsdiameter": [ {

"ownernode":<Double_value>,

"identity":<String_value>,

"realm":<String_value>,

"serverclosepropagation":<String_value>

}]}
```

get

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/nsdiameter/ownernode_value<Double>

Query-parameters:

attrs

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/nsdiameter/ownernode_value<Double>?**attrs=property-name1,property-name2**

Use this query parameter to specify the resource details that you want to retrieve.

view

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/nsdiameter/ownernode_value<Double>?**view=summary**

Note: By default, the retrieved results are displayed in detail view (?view=detail).

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "nsdiameter": [ {

"ownernode":<Double_value>,

"identity":<String_value>,

"realm":<String_value>,

"serverclosepropagation":<String_value>

}]}
```

count

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/nsdiameter?**count=yes**

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "nsdiameter": [ { "__count": "#no"} ] }
```

nsencryptionkey

Configuration for encryption key resource.

Properties

(click to see Operations)

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
name	<String>	Read-write	Key name. This follows the same syntax rules as other expression entity names: It must begin with an alpha character (A-Z or a-z) or an underscore (_). The rest of the characters must be alpha, numeric (0-9) or underscores. It cannot be re or xp (reserved for regular and XPath expressions). It cannot be an expression reserved

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
			word (e.g. SYS or HTTP). It cannot be used for an existing expression object (HTTP callout, patset, dataset, stringmap, or named expression). Minimum length = 1
method	<String>	Read-write	Cipher method to be used to encrypt and decrypt content. NONE - no encryption or decryption is performed The output of ENCRYPT() and DECRYPT() is the same as the input. RC4 - the RC4 stream cipher with a 128 bit (16 byte) key; RC4 is now considered insecure and should only be used if required by existing applciations. DES[-<mode>] - the Data Encryption Standard (DES)

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
			block cipher with a 64-bit (8 byte) key, with 56 data bits and 8 parity bits. DES is considered less secure than DES3 or AES so it should only be used if required by an existing applicastion. The optional mode is described below; DES without a mode is equivalent to DES-CBC. DES3[-<mode>] - the Triple Data Encryption Standard (DES) block cipher with a 192-bit (24 byte) key. The optional mode is described below; DES3 without a mode is equivalent to DES3-CBC. AES<keysize>[-<mode>] - the Advanced Encryption Standard block cipher, available with 128 bit (16

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
			byte), 192 bit (24 byte), and 256 bit (32 byte) keys. The optional mode is described below; AES<keysize> without a mode is equivalent to AES<keysize>-CBC. For a block cipher, the <mode> specifies how multiple blocks of plaintext are encrypted and how the Initialization Vector (IV) is used. Choices are CBC (Cipher Block Chaining) - Each block of plaintext is XORed with the previous ciphertext block, or IV for the first block, before being encrypted. Padding is required if the plaintext is not a multiple of the cipher block size. CFB (Cipher Feedback) - The previous ciphertext

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
			block, or the IV for the first block, is encrypted and the output is XORed with the current plaintext block to create the current ciphertext block. The 128-bit version of CFB is provided. Padding is not required. OFB (Output Feedback) - A keystream is generated by applying the cipher successfully to the IV and XORing the keystream blocks with the plaintext. Padding is not required. ECB (Electronic Codebook) - Each block of plaintext is independently encrypted. An IV is not used. Padding is required. This mode is considered less secure than the other modes because the same plaintext always

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
			produces the same encrypted text and should only be used if required by an existing application. Possible values = NONE, RC4, DES3, AES128, AES192, AES256, DES, DES-CBC, DES-CFB, DES-OFB, DES-ECB, DES3-CBC, DES3-CFB, DES3-OFB, DES3-ECB, AES128-CBC, AES128-CFB, AES128-OFB, AES128-ECB, AES192-CBC, AES192-CFB, AES192-OFB, AES192-ECB, AES256-CBC, AES256-CFB, AES256-OFB, AES256-ECB
keyvalue	<String>	Read-write	The hex-encoded key value. The length is determined by the cipher method: RC4 - 16 bytes

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
			DES - 8 bytes (all modes) DES3 - 24 bytes (all modes) AES128 - 16 bytes (all modes) AES192 - 24 bytes (all modes) AES256 - 32 bytes (all modes) Note that the keyValue will be encrypted when it it is saved.
padding	<String>	Read-write	Enables or disables the padding of plaintext to meet the block size requirements of block ciphers: ON - For encryption, PKCS5/7 padding is used, which appends n bytes of value n on the end of the plaintext to bring it to the cipher block lnegth. If the plaintext length is alraady a multiple of the block length, an additional block with

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
			bytes of value block_length will be added. For decryption, ISO 10126 padding is accepted, which expects the last byte of the block to be the number of added pad bytes. Note that this accepts PKCS5/7 padding, as well as ANSI_X923 padding. Padding ON is the default for the ECB and CBD modes. OFF - No padding. An Undef error will occur with the ECB or CBC modes if the plaintext length is not a multitple of the cipher block size. This can be used with the CFB and OFB modes, and with the ECB and CBC modes if the plaintext will always be an integral number of blocks, or if custom padding is implemented

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
			using a policy extension function. Padding OFf is the default for CFB and OFB modes. Default value: DEFAULT Possible values = OFF, ON
iv	<String>	Read-write	The initalization voector (IV) for a block cipher, one block of data used to initialize the encryption. The best practice is to not specify an IV, in which case a new random IV will be generated for each encryption. The format must be iv_data or keyid_iv_data to include the generated IV in the encrypted data. The IV should only be specified if it cannot be included in the encrypted data. The IV length is the cipher block size: RC4 - not used (error if IV is specified)

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
			DES - 8 bytes (all modes) DES3 - 8 bytes (all modes) AES128 - 16 bytes (all modes) AES192 - 16 bytes (all modes) AES256 - 16 bytes (all modes).
comment	<String>	Read-write	Comments associated with this encryption key.
__count	<Double>	Read-only	count parameter

Operations

(click to see Properties)

ADD| UPDATE| UNSET| DELETE| GET (ALL)| GET| COUNT

Some options that you can use for each operations:

- **Getting warnings in response:**NITRO allows you to get warnings in an operation by specifying the "warning" query parameter as "yes". For example, to get warnings while connecting to the Citrix ADC appliance, the URL is as follows:

  http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/login?warning=yes

  If any, the warnings are displayed in the response payload with the HTTP code "209 X-NITRO-WARNING".
- **Authenticated access for individual NITRO operations:**NITRO allows you to logon to the Citrix ADC appliance to perform individual operations. You can use this option instead of

creating a NITRO session (using the login object) and then using that session to perform all operations,

To do this, you must specify the username and password in the request header of the NITRO request as follows:

X-NITRO-USER:<username>

X-NITRO-PASS:<password>

Note: In such cases, make sure that the request header DOES not include the following:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue>

`*Note:`*

Mandatory parameters are marked in redand placeholder content is marked in <green>.

add

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/nsencryptionkey

HTTP Method:POST

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Content-Type:application/json

Request Payload:

```
{"nsencryptionkey":{

<b>"name":<String_value>,

</b><b>"method":<String_value>,

</b>"keyvalue":<String_value>,

"padding":<String_value>,

"iv":<String_value>,

"comment":<String_value>
```

```
}}
```

Response:

HTTP Status Code on Success: 201 Created HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error

update

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/nsencryptionkey

HTTP Method:PUT

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Content-Type:application/json

Request Payload:

```
{"nsencryptionkey":{

<b>"name":<String_value>,

</b>"method":<String_value>,

"keyvalue":<String_value>,

"padding":<String_value>,

"iv":<String_value>,

"comment":<String_value>

}}
```

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error

unset

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/nsencryptionkey?**action=unset**

HTTP Method:POST

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Content-Type:application/json

Request Payload:

```
{"nsencryptionkey":{

<b>"name":<String_value>,

</b>"padding":true,

"iv":true,

"comment":true

}}
```

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error

delete

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/nsencryptionkey/name_value<String>

HTTP Method:DELETE

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue>

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error

get (all)

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/nsencryptionkey

Query-parameters:

attrs

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/nsencryptionkey?**attrs=property-name1,property-name2**

Use this query parameter to specify the resource details that you want to retrieve.

filter

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/nsencryptionkey?**filter=property-name1:property-val1,property-name2:property-val2**

Use this query-parameter to get the filtered set of nsencryptionkey resources configured on Citrix ADC. Filtering can be done on any of the properties of the resource.

view

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/nsencryptionkey?**view=summary**

Note: By default, the retrieved results are displayed in detail view (?view=detail).

pagination

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/nsencryptionkey?**pagesize=#no;pageno=#no**

Use this query-parameter to get the nsencryptionkey resources in chunks.

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "nsencryptionkey": [ {

"name":<String_value>,

"method":<String_value>,

"padding":<String_value>,

"iv":<String_value>,

"comment":<String_value>

}]}
```

get

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/nsencryptionkey/name_value<String>

Query-parameters:

attrs

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/nsencryptionkey/name_value<String>?**attrs=property-name1,property-name2**

Use this query parameter to specify the resource details that you want to retrieve.

view

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/nsencryptionkey/name_value<String>?**view=summary**

Note: By default, the retrieved results are displayed in detail view (?view=detail).

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "nsencryptionkey": [ {

"name":<String_value>,

"method":<String_value>,

"padding":<String_value>,

"iv":<String_value>,

"comment":<String_value>

}]}
```

count

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/nsencryptionkey?**count=yes**

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "nsencryptionkey": [ { "__count": "#no"} ] }
```

nsencryptionparams

Configuration for default encryption parameters resource.

Properties

(click to see Operations)

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
method	<String>	Read-write	Cipher method (and key length) to be used to encrypt and decrypt content. The default value is AES256. Possible values = NONE, RC4, DES3, AES128, AES192, AES256, DES, DES-CBC, DES-CFB, DES-OFB, DES-ECB, DES3-

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
			CBC, DES3-CFB, DES3-OFB, DES3-ECB, AES128-CBC, AES128-CFB, AES128-OFB, AES128-ECB, AES192-CBC, AES192-CFB, AES192-OFB, AES192-ECB, AES256-CBC, AES256-CFB, AES256-OFB, AES256-ECB
keyvalue	<String>	Read-write	The base64-encoded key generation number, method, and key value. Note: * Do not include this argument if you are changing the encryption method. * To generate a new key value for the current encryption method, specify an empty string \(""\) as the value of this parameter. The parameter is passed implicitly, with its automatically

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
			generated value, to the Citrix ADC packet engines even when it is not included in the command. Passing the parameter to the packet engines enables the appliance to save the key value to the configuration file and to propagate the key value to the secondary appliance in a high availability setup.

Operations

(click to see Properties)

UPDATE| GET (ALL)

Some options that you can use for each operations:

- **Getting warnings in response:**NITRO allows you to get warnings in an operation by specifying the "warning" query parameter as "yes". For example, to get warnings while connecting to the Citrix ADC appliance, the URL is as follows:

  http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/login?warning=yes

  If any, the warnings are displayed in the response payload with the HTTP code "209 X-NITRO-WARNING".

- **Authenticated access for individual NITRO operations:**NITRO allows you to logon to the Citrix ADC appliance to perform individual operations. You can use this option instead of creating a NITRO session (using the login object) and then using that session to perform all operations,

To do this, you must specify the username and password in the request header of the NITRO request as follows:

X-NITRO-USER:<username>

X-NITRO-PASS:<password>

Note: In such cases, make sure that the request header DOES not include the following:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue>

`*Note:`*

Mandatory parameters are marked in redand placeholder content is marked in <green>.

update

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/nsencryptionparams

HTTP Method:PUT

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Content-Type:application/json

Request Payload:

```
{"nsencryptionparams":{

<b>"method":<String_value>,

</b>"keyvalue":<String_value>

}}
```

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error

get (all)

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/nsencryptionparams

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "nsencryptionparams": [ {

"method":<String_value>

}]}
```

nsevents

Configuration for events resource.

Properties

(click to see Operations)

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
eventno	<Double>	Read-write	Event number starting from which events must be shown.

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
time	<Double>	Read-only	Event no.
eventcode	<Double>	Read-only	event Code.
devid	<Double>	Read-only	Device Name.
devname	<String>	Read-only	Device Name.
text	<String>	Read-only	Event no.
data0	<Double>	Read-only	additional event information.
data1	<Double>	Read-only	additional event information.
data2	<Double>	Read-only	additional event information.
data3	<Double>	Read-only	additional event information.
__count	<Double>	Read-only	count parameter

Operations

(click to see Properties)

GET (ALL)| COUNT

Some options that you can use for each operations:

- **Getting warnings in response:**NITRO allows you to get warnings in an operation by specifying the "warning" query parameter as "yes". For example, to get warnings while connecting to the Citrix ADC appliance, the URL is as follows:

  http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/login?warning=yes

  If any, the warnings are displayed in the response payload with the HTTP code "209 X-NITRO-WARNING".

- **Authenticated access for individual NITRO operations:**NITRO allows you to logon to the Citrix ADC appliance to perform individual operations. You can use this option instead of creating a NITRO session (using the login object) and then using that session to perform all operations,

  To do this, you must specify the username and password in the request header of the NITRO request as follows:

  X-NITRO-USER:<username>

  X-NITRO-PASS:<password>

  Note: In such cases, make sure that the request header DOES not include the following:

  Cookie:NITRO_AUTH_TOKEN=<tokenvalue>

`*Note:`*

Mandatory parameters are marked in redand placeholder content is marked in <green>.

get (all)

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/nsevents

Query-parameters:

args

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/nsevents?**args=eventno:<Double_value>**

Use this query-parameter to get nsevents resources based on additional properties.

attrs

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/nsevents?**attrs=property-name1,property-name2**

Use this query parameter to specify the resource details that you want to retrieve.

filter

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/nsevents?**filter=property-name1:property-val1,property-name2:property-val2**

Use this query-parameter to get the filtered set of nsevents resources configured on Citrix ADC. Filtering can be done on any of the properties of the resource.

view

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/nsevents?**view=summary**

Note: By default, the retrieved results are displayed in detail view (?view=detail).

pagination

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/nsevents?**pagesize=#no;pageno=#no**

Use this query-parameter to get the nsevents resources in chunks.

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "nsevents": [ {

eventno:<Double_value>"time":<Double_value>,

"eventcode":<Double_value>,

"devid":<Double_value>,

"devname":<String_value>,

"text":<String_value>,

"data0":<Double_value>,

"data1":<Double_value>,
```

```
"data2":<Double_value>,

"data3":<Double_value>

}]}
```

count

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/nsevents?**count=yes**

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "nsevents": [ { "__count": "#no"} ] }
```

nsextension

Configuration for Extension resource.

Properties

(click to see Operations)

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
src	<String>	Read-write	Local path to and name of, or URL (protocol, host, path, and file name) for, the file in which to store the imported extension. NOTE: The import fails if the object to be imported is on an HTTPS server that requires client certificate authentication for access. Minimum length = 1 Maximum length = 2047
name	<String>	Read-write	Name to assign to the extension object on the Citrix ADC. Minimum length = 1 Maximum length = 31
comment	<String>	Read-write	Any comments to preserve information about the extension object. Maximum length = 128
overwrite	<Boolean>	Read-write	Overwrites the existing file.

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
trace	<String>	Read-write	Enables tracing to the NS log file of extension execution: off - turns off tracing (equivalent to unset ns extension <extension-name> -trace) calls - traces extension function calls with arguments and function returns with the first return value lines - traces the above plus line numbers for executed extension lines all - traces the above plus local variables changed by executed extension lines Note that the DEBUG log level must be enabled to see extension tracing. This can be done by set audit syslogParams

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
			-loglevel ALL or -loglevel DEBUG. Default value: off Possible values = off, calls, lines, all
tracefunctions	<String>	Read-write	Comma-separated list of extension functions to trace. By default, all extension functions are traced. Maximum length = 256
tracevariables	<String>	Read-write	Comma-separated list of variables (in traced extension functions) to trace. By default, all variables are traced. Maximum length = 256
detail	<String>	Read-write	Show detail for extension function. Possible values = brief, all
type	<String>	Read-only	. Possible values = WSDL, CustomSettings, XMLSchema, XMLErrorPage, htmlpage, CustomResp, Extension

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
functionhits	<Double>	Read-only	Number of time function evaluates successfully.
functionundefhits	<Double>	Read-only	Number of times error occured in evaluating extension function.
functionhaltcount	<Double>	Read-only	Number of time function evaluation is halted.
__count	<Double>	Read-only	count parameter

Operations

(click to see Properties)

IMPORT| DELETE| ADD| CHANGE| UPDATE| UNSET| GET (ALL)| GET| COUNT

Some options that you can use for each operations:

- **Getting warnings in response:**NITRO allows you to get warnings in an operation by specifying the "warning" query parameter as "yes". For example, to get warnings while connecting to the Citrix ADC appliance, the URL is as follows:

  http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/login?warning=yes

  If any, the warnings are displayed in the response payload with the HTTP code "209 X-NITRO-WARNING".

- **Authenticated access for individual NITRO operations:**NITRO allows you to logon to the Citrix ADC appliance to perform individual operations. You can use this option instead of creating a NITRO session (using the login object) and then using that session to perform all operations,

  To do this, you must specify the username and password in the request header of the NITRO request as follows:

  X-NITRO-USER:<username>

X-NITRO-PASS:<password>

Note: In such cases, make sure that the request header DOES not include the following:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue>

`*Note:`*

Mandatory parameters are marked in redand placeholder content is marked in <green>.

Import

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/nsextension?**action=Import**

HTTP Method:POST

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Content-Type:application/json

Request Payload:

```
{"nsextension":{

<b>"src":<String_value>,

</b><b>"name":<String_value>,

</b>"comment":<String_value>,

"overwrite":<Boolean_value>

}}
```

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error

delete

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/nsextension/name_value<String>

HTTP Method:DELETE

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue>

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error

add

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/nsextension

HTTP Method:POST

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Content-Type:application/json

Request Payload:

```
{"nsextension":{

<b>"name":<String_value>,

</b>"comment":<String_value>

}}
```

Response:

HTTP Status Code on Success: 201 Created HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error

change

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/nsextension?**action=update**

HTTP Method:POST

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Content-Type:application/json

Request Payload:

```
{"nsextension":{

<b>"name":<String_value>

</b>}}
```

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error

update

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/nsextension

HTTP Method:PUT

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Content-Type:application/json

Request Payload:

```
{"nsextension":{

<b>"name":<String_value>,
```

```
</b>"trace":<String_value>,

"tracefunctions":<String_value>,

"tracevariables":<String_value>,

"comment":<String_value>

}}
```

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error

unset

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/nsextension?**action=unset**

HTTP Method:POST

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Content-Type:application/json

Request Payload:

```
{"nsextension":{

<b>"name":<String_value>,

</b>"trace":true,

"tracefunctions":true,

"tracevariables":true,

"comment":true
```

```
}}
```

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error

get (all)

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/nsextension

Query-parameters:

args

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/nsextension?**args=name:<String_value>,detail:<String_value>**

Use this query-parameter to get nsextension resources based on additional properties.

attrs

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/nsextension?**attrs=property-name1,property-name2**

Use this query parameter to specify the resource details that you want to retrieve.

filter

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/nsextension?**filter=property-name1:property-val1,property-name2:property-val2**

Use this query-parameter to get the filtered set of nsextension resources configured on Citrix ADC. Filtering can be done on any of the properties of the resource.

view

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/nsextension?**view=summary**

Note: By default, the retrieved results are displayed in detail view (?view=detail).

pagination

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/nsextension?**pagesize=#no;pageno=#no**

Use this query-parameter to get the nsextension resources in chunks.

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "nsextension": [ {

name:<String_value>,detail:<String_value>"src":<String_value>,

"type":<String_value>,

"comment":<String_value>,

"functionhits":<Double_value>,

"functionundefhits":<Double_value>,

"functionhaltcount":<Double_value>,

"trace":<String_value>,

"tracefunctions":<String_value>,

"tracevariables":<String_value>

}]}
```

get

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/nsextension/name_value<String>

Query-parameters:

attrs

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/nsextension/name_value<String>?**attrs=property-name1,property-name2**

Use this query parameter to specify the resource details that you want to retrieve.

view

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/nsextension/name_value<String>?**view=summary**

Note: By default, the retrieved results are displayed in detail view (?view=detail).

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "nsextension": [ {

name:<String_value>,detail:<String_value>"src":<String_value>,

"type":<String_value>,
```

```
"comment":<String_value>,

"functionhits":<Double_value>,

"functionundefhits":<Double_value>,

"functionhaltcount":<Double_value>,

"trace":<String_value>,

"tracefunctions":<String_value>,

"tracevariables":<String_value>

}]}
```

count

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/nsextension?**count=yes**

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "nsextension": [ { "__count": "#no"} ] }
```

nsextension_binding

Binding object which returns the resources bound to nsextension.

Properties

(click to see Operations)

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
name	<String>	Read-write	Name of the extension object. Minimum length = 1 Maximum length = 31
nsextension_extensionfunction_binding	<nsextension_extensionfunction_binding[]>	Read-only	extensionfunction that can be bound to nsextension.

Operations

(click to see Properties)

GET| GET (ALL)

Some options that you can use for each operations:

- **Getting warnings in response:**NITRO allows you to get warnings in an operation by specifying the "warning" query parameter as "yes". For example, to get warnings while connecting to the Citrix ADC appliance, the URL is as follows:

  http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/login?warning=yes

  If any, the warnings are displayed in the response payload with the HTTP code "209 X-NITRO-WARNING".
- **Authenticated access for individual NITRO operations:**NITRO allows you to logon to the Citrix ADC appliance to perform individual operations. You can use this option instead of creating a NITRO session (using the login object) and then using that session to perform all operations,

  To do this, you must specify the username and password in the request header of the NITRO request as follows:

X-NITRO-USER:<username>

X-NITRO-PASS:<password>

Note: In such cases, make sure that the request header DOES not include the following:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue>

`*Note:`*

Mandatory parameters are marked in redand placeholder content is marked in <green>.

get

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/nsextension_binding/name_value<String>

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "nsextension_binding": [ {

"name":<String_value>,

"nsextension_extensionfunction_binding":<nsextension_extensionfunction_binding[]_value>

}]}
```

get (all)

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/nsextension_binding

Query-parameters:

bulkbindings

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/nsextension_binding?**bulkbindings=yes**

NITRO allows you to fetch bindings in bulk.

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "nsextension_binding": [ {

"name":<String_value>,

"nsextension_extensionfunction_binding":<nsextension_extensionfunction_binding[]_value>

}]}
```

nsextension_extensionfunction_binding

Binding object showing the extensionfunction that can be bound to nsextension.

Properties

(click to see Operations)

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
name	<String>	Read-write	Name of the extension object. Minimum length = 1 Maximum length = 31
extensionfunctionline number	<Double>	Read-only	Line number of the function in file.
extensionfunctionclasses	<String[]>	Read-only	List of classes (including inherited) that the function is present in.
extensionfunctionargtype	<String[]>	Read-only	List of extension function's arguments types.
activeextensionfunction	<Integer>	Read-only	Extension function is in use or not.
extensionfunctionname	<String>	Read-only	Name of extension function given in the extension. Minimum length = 1 Maximum length = 31
extensionfunctionclasstype	<String>	Read-only	Extension function class type. Minimum length = 1 Maximum length = 31

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
extensionfuncdescription	<String>	Read-only	Any description to preserve information about the extension function. Maximum length = 1023
extensionfunctionallparams	<String[]>	Read-only	List of parameters (including promotions) that the function can accept.
extensionfunctionargcount	<Double>	Read-only	Number of parameters in the extension function.
extensionfunctionreturntype	<String>	Read-only	Extension function return type. Minimum length = 1 Maximum length = 31
extensionfunctionclassescount	<Double>	Read-only	Number of classes the function is present in.
extensionfunctionallparamscount	<Double>	Read-only	Number of parameters (including promotions) that the function can accept.
__count	<Double>	Read-only	count parameter

Operations

(click to see Properties)

GET| GET (ALL)| COUNT

Some options that you can use for each operations:

- **Getting warnings in response:**NITRO allows you to get warnings in an operation by specifying the "warning" query parameter as "yes". For example, to get warnings while connecting to the Citrix ADC appliance, the URL is as follows:

  http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/login?warning=yes

  If any, the warnings are displayed in the response payload with the HTTP code "209 X-NITRO-WARNING".

- **Authenticated access for individual NITRO operations:**NITRO allows you to logon to the Citrix ADC appliance to perform individual operations. You can use this option instead of creating a NITRO session (using the login object) and then using that session to perform all operations,

  To do this, you must specify the username and password in the request header of the NITRO request as follows:

  X-NITRO-USER:<username>

  X-NITRO-PASS:<password>

  Note: In such cases, make sure that the request header DOES not include the following:

  Cookie:NITRO_AUTH_TOKEN=<tokenvalue>

`*Note:`*

Mandatory parameters are marked in redand placeholder content is marked in <green>.

get

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/nsextension_extensionfunction_binding/name_value<String>

Query-parameters:

filter

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/nsextension_extensionfunction_binding/name_value<String>?**filter=property-name1:property-value1,property-name2:property-value2**

Use this query-parameter to get the filtered set of nsextension_extensionfunction_binding resources configured on Citrix ADC. Filtering can be done on any of the properties of the resource.

pagination

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/nsextension_extensionfunction_binding/name_value<String>?**pagesize=#no;pageno=#no**

Use this query-parameter to get the nsextension_extensionfunction_binding resources in chunks.

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "nsextension_extensionfunction_binding": [ {

"name":<String_value>,

"extensionfunctionlinenumber":<Double_value>,

"extensionfunctionclasses":<String[]_value>,

"extensionfunctionargtype":<String[]_value>,

"activeextensionfunction":<Integer_value>,

"extensionfunctionname":<String_value>,

"extensionfunctionclasstype":<String_value>,

"extensionfuncdescription":<String_value>,

"extensionfunctionallparams":<String[]_value>,
```

```
"extensionfunctionargcount":<Double_value>,

"extensionfunctionreturntype":<String_value>,

"extensionfunctionclassescount":<Double_value>,

"extensionfunctionallparamscount":<Double_value>

}]}
```

get (all)

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/nsextension_extensionfunction_binding

Query-parameters:

bulkbindings

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/nsextension_extensionfunction_binding?**bulkbindings=yes**

NITRO allows you to fetch bindings in bulk.

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "nsextension_extensionfunction_binding": [ {
```

```
"name":<String_value>,

"extensionfunctionlinenumber":<Double_value>,

"extensionfunctionclasses":<String[]_value>,

"extensionfunctionargtype":<String[]_value>,

"activeextensionfunction":<Integer_value>,

"extensionfunctionname":<String_value>,

"extensionfunctionclasstype":<String_value>,

"extensionfuncdescription":<String_value>,

"extensionfunctionallparams":<String[]_value>,

"extensionfunctionargcount":<Double_value>,

"extensionfunctionreturntype":<String_value>,

"extensionfunctionclassescount":<Double_value>,

"extensionfunctionallparamscount":<Double_value>

}]}
```

count

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/nsextension_extensionfunction_binding/name_value<String>?**count=yes**

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{"nsextension_extensionfunction_binding": [ { "__count": "#no"} ] }
```

nsfeature

Configuration for feature resource.

Properties

(click to see Operations)

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
feature	<String[]>	Read-write	Feature to be enabled. Multiple features can be specified by providing a blank space between each feature.
wl	<Boolean>	Read-only	Web Logging.
sp	<Boolean>	Read-only	Surge Protection.
lb	<Boolean>	Read-only	Load Balancing.
cs	<Boolean>	Read-only	Content Switching.
cr	<Boolean>	Read-only	Cache Redirect.
sc	<Boolean>	Read-only	Sure Connect.

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
cmp	<Boolean>	Read-only	Compression.
pq	<Boolean>	Read-only	Priority Queuing.
ssl	<Boolean>	Read-only	Secure Sockets Layer.
gslb	<Boolean>	Read-only	Global Server Load Balancing.
hdosp	<Boolean>	Read-only	DOS Protection.
cf	<Boolean>	Read-only	Content Filter.
ic	<Boolean>	Read-only	Integrated Caching.
sslvpn	<Boolean>	Read-only	SSL VPN.
aaa	<Boolean>	Read-only	AAA.
ospf	<Boolean>	Read-only	OSPF Routing.
rip	<Boolean>	Read-only	RIP Routing.
bgp	<Boolean>	Read-only	BGP Routing.
rewrite	<Boolean>	Read-only	Rewrite.
ipv6pt	<Boolean>	Read-only	IPv6 protocol translation.
appfw	<Boolean>	Read-only	Application Firewall.
responder	<Boolean>	Read-only	Responder.
htmlinjection	<Boolean>	Read-only	HTML Injection.
push	<Boolean>	Read-only	Citrix ADC Push.
appflow	<Boolean>	Read-only	AppFlow.
cloudbridge	<Boolean>	Read-only	CloudBridge.

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
isis	<Boolean>	Read-only	ISIS Routing.
ch	<Boolean>	Read-only	Call Home.
appqoe	<Boolean>	Read-only	AppQoS.
contentaccelerator	<Boolean>	Read-only	Transparent Integrated Caching.
rise	<Boolean>	Read-only	RISE.
feo	<Boolean>	Read-only	Optimize Web content (HTML, CSS, JavaScript, images).
lsn	<Boolean>	Read-only	Large Scale NAT.
rdpproxy	<Boolean>	Read-only	RDPPROXY.
rep	<Boolean>	Read-only	Reputation Services.
urlfiltering	<Boolean>	Read-only	URL Filtering.
videooptimization	<Boolean>	Read-only	Video Optimization.
forwardproxy	<Boolean>	Read-only	Forward Proxy.
sslinterception	<Boolean>	Read-only	SSL Interception.
adaptivetcp	<Boolean>	Read-only	Adaptive TCP.
cqa	<Boolean>	Read-only	Connection Quality Analytics.
ci	<Boolean>	Read-only	Content Inspection.

Operations

(click to see Properties)

ENABLE| DISABLE| GET (ALL)

Some options that you can use for each operations:

- **Getting warnings in response:**NITRO allows you to get warnings in an operation by specifying the "warning" query parameter as "yes". For example, to get warnings while connecting to the Citrix ADC appliance, the URL is as follows:

  http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/login?warning=yes

  If any, the warnings are displayed in the response payload with the HTTP code "209 X-NITRO-WARNING".

- **Authenticated access for individual NITRO operations:**NITRO allows you to logon to the Citrix ADC appliance to perform individual operations. You can use this option instead of creating a NITRO session (using the login object) and then using that session to perform all operations,

  To do this, you must specify the username and password in the request header of the NITRO request as follows:

  X-NITRO-USER:<username>

  X-NITRO-PASS:<password>

  Note: In such cases, make sure that the request header DOES not include the following:

  Cookie:NITRO_AUTH_TOKEN=<tokenvalue>

`*Note:`*

Mandatory parameters are marked in redand placeholder content is marked in <green>.

enable

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/nsfeature?**action=enable**

HTTP Method:POST

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Content-Type:application/json

Request Payload:

```
{"nsfeature":{

"feature":<String[]_value>

}}
```

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error

disable

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/nsfeature?**action=disable**

HTTP Method:POST

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Content-Type:application/json

Request Payload:

```
{"nsfeature":{

"feature":<String[]_value>

}}
```

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error

get (all)

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/nsfeature

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "nsfeature": [ {

"feature":<String[]_value>,

"wl":<Boolean_value>,

"sp":<Boolean_value>,

"lb":<Boolean_value>,

"cs":<Boolean_value>,

"cr":<Boolean_value>,

"sc":<Boolean_value>,

"cmp":<Boolean_value>,

"pq":<Boolean_value>,

"ssl":<Boolean_value>,

"gslb":<Boolean_value>,
```

```
"hdosp":<Boolean_value>,

"routing":<Boolean_value>,

"cf":<Boolean_value>,

"ic":<Boolean_value>,

"sslvpn":<Boolean_value>,

"aaa":<Boolean_value>,

"ospf":<Boolean_value>,

"rip":<Boolean_value>,

"bgp":<Boolean_value>,

"rewrite":<Boolean_value>,

"ipv6pt":<Boolean_value>,

"appfw":<Boolean_value>,

"responder":<Boolean_value>,

"htmlinjection":<Boolean_value>,

"push":<Boolean_value>,

"appflow":<Boolean_value>,

"cloudbridge":<Boolean_value>,

"isis":<Boolean_value>,

"ch":<Boolean_value>,

"appqoe":<Boolean_value>,

"contentaccelerator":<Boolean_value>,

"rise":<Boolean_value>,
```

```
"feo":<Boolean_value>,

"lsn":<Boolean_value>,

"rdpproxy":<Boolean_value>,

"rep":<Boolean_value>,

"urlfiltering":<Boolean_value>,

"videooptimization":<Boolean_value>,

"forwardproxy":<Boolean_value>,

"sslinterception":<Boolean_value>,

"adaptivetcp":<Boolean_value>,

"cqa":<Boolean_value>,

"ci":<Boolean_value>

}]}
```

nshardware

Configuration for hardware resource.

Properties

(click to see Operations)

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
hwdescription	<String>	Read-only	Hardware and it's ports detail.
sysid	<Double>	Read-only	System id.
manufactureday	<Integer>	Read-only	Manufacturing day.

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
manufacturemonth	<Integer>	Read-only	Manufacturing month.
manufactureyear	<Integer>	Read-only	Manufacturing year.
cpufrequncy	<Integer>	Read-only	CPU Frequency.
hostid	<Integer>	Read-only	host id.
host	<String>	Read-only	host id.
serialno	<String>	Read-only	Serial no.
encodedserialno	<String>	Read-only	Encoded serial no.
netscaleruuid	<String>	Read-only	Citrix ADC UUID.
bmcrevision	<String>	Read-only	BMC Firmware Revision.

Operations

(click to see Properties)

GET (ALL)

Some options that you can use for each operations:

- **Getting warnings in response:**NITRO allows you to get warnings in an operation by specifying the "warning" query parameter as "yes". For example, to get warnings while connecting to the Citrix ADC appliance, the URL is as follows:

  http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/login?warning=yes

  If any, the warnings are displayed in the response payload with the HTTP code "209 X-NITRO-WARNING".

- **Authenticated access for individual NITRO operations:**NITRO allows you to logon to the Citrix ADC appliance to perform individual operations. You can use this option instead of creating a NITRO session (using the login object) and then using that session to perform all operations,

To do this, you must specify the username and password in the request header of the NITRO request as follows:

X-NITRO-USER:<username>

X-NITRO-PASS:<password>

Note: In such cases, make sure that the request header DOES not include the following:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue>

`*Note:`*

Mandatory parameters are marked in redand placeholder content is marked in <green>.

get (all)

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/nshardware

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "nshardware": [ {

"hwdescription":<String_value>,

"sysid":<Double_value>,

"manufactureday":<Integer_value>,

"manufacturemonth":<Integer_value>,
```

```
"manufactureyear":<Integer_value>,

"cpufrequncy":<Integer_value>,

"hostid":<Integer_value>,

"host":<String_value>,

"serialno":<String_value>,

"encodedserialno":<String_value>,

"netscaleruuid":<String_value>,

"bmcrevision":<String_value>

}]}
```

nshmackey

Configuration for HMAC key resource.

Properties

(click to see Operations)

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
name	<String>	Read-write	Key name. This follows the same syntax rules as other expression entity names: It must begin with an alpha character (A-Z or a-z) or an underscore (_). The rest of the characters must be

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
			alpha, numeric (0-9) or underscores. It cannot be re or xp (reserved for regular and XPath expressions). It cannot be an expression reserved word (e.g. SYS or HTTP). It cannot be used for an existing expression object (HTTP callout, patset, dataset, stringmap, or named expression). Minimum length = 1
digest	<String>	Read-write	Digest (hash) function to be used in the HMAC computation. Possible values = MD2, MD4, MD5, SHA1, SHA224, SHA256, SHA384, SHA512
keyvalue	<String>	Read-write	The hex-encoded key to be used in the HMAC computation. The key can be any length (up to a Citrix ADC-imposed

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
			maximum of 255 bytes). If the length is less than the digest block size, it will be zero padded up to the block size. If it is greater than the block size, it will be hashed using the digest function to the block size. The block size for each digest is: MD2 - 16 bytes MD4 - 16 bytes MD5 - 16 bytes SHA1 - 20 bytes SHA224 - 28 bytes SHA256 - 32 bytes SHA384 - 48 bytes SHA512 - 64 bytes Note that the key will be encrypted when it it is saved.
comment	<String>	Read-write	Comments associated with this encryption key.
__count	<Double>	Read-only	count parameter

Operations

(click to see Properties)

ADD| UPDATE| UNSET| DELETE| GET (ALL)| GET| COUNT

Some options that you can use for each operations:

- **Getting warnings in response:**NITRO allows you to get warnings in an operation by specifying the "warning" query parameter as "yes". For example, to get warnings while connecting to the Citrix ADC appliance, the URL is as follows:

  http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/login?warning=yes

  If any, the warnings are displayed in the response payload with the HTTP code "209 X-NITRO-WARNING".

- **Authenticated access for individual NITRO operations:**NITRO allows you to logon to the Citrix ADC appliance to perform individual operations. You can use this option instead of creating a NITRO session (using the login object) and then using that session to perform all operations,

  To do this, you must specify the username and password in the request header of the NITRO request as follows:

  X-NITRO-USER:<username>

  X-NITRO-PASS:<password>

  Note: In such cases, make sure that the request header DOES not include the following:

  Cookie:NITRO_AUTH_TOKEN=<tokenvalue>

`*Note:`*

Mandatory parameters are marked in redand placeholder content is marked in <green>.

add

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/nshmackey

HTTP Method:POST

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Content-Type:application/json

Request Payload:

```
{"nshmackey":{

<b>"name":<String_value>,

</b><b>"digest":<String_value>,

</b>"keyvalue":<String_value>,

"comment":<String_value>

}}
```

Response:

HTTP Status Code on Success: 201 Created HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error

update

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/nshmackey

HTTP Method:PUT

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Content-Type:application/json

Request Payload:

```
{"nshmackey":{

<b>"name":<String_value>,

</b>"digest":<String_value>,

"keyvalue":<String_value>,
```

```
"comment":<String_value>

}}
```

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error

unset

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/nshmackey?**action=unset**

HTTP Method:POST

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Content-Type:application/json

Request Payload:

```
{"nshmackey":{

<b>"name":<String_value>,

</b>"comment":true

}}
```

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error

delete

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/nshmackey/name_value<String>

HTTP Method:DELETE

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue>

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error

get (all)

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/nshmackey

Query-parameters:

attrs

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/nshmackey?**attrs=property-name1,property-name2**

Use this query parameter to specify the resource details that you want to retrieve.

filter

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/nshmackey?**filter=property-name1:property-val1,property-name2:property-val2**

Use this query-parameter to get the filtered set of nshmackey resources configured on Citrix ADC. Filtering can be done on any of the properties of the resource.

view

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/nshmackey?**view=summary**

Note: By default, the retrieved results are displayed in detail view (?view=detail).

pagination

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/nshmackey?**pagesize=#no;pageno=#no**

Use this query-parameter to get the nshmackey resources in chunks.

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "nshmackey": [ {

"name":<String_value>,

"digest":<String_value>,

"comment":<String_value>

}]}
```

get

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/nshmackey/name_value<String>

Query-parameters:

attrs

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/nshmackey/name_value<String>?**attrs=property-name1,property-name2**

Use this query parameter to specify the resource details that you want to retrieve.

view

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/nshmackey/name_value<String>?**view=summary**

Note: By default, the retrieved results are displayed in detail view (?view=detail).

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "nshmackey": [ {

"name":<String_value>,

"digest":<String_value>,

"comment":<String_value>

}]}
```

count

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/nshmackey?**count=yes**

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "nshmackey": [ { "__count": "#no"} ] }
```

nshostname

Configuration for host name resource.

Properties

(click to see Operations)

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
hostname	<String>	Read-write	Host name for the Citrix ADC. Minimum length = 1 Maximum length = 255
ownernode	<Double>	Read-write	ID of the cluster node for which you are setting the hostname. Can be configured only through the cluster IP address. Default value: 255 Minimum value = 0

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
			Maximum value = 31
__count	<Double>	Read-only	count parameter

Operations

(click to see Properties)

UPDATE| GET (ALL)| COUNT

Some options that you can use for each operations:

- **Getting warnings in response:**NITRO allows you to get warnings in an operation by specifying the "warning" query parameter as "yes". For example, to get warnings while connecting to the Citrix ADC appliance, the URL is as follows:

  http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/login?warning=yes

  If any, the warnings are displayed in the response payload with the HTTP code "209 X-NITRO-WARNING".

- **Authenticated access for individual NITRO operations:**NITRO allows you to logon to the Citrix ADC appliance to perform individual operations. You can use this option instead of creating a NITRO session (using the login object) and then using that session to perform all operations,

  To do this, you must specify the username and password in the request header of the NITRO request as follows:

  X-NITRO-USER:<username>

  X-NITRO-PASS:<password>

  Note: In such cases, make sure that the request header DOES not include the following:

  Cookie:NITRO_AUTH_TOKEN=<tokenvalue>

`*Note:`*

Mandatory parameters are marked in redand placeholder content is marked in <green>.

update

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/nshostname

HTTP Method:PUT

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Content-Type:application/json

Request Payload:

```
{"nshostname":{

<b>"hostname":<String_value>,

</b>"ownernode":<Double_value>

}}
```

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error

get (all)

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/nshostname

Query-parameters:

attrs

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/nshostname?**attrs=property-name1,property-name2**

Use this query parameter to specify the resource details that you want to retrieve.

filter

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/nshostname?**filter=property-name1:property-val1,property-name2:property-val2**

Use this query-parameter to get the filtered set of nshostname resources configured on Citrix ADC. Filtering can be done on any of the properties of the resource.

view

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/nshostname?**view=summary**

Note: By default, the retrieved results are displayed in detail view (?view=detail).

pagination

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/nshostname?**pagesize=#no;pageno=#no**

Use this query-parameter to get the nshostname resources in chunks.

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "nshostname": [ {

"hostname":<String_value>,

"ownernode":<Double_value>

}]}
```

count

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/nshostname?**count=yes**

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "nshostname": [ { "__count": "#no"} ] }
```

nshttpparam

Configuration for HTTP parameter resource.

Properties

(click to see Operations)

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
dropinvalreqs	<String>	Read-write	Drop invalid HTTP requests or responses. Default value: OFF Possible values = ON, OFF

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
markhttp09inval	<String>	Read-write	Mark HTTP/0.9 requests as invalid. Default value: OFF Possible values = ON, OFF
markconnreqinval	<String>	Read-write	Mark CONNECT requests as invalid. Default value: OFF Possible values = ON, OFF
insnssrvrhdr	<String>	Read-write	Enable or disable Citrix ADC server header insertion for Citrix ADC generated HTTP responses. Default value: OFF Possible values = ON, OFF
nssrvrhdr	<String>	Read-write	The server header value to be inserted. If no explicit header is specified then NSBUILD.RELEASE is used as default server header. Minimum length = 1
logerrresp	<String>	Read-write	Server header value to be inserted. Default value: ON Possible values = ON, OFF

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
conmultiplex	<String>	Read-write	Reuse server connections for requests from more than one client connections. Default value: ENABLED Possible values = ENABLED, DISABLED
maxreusepool	<Double>	Read-write	Maximum limit on the number of connections, from the Citrix ADC to a particular server that are kept in the reuse pool. This setting is helpful for optimal memory utilization and for reducing the idle connections to the server just after the peak time. Default value: 0 Minimum value = 0 Maximum value = 360000
http2serverside	<String>	Read-write	Enable/Disable HTTP/2 on server side. Default value: OFF Possible values = ON, OFF

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
ignoreconnectcoding scheme	<String>	Read-write	Ignore Coding scheme in CONNECT request. Default value: DISABLED Possible values = ENABLED, DISABLED

Operations

(click to see Properties)

UPDATE| UNSET| GET (ALL)

Some options that you can use for each operations:

- **Getting warnings in response:**NITRO allows you to get warnings in an operation by specifying the "warning" query parameter as "yes". For example, to get warnings while connecting to the Citrix ADC appliance, the URL is as follows:

  http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/login?warning=yes

  If any, the warnings are displayed in the response payload with the HTTP code "209 X-NITRO-WARNING".

- **Authenticated access for individual NITRO operations:**NITRO allows you to logon to the Citrix ADC appliance to perform individual operations. You can use this option instead of creating a NITRO session (using the login object) and then using that session to perform all operations,

  To do this, you must specify the username and password in the request header of the NITRO request as follows:

  X-NITRO-USER:<username>

  X-NITRO-PASS:<password>

  Note: In such cases, make sure that the request header DOES not include the following:

  Cookie:NITRO_AUTH_TOKEN=<tokenvalue>

`*Note:`*

Mandatory parameters are marked in redand placeholder content is marked in <green>.

update

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/nshttpparam

HTTP Method:PUT

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Content-Type:application/json

Request Payload:

```
{"nshttpparam":{

"dropinvalreqs":<String_value>,

"markhttp09inval":<String_value>,

"markconnreqinval":<String_value>,

"insnssrvrhdr":<String_value>,

"nssrvrhdr":<String_value>,

"logerrresp":<String_value>,

"conmultiplex":<String_value>,

"maxreusepool":<Double_value>,

"http2serverside":<String_value>,

"ignoreconnectcodingscheme":<String_value>

}}
```

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error

unset

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/nshttpparam?**action=unset**

HTTP Method:POST

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Content-Type:application/json

Request Payload:

```
{"nshttpparam":{

"dropinvalreqs":true,

"markhttp09inval":true,

"markconnreqinval":true,

"insnssrvrhdr":true,

"nssrvrhdr":true,

"logerrresp":true,

"conmultiplex":true,

"maxreusepool":true,

"http2serverside":true,

"ignoreconnectcodingscheme":true

}}
```

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error

get (all)

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/nshttpparam

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "nshttpparam": [ {

"dropinvalreqs":<String_value>,

"markhttp09inval":<String_value>,

"markconnreqinval":<String_value>,

"insnssrvrhdr":<String_value>,

"nssrvrhdr":<String_value>,

"logerrresp":<String_value>,

"conmultiplex":<String_value>,

"maxreusepool":<Double_value>,
```

```
"http2serverside":<String_value>,

"ignoreconnectcodingscheme":<String_value>

}]}
```

nshttpprofile

Configuration for HTTP profile resource.

Properties

(click to see Operations)

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
name	<String>	Read-write	Name for an HTTP profile. Must begin with a letter, number, or the underscore \(_\) character. Other characters allowed, after the first character, are the hyphen \(-\), period \(.\), hash \(#\), space \(\), at \(@\), colon \(:\), and equal \(=\) characters. The name of a HTTP profile cannot be changed after it is created. CLI Users: If the name includes one or more spaces, enclose the name in double or single quotation marks \(for example, "my

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
			http profile" or 'my http profile'\). Minimum length = 1 Maximum length = 127
dropinvalreqs	<String>	Read-write	Drop invalid HTTP requests or responses. Default value: DISABLED Possible values = ENABLED, DISABLED
markhttp09inval	<String>	Read-write	Mark HTTP/0.9 requests as invalid. Default value: DISABLED Possible values = ENABLED, DISABLED
markconnreqinval	<String>	Read-write	Mark CONNECT requests as invalid. Default value: DISABLED Possible values = ENABLED, DISABLED
marktracereqinval	<String>	Read-write	Mark TRACE requests as invalid.

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
			Default value: DISABLED Possible values = ENABLED, DISABLED
cmponpush	<String>	Read-write	Start data compression on receiving a TCP packet with PUSH flag set. Default value: DISABLED Possible values = ENABLED, DISABLED
conmultiplex	<String>	Read-write	Reuse server connections for requests from more than one client connections. Default value: ENABLED Possible values = ENABLED, DISABLED
maxreusepool	<Double>	Read-write	Maximum limit on the number of connections, from the Citrix ADC to a particular server that are kept in the reuse pool. This setting is helpful for optimal memory utilization

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
			and for reducing the idle connections to the server just after the peak time. Zero implies no limit on reuse pool size. If non-zero value is given, it has to be greater than or equal to the number of running Packet Engines. Default value: 0 Minimum value = 0 Maximum value = 360000
dropextracrlf	<String>	Read-write	Drop any extra 'CR' and 'LF' characters present after the header. Default value: ENABLED Possible values = ENABLED, DISABLED
incomphdrdelay	<Double>	Read-write	Maximum time to wait, in milliseconds, between incomplete header packets. If the header packets take longer to arrive at Citrix ADC, the connection is silently dropped. Default value: 7000

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
			Minimum value = 1 Maximum value = 360000
websocket	<String>	Read-write	HTTP connection to be upgraded to a web socket connection. Once upgraded, Citrix ADC does not process Layer 7 traffic on this connection. Default value: DISABLED Possible values = ENABLED, DISABLED
rtsptunnel	<String>	Read-write	Allow RTSP tunnel in HTTP. Once application/x-rtsp-tunnelled is seen in Accept or Content-Type header, Citrix ADC does not process Layer 7 traffic on this connection. Default value: DISABLED Possible values = ENABLED, DISABLED
reqtimeout	<Double>	Read-write	Time, in seconds, within which the HTTP request must

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
			complete. If the request does not complete within this time, the specified request timeout action is executed. Zero disables the timeout. Default value: 0 Minimum value = 0 Maximum value = 86400
adpttimeout	<String>	Read-write	Adapts the configured request timeout based on flow conditions. The timeout is increased or decreased internally and applied on the flow. Default value: DISABLED Possible values = ENABLED, DISABLED
reqtimeoutaction	<String>	Read-write	Action to take when the HTTP request does not complete within the specified request timeout duration. You can configure the following actions: * RESET - Send RST (reset) to client

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
			when timeout occurs. * DROP - Drop silently when timeout occurs. * Custom responder action - Name of the responder action to trigger when timeout occurs, used to send custom message.
dropextradata	<String>	Read-write	Drop any extra data when server sends more data than the specified content-length. Default value: DISABLED Possible values = ENABLED, DISABLED
weblog	<String>	Read-write	Enable or disable web logging. Default value: ENABLED Possible values = ENABLED, DISABLED
clientiphdrexpr	<String>	Read-write	Name of the header that contains the real client IP address.

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
maxreq	<Double>	Read-write	Maximum number of requests allowed on a single connection. Zero implies no limit on the number of requests. Default value: 0 Minimum value = 0 Maximum value = 65534
persistentetag	<String>	Read-write	Generate the persistent Citrix ADC specific ETag for the HTTP response with ETag header. Default value: DISABLED Possible values = ENABLED, DISABLED
spdy	<String>	Read-write	Enable SPDYv2 or SPDYv3 or both over SSL vserver. SSL will advertise SPDY support either during NPN Handshake or when client will advertises SPDY support during ALPN handshake. Both SPDY versions are enabled when this parameter is set to ENABLED.

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
			Default value: DISABLED Possible values = DISABLED, ENABLED, V2, V3
http2	<String>	Read-write	Choose whether to enable support for HTTP/2. Default value: DISABLED Possible values = ENABLED, DISABLED
http2direct	<String>	Read-write	Choose whether to enable support for Direct HTTP/2. Default value: DISABLED Possible values = ENABLED, DISABLED
altsvc	<String>	Read-write	Choose whether to enable support for Alternative Service. Default value: DISABLED Possible values = ENABLED, DISABLED
reusepooltimeout	<Double>	Read-write	Idle timeout (in seconds) for server connections in re-

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
			use pool. Connections in the re-use pool are flushed, if they remain idle for the configured timeout. Default value: 0 Minimum value = 0 Maximum value = 31536000
maxheaderlen	<Double>	Read-write	Number of bytes to be queued to look for complete header before returning error. If complete header is not obtained after queuing these many bytes, request will be marked as invalid and no L7 processing will be done for that TCP connection. Default value: 24820 Minimum value = 2048 Maximum value = 61440
minreusepool	<Double>	Read-write	Minimum limit on the number of connections, from the Citrix ADC to a particular server that

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
			are kept in the reuse pool. This setting is helpful for optimal memory utilization and for reducing the idle connections to the server just after the peak time. Zero implies no limit on reuse pool size. Default value: 0 Minimum value = 0 Maximum value = 360000
http2maxheaderlistsize	<Double>	Read-write	Maximum size of header list that the Citrix ADC is prepared to accept, in bytes. NOTE: The actual plain text header size that the Citrix ADC accepts is limited by maxHeaderLen. Please change maxHeaderLen parameter as well when modifying http2MaxHeaderListSize. Default value: 24576 Minimum value = 8192 Maximum value = 65535

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
http2maxframesize	<Double>	Read-write	Maximum size of the frame payload that the Citrix ADC is willing to receive, in bytes. Default value: 16384 Minimum value = 16384 Maximum value = 32768
http2maxconcurrentstreams	<Double>	Read-write	Maximum number of concurrent streams that is allowed per connection. Default value: 100 Minimum value = 0 Maximum value = 1000
http2initialwindowsize	<Double>	Read-write	Initial window size for stream level flow control, in bytes. Default value: 65535 Minimum value = 8192 Maximum value = 20971520
http2headertablesize	<Double>	Read-write	Maximum size of the header compression table used to decode

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
			header blocks, in bytes. Default value: 4096 Minimum value = 0 Maximum value = 16384
http2minseverconn	<Double>	Read-write	Minimum number of HTTP2 connections established to backend server, on receiving HTTP requests from client before multiplexing the streams into the available HTTP/2 connections. Default value: 20 Minimum value = 1 Maximum value = 360000
apdexcltresptimethreshold	<Double>	Read-write	This option sets the satisfactory threshold (T) for client response time in milliseconds to be used for APDEX calculations. This means a transaction responding in less than this threshold is considered satisfactory. Transaction responding between T and 4*T is

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
			considered tolerable. Any transaction responding in more than 4*T time is considered frustrating. Citrix ADC maintains stats for such tolerable and frustrating transcations. And client response time related apdex counters are only updated on a vserver which receives clients traffic. Default value: 500 Minimum value = 1 Maximum value = 3600000
refcnt	<Double>	Read-only	Number of entities using this profile.
apdexsvrresptimethreshold	<Double>	Read-only	This option sets the satisfactory threshold (T) for server response time in milliseconds to be used for APDEX calculations. This means a transaction responding in less than this threshold is considered satisfactory. Transaction responding between T and 4*T is

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
			considered tolerable. Any transaction responding in more than 4*T time is considered frustrating. Citrix ADC maintains stats for such tolerable and frustrating transcations. Server Response time related apdex counters are only updated on backend services or a backend vserver which is not accepting client traffic. Default value: 400 Minimum value = 1 Maximum value = 3600000
__count	<Double>	Read-only	count parameter

Operations

(click to see Properties)

ADD| DELETE| UPDATE| UNSET| GET (ALL)| GET| COUNT

Some options that you can use for each operations:

- **Getting warnings in response:**NITRO allows you to get warnings in an operation by specifying the "warning" query parameter as "yes". For example, to get warnings while connecting to the Citrix ADC appliance, the URL is as follows:

  http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/login?warning=yes

If any, the warnings are displayed in the response payload with the HTTP code "209 X-NITRO-WARNING".

- **Authenticated access for individual NITRO operations:**NITRO allows you to logon to the Citrix ADC appliance to perform individual operations. You can use this option instead of creating a NITRO session (using the login object) and then using that session to perform all operations,

  To do this, you must specify the username and password in the request header of the NITRO request as follows:

  X-NITRO-USER:<username>

  X-NITRO-PASS:<password>

  Note: In such cases, make sure that the request header DOES not include the following:

  Cookie:NITRO_AUTH_TOKEN=<tokenvalue>

`*Note:`*

Mandatory parameters are marked in redand placeholder content is marked in <green>.

add

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/nshttpprofile

HTTP Method:POST

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Content-Type:application/json

Request Payload:

```
{"nshttpprofile":{

<b>"name":<String_value>,

</b>"dropinvalreqs":<String_value>,

"markhttp09inval":<String_value>,
```

```
"markconnreqinval":<String_value>,

"marktracereqinval":<String_value>,

"cmponpush":<String_value>,

"conmultiplex":<String_value>,

"maxreusepool":<Double_value>,

"dropextracrlf":<String_value>,

"incomphdrdelay":<Double_value>,

"websocket":<String_value>,

"rtsptunnel":<String_value>,

"reqtimeout":<Double_value>,

"adpttimeout":<String_value>,

"reqtimeoutaction":<String_value>,

"dropextradata":<String_value>,

"weblog":<String_value>,

"clientiphdrexpr":<String_value>,

"maxreq":<Double_value>,

"persistentetag":<String_value>,

"spdy":<String_value>,

"http2":<String_value>,

"http2direct":<String_value>,

"altsvc":<String_value>,

"reusepooltimeout":<Double_value>,
```

```
"maxheaderlen":<Double_value>,

"minreusepool":<Double_value>,

"http2maxheaderlistsize":<Double_value>,

"http2maxframesize":<Double_value>,

"http2maxconcurrentstreams":<Double_value>,

"http2initialwindowsize":<Double_value>,

"http2headertablesize":<Double_value>,

"http2minseverconn":<Double_value>,

"apdexcltresptimethreshold":<Double_value>

}}
```

Response:

HTTP Status Code on Success: 201 Created HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error

delete

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/nshttpprofile/name_value<String>

HTTP Method:DELETE

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue>

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error

update

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/nshttpprofile

HTTP Method:PUT

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Content-Type:application/json

Request Payload:

```
{"nshttpprofile":{

<b>"name":<String_value>,

</b>"dropinvalreqs":<String_value>,

"markhttp09inval":<String_value>,

"markconnreqinval":<String_value>,

"marktracereqinval":<String_value>,

"cmponpush":<String_value>,

"conmultiplex":<String_value>,

"maxreusepool":<Double_value>,

"dropextracrlf":<String_value>,

"incomphdrdelay":<Double_value>,

"websocket":<String_value>,

"rtsptunnel":<String_value>,

"reqtimeout":<Double_value>,

"adpttimeout":<String_value>,
```

```
"reqtimeoutaction":<String_value>,

"dropextradata":<String_value>,

"weblog":<String_value>,

"clientiphdrexpr":<String_value>,

"maxreq":<Double_value>,

"persistentetag":<String_value>,

"spdy":<String_value>,

"http2":<String_value>,

"http2direct":<String_value>,

"altsvc":<String_value>,

"http2maxheaderlistsize":<Double_value>,

"http2maxframesize":<Double_value>,

"http2maxconcurrentstreams":<Double_value>,

"http2initialwindowsize":<Double_value>,

"http2headertablesize":<Double_value>,

"http2minseverconn":<Double_value>,

"reusepooltimeout":<Double_value>,

"maxheaderlen":<Double_value>,

"minreusepool":<Double_value>,

"apdexcltresptimethreshold":<Double_value>

}}
```

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error

unset

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/nshttpprofile?**action=unset**

HTTP Method:POST

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Content-Type:application/json

Request Payload:

```
{"nshttpprofile":{

<b>"name":<String_value>,

</b>"dropinvalreqs":true,

"markhttp09inval":true,

"markconnreqinval":true,

"marktracereqinval":true,

"cmponpush":true,

"conmultiplex":true,

"maxreusepool":true,

"dropextracrlf":true,

"incomphdrdelay":true,
```

```
"websocket":true,

"dropextradata":true,

"clientiphdrexpr":true,

"reqtimeout":true,

"adpttimeout":true,

"reqtimeoutaction":true,

"weblog":true,

"maxreq":true,

"persistentetag":true,

"spdy":true,

"http2":true,

"http2maxheaderlistsize":true,

"http2maxframesize":true,

"http2maxconcurrentstreams":true,

"http2initialwindowsize":true,

"http2headertablesize":true,

"http2minseverconn":true,

"reusepooltimeout":true,

"maxheaderlen":true,

"rtsptunnel":true,

"minreusepool":true,

"apdexcltresptimethreshold":true,
```

```
"http2direct":true,

"altsvc":true

}}
```

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error

get (all)

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/nshttpprofile

Query-parameters:

attrs

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/nshttpprofile?**attrs=property-name1,property-name2**

Use this query parameter to specify the resource details that you want to retrieve.

filter

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/nshttpprofile?**filter=property-name1:property-val1,property-name2:property-val2**

Use this query-parameter to get the filtered set of nshttpprofile resources configured on Citrix ADC. Filtering can be done on any of the properties of the resource.

view

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/nshttpprofile?**view=summary**

Note: By default, the retrieved results are displayed in detail view (?view=detail).

pagination

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/nshttpprofile?**pagesize=#no;pageno=#no**

Use this query-parameter to get the nshttpprofile resources in chunks.

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "nshttpprofile": [ {

"name":<String_value>,

"dropinvalreqs":<String_value>,

"markhttp09inval":<String_value>,

"markconnreqinval":<String_value>,

"marktracereqinval":<String_value>,

"cmponpush":<String_value>,

"conmultiplex":<String_value>,

"maxreusepool":<Double_value>,

"websocket":<String_value>,

"refcnt":<Double_value>,

"dropextracrlf":<String_value>,
```

```
"incomphdrdelay":<Double_value>,

"reqtimeout":<Double_value>,

"adpttimeout":<String_value>,

"reqtimeoutaction":<String_value>,

"dropextradata":<String_value>,

"weblog":<String_value>,

"clientiphdrexpr":<String_value>,

"maxreq":<Double_value>,

"persistentetag":<String_value>,

"spdy":<String_value>,

"http2":<String_value>,

"http2direct":<String_value>,

"altsvc":<String_value>,

"http2maxheaderlistsize":<Double_value>,

"http2maxframesize":<Double_value>,

"http2maxconcurrentstreams":<Double_value>,

"http2initialwindowsize":<Double_value>,

"http2headertablesize":<Double_value>,

"http2minseverconn":<Double_value>,

"reusepooltimeout":<Double_value>,

"maxheaderlen":<Double_value>,

"rtsptunnel":<String_value>,
```

```
"minreusepool":<Double_value>,

"apdexsvrresptimethreshold":<Double_value>,

"apdexcltresptimethreshold":<Double_value>

}]}
```

get

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/nshttpprofile/name_value<String>

Query-parameters:

attrs

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/nshttpprofile/name_value<String>?**attrs=property-name1,property-name2**

Use this query parameter to specify the resource details that you want to retrieve.

view

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/nshttpprofile/name_value<String>?**view=summary**

Note: By default, the retrieved results are displayed in detail view (?view=detail).

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "nshttpprofile": [ {

"name":<String_value>,

"dropinvalreqs":<String_value>,

"markhttp09inval":<String_value>,

"markconnreqinval":<String_value>,

"marktracereqinval":<String_value>,

"cmponpush":<String_value>,

"conmultiplex":<String_value>,

"maxreusepool":<Double_value>,

"websocket":<String_value>,

"refcnt":<Double_value>,

"dropextracrlf":<String_value>,

"incomphdrdelay":<Double_value>,

"reqtimeout":<Double_value>,

"adpttimeout":<String_value>,

"reqtimeoutaction":<String_value>,

"dropextradata":<String_value>,

"weblog":<String_value>,

"clientiphdrexpr":<String_value>,

"maxreq":<Double_value>,

"persistentetag":<String_value>,
```

```
"spdy":<String_value>,

"http2":<String_value>,

"http2direct":<String_value>,

"altsvc":<String_value>,

"http2maxheaderlistsize":<Double_value>,

"http2maxframesize":<Double_value>,

"http2maxconcurrentstreams":<Double_value>,

"http2initialwindowsize":<Double_value>,

"http2headertablesize":<Double_value>,

"http2minseverconn":<Double_value>,

"reusepooltimeout":<Double_value>,

"maxheaderlen":<Double_value>,

"rtsptunnel":<String_value>,

"minreusepool":<Double_value>,

"apdexsvrresptimethreshold":<Double_value>,

"apdexcltresptimethreshold":<Double_value>

}]}
```

count

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/nshttpprofile?**count=yes**

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "nshttpprofile": [ { "__count": "#no"} ] }
```

nsicapprofile

Configuration for ICAP profile resource.

Properties

(click to see Operations)

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
name	<String>	Read-write	Name for an ICAP profile. Must begin with a letter, number, or the underscore \(_\) character. Other characters allowed, after the first character, are the hyphen \(-\), period \(.\), hash \(#\), space \(\), at \(@\), colon \(:\), and equal \(=\) characters. The name of a ICAP profile cannot be

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
			changed after it is created. CLI Users: If the name includes one or more spaces, enclose the name in double or single quotation marks \ (for example, "my icap profile" or 'my icap profile'\). Minimum length = 1 Maximum length = 127
preview	<String>	Read-write	Enable or Disable preview header with ICAP request. This feature allows an ICAP server to see the beginning of a transaction, then decide if it wants to opt-out of the transaction early instead of receiving the remainder of the request message. Default value: DISABLED Possible values = ENABLED, DISABLED
previewlength	<Double>	Read-write	Value of Preview Header field. Citrix ADC uses the

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
			minimum of this set value and the preview size received on OPTIONS response. Default value: 4096 Minimum value = 0 Maximum value = 4294967294
uri	<String>	Read-write	URI representing icap service. It is a mandatory argument while creating an icapprofile. Minimum length = 1
hostheader	<String>	Read-write	ICAP Host Header. Minimum length = 1
useragent	<String>	Read-write	ICAP User Agent Header String. Minimum length = 1
mode	<String>	Read-write	ICAP Mode of operation. It is a mandatory argument while creating an icapprofile. Possible values = REQMOD, RESPMOD
queryparams	<String>	Read-write	Query parameters to be included with ICAP request URI. Entered values should be in arg=value format.

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
			For more than one parameters, add ; separated values. e.g.: arg1=val1;arg2=val2. Minimum length = 1
connectionkeepalive	<String>	Read-write	If enabled, Citrix ADC keeps the ICAP connection alive after a transaction to reuse it to send next ICAP request. Default value: ENABLED Possible values = ENABLED, DISABLED
allow204	<String>	Read-write	Enable or Disable sending Allow: 204 header in ICAP request. Default value: ENABLED Possible values = ENABLED, DISABLED
inserticapheaders	<String>	Read-write	Insert custom ICAP headers in the ICAP request to send to ICAP server. The headers can be static or can be dynamically constructed using PI Policy Expression.

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
			For example, to send static user agent and Client's IP address, the expression can be specified as "User-Agent: NS-ICAP-Client/ V1.0\r\nX-Client-IP: "+CLIENT.IP.SRC+"\r \n". The Citrix ADC does not check the validity of the specified header name-value. You must manually validate the specified header syntax. Minimum length = 1
inserthttprequest	<String>	Read-write	Exact HTTP request, in the form of an expression, which the Citrix ADC encapsulates and sends to the ICAP server. If you set this parameter, the ICAP request is sent using only this header. This can be used when the HTTP header is not available to send or ICAP server only needs part of the incoming HTTP request. The request expression is

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
			constrained by the feature for which it is used. The Citrix ADC does not check the validity of this request. You must manually validate the request. Minimum length = 1
reqtimeout	<Double>	Read-write	Time, in seconds, within which the remote service request must complete. If the request does not complete within this time, the specified request timeout action is executed. Zero disables the timeout. Default value: 0 Minimum value = 0 Maximum value = 86400
reqtimeoutaction	<String>	Read-write	Name of the action to perform if the Vserver/Server representing the remote service does not respond. There are also some built-in actions which can be used. These are:

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
			* BYPASS - ignore this remote service action and send the request as is.This is done by default. * RESET - Reset the client connection by closing it. The client program, such as a browser, will handle this and may inform the user. The client may then resend the request if desired. * DROP - Drop the request without sending a response to the user. Default value: BYPASS Possible values = BYPASS, DROP, RESET
logaction	<String>	Read-write	Name of the audit message action which would be evaluated on receiving the ICAP response to emit the logs.
__count	<Double>	Read-only	count parameter

Operations

(click to see Properties)

ADD| DELETE| UPDATE| UNSET| GET (ALL)| GET| COUNT

Some options that you can use for each operations:

- **Getting warnings in response:**NITRO allows you to get warnings in an operation by specifying the "warning" query parameter as "yes". For example, to get warnings while connecting to the Citrix ADC appliance, the URL is as follows:

  http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/login?warning=yes

  If any, the warnings are displayed in the response payload with the HTTP code "209 X-NITRO-WARNING".

- **Authenticated access for individual NITRO operations:**NITRO allows you to logon to the Citrix ADC appliance to perform individual operations. You can use this option instead of creating a NITRO session (using the login object) and then using that session to perform all operations,

  To do this, you must specify the username and password in the request header of the NITRO request as follows:

  X-NITRO-USER:<username>

  X-NITRO-PASS:<password>

  Note: In such cases, make sure that the request header DOES not include the following:

  Cookie:NITRO_AUTH_TOKEN=<tokenvalue>

`*Note:`*

Mandatory parameters are marked in redand placeholder content is marked in <green>.

add

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/nsicapprofile

HTTP Method:POST

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Content-Type:application/json

Request Payload:

```
{"nsicapprofile":{
<b>"name":<String_value>,
</b>"preview":<String_value>,
"previewlength":<Double_value>,
<b>"uri":<String_value>,
</b>"hostheader":<String_value>,
"useragent":<String_value>,
<b>"mode":<String_value>,
</b>"queryparams":<String_value>,
"connectionkeepalive":<String_value>,
"allow204":<String_value>,
"inserticapheaders":<String_value>,
"inserthttprequest":<String_value>,
"reqtimeout":<Double_value>,
"reqtimeoutaction":<String_value>,
"logaction":<String_value>
}}
```

Response:

HTTP Status Code on Success: 201 Created HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error

delete

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/nsicapprofile/name_value<String>

HTTP Method:DELETE

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue>

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error

update

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/nsicapprofile

HTTP Method:PUT

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Content-Type:application/json

Request Payload:

```
{"nsicapprofile":{

<b>"name":<String_value>,

</b>"preview":<String_value>,

"previewlength":<Double_value>,

"uri":<String_value>,

"hostheader":<String_value>,
```

```
"useragent":<String_value>,

"mode":<String_value>,

"queryparams":<String_value>,

"connectionkeepalive":<String_value>,

"allow204":<String_value>,

"inserticapheaders":<String_value>,

"inserthttprequest":<String_value>,

"reqtimeout":<Double_value>,

"reqtimeoutaction":<String_value>,

"logaction":<String_value>

}}
```

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error

unset

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/nsicapprofile?**action=unset**

HTTP Method:POST

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Content-Type:application/json

Request Payload:

```
{"nsicapprofile":{
<b>"name":<String_value>,
</b>"preview":true,
"previewlength":true,
"hostheader":true,
"useragent":true,
"queryparams":true,
"connectionkeepalive":true,
"allow204":true,
"inserticapheaders":true,
"inserthttprequest":true,
"reqtimeout":true,
"reqtimeoutaction":true,
"logaction":true
}}
```

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error

get (all)

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/nsicapprofile

Query-parameters:

attrs

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/nsicapprofile?**attrs=property-name1,property-name2**

Use this query parameter to specify the resource details that you want to retrieve.

filter

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/nsicapprofile?**filter=property-name1:property-val1,property-name2:property-val2**

Use this query-parameter to get the filtered set of nsicapprofile resources configured on Citrix ADC. Filtering can be done on any of the properties of the resource.

view

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/nsicapprofile?**view=summary**

Note: By default, the retrieved results are displayed in detail view (?view=detail).

pagination

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/nsicapprofile?**pagesize=#no;pageno=#no**

Use this query-parameter to get the nsicapprofile resources in chunks.

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "nsicapprofile": [ {
"name":<String_value>,
"preview":<String_value>,
"previewlength":<Double_value>,
"uri":<String_value>,
"hostheader":<String_value>,
"useragent":<String_value>,
"mode":<String_value>,
"queryparams":<String_value>,
"connectionkeepalive":<String_value>,
"allow204":<String_value>,
"inserticapheaders":<String_value>,
"inserthttprequest":<String_value>,
"reqtimeout":<Double_value>,
"reqtimeoutaction":<String_value>,
"logaction":<String_value>
}]}
```

get

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/nsicapprofile/name_value<String>

Query-parameters:

attrs

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/nsicapprofile/name_value<String>?**attrs=property-name1,property-name2**

Use this query parameter to specify the resource details that you want to retrieve.

view

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/nsicapprofile/name_value<String>?**view=summary**

Note: By default, the retrieved results are displayed in detail view (?view=detail).

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "nsicapprofile": [ {

"name":<String_value>,

"preview":<String_value>,

"previewlength":<Double_value>,

"uri":<String_value>,

"hostheader":<String_value>,
```

```
"useragent":<String_value>,

"mode":<String_value>,

"queryparams":<String_value>,

"connectionkeepalive":<String_value>,

"allow204":<String_value>,

"inserticapheaders":<String_value>,

"inserthttprequest":<String_value>,

"reqtimeout":<Double_value>,

"reqtimeoutaction":<String_value>,

"logaction":<String_value>

}]}
```

count

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/nsicapprofile?**count=yes**

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "nsicapprofile": [ { "__count": "#no"} ] }
```

nsip

Configuration for ip resource.

Properties

(click to see Operations)

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
ipaddress	<String>	Read-write	IPv4 address to create on the Citrix ADC. Cannot be changed after the IP address is created. Minimum length = 1
netmask	<String>	Read-write	Subnet mask associated with the IP address.
type	<String>	Read-write	Type of the IP address to create on the Citrix ADC. Cannot be changed after the IP address is created. The following are the different types of Citrix ADC owned IP addresses: * A Subnet IP (SNIP) address is used by the Citrix ADC to communicate with the servers. The

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
			Citrix ADC also uses the subnet IP address when generating its own packets, such as packets related to dynamic routing protocols, or to send monitor probes to check the health of the servers. * A Virtual IP (VIP) address is the IP address associated with a virtual server. It is the IP address to which clients connect. An appliance managing a wide range of traffic may have many VIPs configured. Some of the attributes of the VIP address are customized to meet the requirements of the virtual server. * A GSLB site IP (GSLBIP) address is associated with a GSLB site. It is not mandatory to specify a GSLBIP

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
			address when you initially configure the Citrix ADC. A GSLBIP address is used only when you create a GSLB site. * A Cluster IP (CLIP) address is the management address of the cluster. All cluster configurations must be performed by accessing the cluster through this IP address. Default value: SNIP Possible values = SNIP, VIP, NSIP, GSLBsiteIP, CLIP
arp	<String>	Read-write	Respond to ARP requests for this IP address. Default value: ENABLED Possible values = ENABLED, DISABLED
icmp	<String>	Read-write	Respond to ICMP requests for this IP address.

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
			Default value: ENABLED Possible values = ENABLED, DISABLED
vserver	<String>	Read-write	Use this option to set (enable or disable) the virtual server attribute for this IP address. Default value: ENABLED Possible values = ENABLED, DISABLED
telnet	<String>	Read-write	Allow Telnet access to this IP address. Default value: ENABLED Possible values = ENABLED, DISABLED
ftp	<String>	Read-write	Allow File Transfer Protocol (FTP) access to this IP address. Default value: ENABLED Possible values = ENABLED, DISABLED

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
gui	<String>	Read-write	Allow graphical user interface (GUI) access to this IP address. Default value: ENABLED Possible values = ENABLED, SECUREONLY, DISABLED
ssh	<String>	Read-write	Allow secure shell (SSH) access to this IP address. Default value: ENABLED Possible values = ENABLED, DISABLED
snmp	<String>	Read-write	Allow Simple Network Management Protocol (SNMP) access to this IP address. Default value: ENABLED Possible values = ENABLED, DISABLED
mgmtaccess	<String>	Read-write	Allow access to management applications on this IP address.

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
			Default value: DISABLED Possible values = ENABLED, DISABLED
restrictaccess	<String>	Read-write	Block access to nonmanagement applications on this IP. This option is applicable for MIPs, SNIPs, and NSIP, and is disabled by default. Nonmanagement applications can run on the underlying Citrix ADC Free BSD operating system. Default value: DISABLED Possible values = ENABLED, DISABLED
dynamicrouting	<String>	Read-write	Allow dynamic routing on this IP address. Specific to Subnet IP (SNIP) address. Default value: DISABLED Possible values = ENABLED, DISABLED

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
decrementttl	<String>	Read-write	Decrement TTL by 1 when ENABLED.This setting is applicable only for UDP traffic. Default value: DISABLED Possible values = ENABLED, DISABLED
ospf	<String>	Read-write	Use this option to enable or disable OSPF on this IP address for the entity. Default value: DISABLED Possible values = ENABLED, DISABLED
bgp	<String>	Read-write	Use this option to enable or disable BGP on this IP address for the entity. Default value: DISABLED Possible values = ENABLED, DISABLED
rip	<String>	Read-write	Use this option to enable or disable RIP on this IP

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
			address for the entity. Default value: DISABLED Possible values = ENABLED, DISABLED
hostroute	<String>	Read-write	Option to push the VIP to ZebOS routing table for Kernel route redistribution through dynamic routing protocols. Possible values = ENABLED, DISABLED
advertiseondefaultpartition	<String>	Read-write	Advertise VIPs from Shared VLAN on Default Partition. Default value: DISABLED Possible values = ENABLED, DISABLED
networkroute	<String>	Read-write	Option to push the SNIP subnet to ZebOS routing table for Kernel route redistribution through dynamic routing protocol.

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
			Possible values = ENABLED, DISABLED
tag	<Double>	Read-write	Tag value for the network/host route associated with this IP. Default value: 0
hostrtgw	<String>	Read-write	IP address of the gateway of the route for this VIP address. Default value: -1
metric	<Integer>	Read-write	Integer value to add to or subtract from the cost of the route advertised for the VIP address. Minimum value = -16777215
vserverrhilevel	<String>	Read-write	Advertise the route for the Virtual IP (VIP) address on the basis of the state of the virtual servers associated with that VIP. * NONE - Advertise the route for the VIP address, regardless of the state of the virtual servers associated with the address.

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
			* ONE VSERVER - Advertise the route for the VIP address if at least one of the associated virtual servers is in UP state. * ALL VSERVER - Advertise the route for the VIP address if all of the associated virtual servers are in UP state. * VSVR_CNTRLD - Advertise the route for the VIP address according to the RHIstate (RHI STATE) parameter setting on all the associated virtual servers of the VIP address along with their states. When Vserver RHI Level (RHI) parameter is set to VSVR_CNTRLD, the following are different RHI behaviors for the VIP address on the basis of RHIstate

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
			(RHI STATE) settings on the virtual servers associated with the VIP address: * If you set RHI STATE to PASSIVE on all virtual servers, the Citrix ADC always advertises the route for the VIP address. * If you set RHI STATE to ACTIVE on all virtual servers, the Citrix ADC advertises the route for the VIP address if at least one of the associated virtual servers is in UP state. *If you set RHI STATE to ACTIVE on some and PASSIVE on others, the Citrix ADC advertises the route for the VIP address if at least one of the associated virtual servers, whose RHI STATE set to

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
			ACTIVE, is in UP state. Default value: ONE_VSERVER Possible values = ONE_VSERVER, ALL_VSERVERS, NONE, VSVR_CNTRLD
vserverrhimode	<String>	Read-write	Advertise the route for the Virtual IP (VIP) address using dynamic routing protocols or using RISE * DYNMAIC_ROUTING - Advertise the route for the VIP address using dynamic routing protocols (default) * RISE - Advertise the route for the VIP address using RISE. Default value: DYNAMIC_ROUTING Possible values = DYNAMIC_ROUTING, RISE

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
ospflsatype	<String>	Read-write	Type of LSAs to be used by the OSPF protocol, running on the Citrix ADC, for advertising the route for this VIP address. Default value: TYPE5 Possible values = TYPE1, TYPE5
ospfarea	<Double>	Read-write	ID of the area in which the type1 link-state advertisements (LSAs) are to be advertised for this virtual IP (VIP) address by the OSPF protocol running on the Citrix ADC. When this parameter is not set, the VIP is advertised on all areas. Default value: -1 Minimum value = 0 Maximum value = 4294967294LU
state	<String>	Read-write	Enable or disable the IP address. Default value: ENABLED Possible values = ENABLED, DISABLED

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
vrid	<Double>	Read-write	A positive integer that uniquely identifies a VMAC address for binding to this VIP address. This binding is used to set up Citrix ADCs in an active-active configuration using VRRP. Minimum value = 1 Maximum value = 255
icmpresponse	<String>	Read-write	Respond to ICMP requests for a Virtual IP (VIP) address on the basis of the states of the virtual servers associated with that VIP. Available settings function as follows: * NONE - The Citrix ADC responds to any ICMP request for the VIP address, irrespective of the states of the virtual servers associated with the address. * ONE VSERVER - The Citrix ADC responds to any ICMP request for the VIP address if at least one of the

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
			associated virtual servers is in UP state. * ALL VSERVER - The Citrix ADC responds to any ICMP request for the VIP address if all of the associated virtual servers are in UP state. * VSVR_CNTRLD - The behavior depends on the ICMP VSERVER RESPONSE setting on all the associated virtual servers. The following settings can be made for the ICMP VSERVER RESPONSE parameter on a virtual server: * If you set ICMP VSERVER RESPONSE to PASSIVE on all virtual servers,

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
			Citrix ADC always responds. * If you set ICMP VSERVER RESPONSE to ACTIVE on all virtual servers, Citrix ADC responds if even one virtual server is UP. * When you set ICMP VSERVER RESPONSE to ACTIVE on some and PASSIVE on others, Citrix ADC responds if even one virtual server set to ACTIVE is UP. Default value: 5 Possible values = NONE, ONE_VSERVER, ALL_VSERVERS, VSVR_CNTRLD
ownernode	<Double>	Read-write	The owner node in a Cluster for this IP address. Owner node can vary from 0 to 31. If ownernode is not specified then

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
			the IP is treated as Striped IP. Default value: 255
arpresponse	<String>	Read-write	Respond to ARP requests for a Virtual IP (VIP) address on the basis of the states of the virtual servers associated with that VIP. Available settings function as follows: * NONE - The Citrix ADC responds to any ARP request for the VIP address, irrespective of the states of the virtual servers associated with the address. * ONE VSERVER - The Citrix ADC responds to any ARP request for the VIP address if at least one of the associated virtual servers is in UP state. * ALL VSERVER - The Citrix ADC responds to any ARP request for the VIP address if all of the associated

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
			virtual servers are in UP state. Default value: 5 Possible values = NONE, ONE_VSERVER, ALL_VSERVERS
ownerdownresponse	<String>	Read-write	in cluster system, if the owner node is down, whether should it respond to icmp/arp. Default value: YES Possible values = YES, NO
td	<Double>	Read-write	Integer value that uniquely identifies the traffic domain in which you want to configure the entity. If you do not specify an ID, the entity becomes part of the default traffic domain, which has an ID of 0. TD id 4095 is used reserved for LSN use . Minimum value = 0 Maximum value = 4095

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
flags	<Double>	Read-only	The flags for this entry.
hostrtgwact	<String>	Read-only	Actual Gateway used for advertising host route.
ospfareaval	<Double>	Read-only	The area ID of the area in which OSPF Type1 LSAs are advertised.
viprtadv2bsd	<Boolean>	Read-only	Whether this route is advertised to FreeBSD.
vipvsercount	<Double>	Read-only	Number of vservers bound to this VIP.
vipvserdowncount	<Double>	Read-only	Number of vservers bound to this VIP, which are down.
vipvsrvrrhiactivecoun t	<Double>	Read-only	Number of vservers that have RHI state ACTIVE.
vipvsrvrrhiactiveupco unt	<Double>	Read-only	Number of vservers that have RHI state ACTIVE, which are UP.
freeports	<Double>	Read-only	Number of free Ports available on this IP.
riserhimsgcode	<Integer>	Read-only	The code indicating the rise rhi status.
iptype	<String[]>	Read-only	.

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
			Possible values = SNIP, VIP, NSIP, GSLBsiteIP, CLIP
__count	<Double>	Read-only	count parameter

Operations

(click to see Properties)

ADD| DELETE| UPDATE| UNSET| ENABLE| DISABLE| GET (ALL)| COUNT

Some options that you can use for each operations:

- **Getting warnings in response:**NITRO allows you to get warnings in an operation by specifying the "warning" query parameter as "yes". For example, to get warnings while connecting to the Citrix ADC appliance, the URL is as follows:

  http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/login?warning=yes

  If any, the warnings are displayed in the response payload with the HTTP code "209 X-NITRO-WARNING".

- **Authenticated access for individual NITRO operations:**NITRO allows you to logon to the Citrix ADC appliance to perform individual operations. You can use this option instead of creating a NITRO session (using the login object) and then using that session to perform all operations,

  To do this, you must specify the username and password in the request header of the NITRO request as follows:

  X-NITRO-USER:<username>

  X-NITRO-PASS:<password>

  Note: In such cases, make sure that the request header DOES not include the following:

  Cookie:NITRO_AUTH_TOKEN=<tokenvalue>

`*Note:`*

Mandatory parameters are marked in redand placeholder content is marked in <green>.

add

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/nsip

HTTP Method:POST

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Content-Type:application/json

Request Payload:

```
{"nsip":{

<b>"ipaddress":<String_value>,

</b><b>"netmask":<String_value>,

</b>"type":<String_value>,

"arp":<String_value>,

"icmp":<String_value>,

"vserver":<String_value>,

"telnet":<String_value>,

"ftp":<String_value>,

"gui":<String_value>,

"ssh":<String_value>,

"snmp":<String_value>,

"mgmtaccess":<String_value>,

"restrictaccess":<String_value>,

"dynamicrouting":<String_value>,
```

```
"decrementttl":<String_value>,

"ospf":<String_value>,

"bgp":<String_value>,

"rip":<String_value>,

"hostroute":<String_value>,

"advertiseondefaultpartition":<String_value>,

"networkroute":<String_value>,

"tag":<Double_value>,

"hostrtgw":<String_value>,

"metric":<Integer_value>,

"vserverrhilevel":<String_value>,

"vserverrhimode":<String_value>,

"ospflsatype":<String_value>,

"ospfarea":<Double_value>,

"state":<String_value>,

"vrid":<Double_value>,

"icmpresponse":<String_value>,

"ownernode":<Double_value>,

"arpresponse":<String_value>,

"ownerdownresponse":<String_value>,

"td":<Double_value>

}}
```

Response:

HTTP Status Code on Success: 201 Created HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error

delete

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/nsip/ipaddress_value<String>

HTTP Method:DELETE

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue>

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error

update

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/nsip

HTTP Method:PUT

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Content-Type:application/json

Request Payload:

```
{"nsip":{

<b>"ipaddress":<String_value>,

</b>"td":<Double_value>,

"netmask":<String_value>,
```

```
"arp":<String_value>,

"icmp":<String_value>,

"vserver":<String_value>,

"telnet":<String_value>,

"ftp":<String_value>,

"gui":<String_value>,

"ssh":<String_value>,

"snmp":<String_value>,

"mgmtaccess":<String_value>,

"restrictaccess":<String_value>,

"decrementttl":<String_value>,

"dynamicrouting":<String_value>,

"ospf":<String_value>,

"bgp":<String_value>,

"rip":<String_value>,

"hostroute":<String_value>,

"advertiseondefaultpartition":<String_value>,

"networkroute":<String_value>,

"hostrtgw":<String_value>,

"metric":<Integer_value>,

"vserverrhilevel":<String_value>,

"vserverrhimode":<String_value>,
```

```
"ospflsatype":<String_value>,

"ospfarea":<Double_value>,

"tag":<Double_value>,

"vrid":<Double_value>,

"icmpresponse":<String_value>,

"arpresponse":<String_value>,

"ownerdownresponse":<String_value>

}}
```

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error

unset

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/nsip?**action=unset**

HTTP Method:POST

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Content-Type:application/json

Request Payload:

```
{"nsip":{

<b>"ipaddress":<String_value>,

</b>"td":<Double_value>,
```

```
"ospfarea":true,

"hostrtgw":true,

"netmask":true,

"arp":true,

"icmp":true,

"vserver":true,

"telnet":true,

"ftp":true,

"gui":true,

"ssh":true,

"snmp":true,

"mgmtaccess":true,

"restrictaccess":true,

"decrementttl":true,

"dynamicrouting":true,

"ospf":true,

"bgp":true,

"rip":true,

"hostroute":true,

"advertiseondefaultpartition":true,

"networkroute":true,

"metric":true,
```

```
"vserverrhilevel":true,

"vserverrhimode":true,

"ospflsatype":true,

"tag":true,

"vrid":true,

"icmpresponse":true,

"arpresponse":true,

"ownerdownresponse":true

}}
```

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error

enable

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/nsip?**action=enable**

HTTP Method:POST

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Content-Type:application/json

Request Payload:

```
{"nsip":{

<b>"ipaddress":<String_value>,
```

```
</b>"td":<Double_value>

}}
```

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error

disable

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/nsip?**action=disable**

HTTP Method:POST

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Content-Type:application/json

Request Payload:

```
{"nsip":{

<b>"ipaddress":<String_value>,

</b>"td":<Double_value>

}}
```

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error

get (all)

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/nsip

Query-parameters:

args

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/nsip?**args=ipaddress:<String_value>,td:<Double_value>,type:<String_value>**

Use this query-parameter to get nsip resources based on additional properties.

attrs

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/nsip?**attrs=property-name1,property-name2**

Use this query parameter to specify the resource details that you want to retrieve.

filter

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/nsip?**filter=property-name1:property-val1,property-name2:property-val2**

Use this query-parameter to get the filtered set of nsip resources configured on Citrix ADC. Filtering can be done on any of the properties of the resource.

view

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/nsip?**view=summary**

Note: By default, the retrieved results are displayed in detail view (?view=detail).

pagination

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/nsip?**pagesize=#no;pageno=#no**

Use this query-parameter to get the nsip resources in chunks.

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "nsip": [ {

ipaddress:<String_value>,td:<Double_value>,type:<String_value>"netmask":<S
tring_value>,

"flags":<Double_value>,

"arp":<String_value>,

"icmp":<String_value>,

"vserver":<String_value>,

"telnet":<String_value>,

"ssh":<String_value>,

"gui":<String_value>,

"snmp":<String_value>,

"ftp":<String_value>,

"mgmtaccess":<String_value>,

"restrictaccess":<String_value>,

"decrementttl":<String_value>,

"dynamicrouting":<String_value>,

"bgp":<String_value>,
```

```
"ospf":<String_value>,

"rip":<String_value>,

"hostroute":<String_value>,

"advertiseondefaultpartition":<String_value>,

"networkroute":<String_value>,

"tag":<Double_value>,

"hostrtgw":<String_value>,

"hostrtgwact":<String_value>,

"metric":<Integer_value>,

"ospfarea":<Double_value>,

"ospfareaval":<Double_value>,

"vserverrhilevel":<String_value>,

"vserverrhimode":<String_value>,

"viprtadv2bsd":<Boolean_value>,

"vipvsercount":<Double_value>,

"vipvserdowncount":<Double_value>,

"vipvsrvrrhiactivecount":<Double_value>,

"vipvsrvrrhiactiveupcount":<Double_value>,

"ospflsatype":<String_value>,

"state":<String_value>,

"freeports":<Double_value>,

"vrid":<Double_value>,
```

```
"riserhimsgcode":<Integer_value>,

"iptype":<String[]_value>,

"icmpresponse":<String_value>,

"ownernode":<Double_value>,

"arpresponse":<String_value>,

"ownerdownresponse":<String_value>

}]}
```

count

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/nsip?**count=yes**

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "nsip": [ { "__count": "#no"} ] }
```

nsip6

Configuration for ip6 resource.

Properties

(click to see Operations)

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
ipv6address	<String>	Read-write	IPv6 address to create on the Citrix ADC. Minimum length = 1
scope	<String>	Read-write	Scope of the IPv6 address to be created. Cannot be changed after the IP address is created. Default value: global Possible values = global, link-local
type	<String>	Read-write	Type of IP address to be created on the Citrix ADC. Cannot be changed after the IP address is created. Default value: SNIP Possible values = NSIP, VIP, SNIP, GSLBsiteIP, ADNSsvcIP, RADIUSListenersvcIP, CLIP
vlan	<Double>	Read-write	The VLAN number. Default value: 0 Minimum value = 0

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
			Maximum value = 4094
nd	<String>	Read-write	Respond to Neighbor Discovery (ND) requests for this IP address. Default value: ENABLED Possible values = ENABLED, DISABLED
icmp	<String>	Read-write	Respond to ICMP requests for this IP address. Default value: ENABLED Possible values = ENABLED, DISABLED
vserver	<String>	Read-write	Enable or disable the state of all the virtual servers associated with this VIP6 address. Default value: ENABLED Possible values = ENABLED, DISABLED
telnet	<String>	Read-write	Allow Telnet access to this IP address.

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
			Default value: ENABLED Possible values = ENABLED, DISABLED
ftp	<String>	Read-write	Allow File Transfer Protocol (FTP) access to this IP address. Default value: ENABLED Possible values = ENABLED, DISABLED
gui	<String>	Read-write	Allow graphical user interface (GUI) access to this IP address. Default value: ENABLED Possible values = ENABLED, SECUREONLY, DISABLED
ssh	<String>	Read-write	Allow secure Shell (SSH) access to this IP address. Default value: ENABLED Possible values = ENABLED, DISABLED

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
snmp	<String>	Read-write	Allow Simple Network Management Protocol (SNMP) access to this IP address. Default value: ENABLED Possible values = ENABLED, DISABLED
mgmtaccess	<String>	Read-write	Allow access to management applications on this IP address. Default value: DISABLED Possible values = ENABLED, DISABLED
restrictaccess	<String>	Read-write	Block access to nonmanagement applications on this IP address. This option is applicable forMIP6s, SNIP6s, and NSIP6s, and is disabled by default. Nonmanagement applications can run on the underlying Citrix ADC Free BSD operating system. Default value: DISABLED

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
			Possible values = ENABLED, DISABLED
dynamicrouting	<String>	Read-write	Allow dynamic routing on this IP address. Specific to Subnet IPv6 (SNIP6) address. Default value: DISABLED Possible values = ENABLED, DISABLED
decrementhoplimit	<String>	Read-write	Decrement Hop Limit by 1 when ENABLED.This setting is applicable only for UDP traffic. Default value: DISABLED Possible values = ENABLED, DISABLED
hostroute	<String>	Read-write	Option to push the VIP6 to ZebOS routing table for Kernel route redistribution through dynamic routing protocols. Possible values = ENABLED, DISABLED

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
advertiseondefaultpartition	<String>	Read-write	Advertise VIPs from Shared VLAN on Default Partition. Default value: DISABLED Possible values = ENABLED, DISABLED
networkroute	<String>	Read-write	Option to push the SNIP6 subnet to ZebOS routing table for Kernel route redistribution through dynamic routing protocol. Possible values = ENABLED, DISABLED
tag	<Double>	Read-write	Tag value for the network/host route associated with this IP. Default value: 0
ip6hostrtgw	<String>	Read-write	IPv6 address of the gateway for the route. If Gateway is not set, VIP uses :: as the gateway. Default value: 0
metric	<Integer>	Read-write	Integer value to add to or subtract from the cost of the route advertised for the VIP6 address.

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
			Minimum value = -16777215
vserverrhilevel	<String>	Read-write	Advertise or do not advertise the route for the Virtual IP (VIP6) address on the basis of the state of the virtual servers associated with that VIP6. * NONE - Advertise the route for the VIP6 address, irrespective of the state of the virtual servers associated with the address. * ONE VSERVER - Advertise the route for the VIP6 address if at least one of the associated virtual servers is in UP state. * ALL VSERVER - Advertise the route for the VIP6 address if all of the associated virtual servers are in UP state.

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
			* VSVR_CNTRLD. Advertise the route for the VIP address according to the RHIstate (RHI STATE) parameter setting on all the associated virtual servers of the VIP address along with their states. When Vserver RHI Level (RHI) parameter is set to VSVR_CNTRLD, the following are different RHI behaviors for the VIP address on the basis of RHIstate (RHI STATE) settings on the virtual servers associated with the VIP address: * If you set RHI STATE to PASSIVE on all virtual servers, the Citrix ADC always advertises the route for the VIP address.

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
			* If you set RHI STATE to ACTIVE on all virtual servers, the Citrix ADC advertises the route for the VIP address if at least one of the associated virtual servers is in UP state. *If you set RHI STATE to ACTIVE on some and PASSIVE on others, the Citrix ADC advertises the route for the VIP address if at least one of the associated virtual servers, whose RHI STATE set to ACTIVE, is in UP state. Default value: ONE_VSERVER Possible values = ONE_VSERVER, ALL_VSERVERS, NONE, VSVR_CNTRLD
ospf6lsatype	<String>	Read-write	Type of LSAs to be used by the IPv6 OSPF protocol,

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
			running on the Citrix ADC, for advertising the route for the VIP6 address. Default value: EXTERNAL Possible values = INTRA_AREA, EXTERNAL
ospfarea	<Double>	Read-write	ID of the area in which the Intra-Area-Prefix LSAs are to be advertised for the VIP6 address by the IPv6 OSPF protocol running on the Citrix ADC. When ospfArea is not set, VIP6 is advertised on all areas. Default value: -1 Minimum value = 0 Maximum value = 4294967294LU
state	<String>	Read-write	Enable or disable the IP address. Default value: ENABLED Possible values = DISABLED, ENABLED

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
map	<String>	Read-write	Mapped IPV4 address for the IPV6 address.
vrid6	<Double>	Read-write	A positive integer that uniquely identifies a VMAC address for binding to this VIP address. This binding is used to set up Citrix ADCs in an active-active configuration using VRRP. Minimum value = 1 Maximum value = 255
ownernode	<Double>	Read-write	ID of the cluster node for which you are adding the IP address. Must be used if you want the IP address to be active only on the specific node. Can be configured only through the cluster IP address. Cannot be changed after the IP address is created. Default value: 255
ownerdownresponse	<String>	Read-write	in cluster system, if the owner node is down, whether should it respond to icmp/arp.

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
			Default value: YES Possible values = YES, NO
td	<Double>	Read-write	Integer value that uniquely identifies the traffic domain in which you want to configure the entity. If you do not specify an ID, the entity becomes part of the default traffic domain, which has an ID of 0. Minimum value = 0 Maximum value = 4094
iptype	<String[]>	Read-only	The type of the IPv6 address. Possible values = NSIP, VIP, SNIP, GSLBsiteIP, ADNSsvcIP, RADIUSListenersvcIP, CLIP
curstate	<String>	Read-only	Current state of this IP. Default value: ENABLED Possible values = DISABLED, ENABLED

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
viprtadv2bsd	<Boolean>	Read-only	Whether this route is advertised to FreeBSD.
vipvsercount	<Double>	Read-only	Number of vservers bound to this VIP.
vipvserdowncount	<Double>	Read-only	Number of vservers bound to this VIP, which are down.
systemtype	<String>	Read-only	The type of the System. Possible Values: Standalone, HA, Cluster. Used for display purpose. Possible values = Stand-alone, HA, Cluster
__count	<Double>	Read-only	count parameter

Operations

(click to see Properties)

ADD| DELETE| UPDATE| UNSET| GET (ALL)| COUNT

Some options that you can use for each operations:

- **Getting warnings in response:**NITRO allows you to get warnings in an operation by specifying the "warning" query parameter as "yes". For example, to get warnings while connecting to the Citrix ADC appliance, the URL is as follows:

  http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/login?warning=yes

  If any, the warnings are displayed in the response payload with the HTTP code "209 X-NITRO-WARNING".
- **Authenticated access for individual NITRO operations:**NITRO allows you to logon to the Citrix ADC appliance to perform individual operations. You can use this option instead of

creating a NITRO session (using the login object) and then using that session to perform all operations,

To do this, you must specify the username and password in the request header of the NITRO request as follows:

X-NITRO-USER:<username>

X-NITRO-PASS:<password>

Note: In such cases, make sure that the request header DOES not include the following:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue>

`*Note:`*

Mandatory parameters are marked in redand placeholder content is marked in <green>.

add

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/nsip6

HTTP Method:POST

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Content-Type:application/json

Request Payload:

```
{"nsip6":{

<b>"ipv6address":<String_value>,

</b>"scope":<String_value>,

"type":<String_value>,

"vlan":<Double_value>,

"nd":<String_value>,

"icmp":<String_value>,
```

```
"vserver":<String_value>,

"telnet":<String_value>,

"ftp":<String_value>,

"gui":<String_value>,

"ssh":<String_value>,

"snmp":<String_value>,

"mgmtaccess":<String_value>,

"restrictaccess":<String_value>,

"dynamicrouting":<String_value>,

"decrementhoplimit":<String_value>,

"hostroute":<String_value>,

"advertiseondefaultpartition":<String_value>,

"networkroute":<String_value>,

"tag":<Double_value>,

"ip6hostrtgw":<String_value>,

"metric":<Integer_value>,

"vserverrhilevel":<String_value>,

"ospf6lsatype":<String_value>,

"ospfarea":<Double_value>,

"state":<String_value>,

"map":<String_value>,

"vrid6":<Double_value>,
```

```
"ownernode":<Double_value>,

"ownerdownresponse":<String_value>,

"td":<Double_value>

}}
```

Response:

HTTP Status Code on Success: 201 Created HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error

delete

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/nsip6/ipv6address_value<String>

HTTP Method:DELETE

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue>

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error

update

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/nsip6

HTTP Method:PUT

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Content-Type:application/json

Request Payload:

```
{"nsip6":{

<b>"ipv6address":<String_value>,

</b>"td":<Double_value>,

"nd":<String_value>,

"icmp":<String_value>,

"vserver":<String_value>,

"telnet":<String_value>,

"ftp":<String_value>,

"gui":<String_value>,

"ssh":<String_value>,

"snmp":<String_value>,

"mgmtaccess":<String_value>,

"ownerdownresponse":<String_value>,

"restrictaccess":<String_value>,

"state":<String_value>,

"map":<String_value>,

"decrementhoplimit":<String_value>,

"dynamicrouting":<String_value>,

"hostroute":<String_value>,

"advertiseondefaultpartition":<String_value>,

"networkroute":<String_value>,
```

```
"ip6hostrtgw":<String_value>,

"metric":<Integer_value>,

"vserverrhilevel":<String_value>,

"ospf6lsatype":<String_value>,

"ospfarea":<Double_value>,

"tag":<Double_value>,

"vrid6":<Double_value>

}}
```

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error

unset

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/nsip6?**action=unset**

HTTP Method:POST

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Content-Type:application/json

Request Payload:

```
{"nsip6":{

<b>"ipv6address":<String_value>,

</b>"td":<Double_value>,
```

```
"ospfarea":true,

"nd":true,

"icmp":true,

"vserver":true,

"telnet":true,

"ftp":true,

"gui":true,

"ssh":true,

"snmp":true,

"mgmtaccess":true,

"ownerdownresponse":true,

"restrictaccess":true,

"state":true,

"map":true,

"decrementhoplimit":true,

"dynamicrouting":true,

"hostroute":true,

"advertiseondefaultpartition":true,

"networkroute":true,

"ip6hostrtgw":true,

"metric":true,

"vserverrhilevel":true,
```

```
"ospf6lsatype":true,

"tag":true,

"vrid6":true

}}
```

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error

get (all)

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/nsip6

Query-parameters:

attrs

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/nsip6?**attrs=property-name1,property-name2**

Use this query parameter to specify the resource details that you want to retrieve.

filter

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/nsip6?**filter=property-name1:property-val1,property-name2:property-val2**

Use this query-parameter to get the filtered set of nsip6 resources configured on Citrix ADC. Filtering can be done on any of the properties of the resource.

view

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/nsip6?**view=summary**

Note: By default, the retrieved results are displayed in detail view (?view=detail).

pagination

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/nsip6?**pagesize=#no;pageno=#no**

Use this query-parameter to get the nsip6 resources in chunks.

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "nsip6": [ {

"ipv6address":<String_value>,

"td":<Double_value>,

"scope":<String_value>,

"type":<String_value>,

"iptype":<String[]_value>,

"vlan":<Double_value>,

"nd":<String_value>,

"icmp":<String_value>,

"vserver":<String_value>,

"telnet":<String_value>,
```

```
"ssh":<String_value>,

"gui":<String_value>,

"snmp":<String_value>,

"ftp":<String_value>,

"mgmtaccess":<String_value>,

"restrictaccess":<String_value>,

"state":<String_value>,

"curstate":<String_value>,

"map":<String_value>,

"decrementhoplimit":<String_value>,

"dynamicrouting":<String_value>,

"hostroute":<String_value>,

"advertiseondefaultpartition":<String_value>,

"networkroute":<String_value>,

"tag":<Double_value>,

"ip6hostrtgw":<String_value>,

"metric":<Integer_value>,

"vserverrhilevel":<String_value>,

"viprtadv2bsd":<Boolean_value>,

"vipvsercount":<Double_value>,

"vipvserdowncount":<Double_value>,

"ospf6lsatype":<String_value>,
```

```
"ospfarea":<Double_value>,

"ownernode":<Double_value>,

"vrid6":<Double_value>,

"ownerdownresponse":<String_value>,

"systemtype":<String_value>

}]}
```

count

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/nsip6?**count=yes**

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "nsip6": [ { "__count": "#no"} ] }
```

nslicense

Configuration for license resource.

Properties

(click to see Operations)

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
wl	<Boolean>	Read-only	Web Logging.
sp	<Boolean>	Read-only	Surge Protection.
lb	<Boolean>	Read-only	Load Balancing.
cs	<Boolean>	Read-only	Content Switching.
cr	<Boolean>	Read-only	Cache Redirect.
sc	<Boolean>	Read-only	Sure Connect.
cmp	<Boolean>	Read-only	Compression.
delta	<Boolean>	Read-only	Delta Compression.
pq	<Boolean>	Read-only	Priority Queuing.
ssl	<Boolean>	Read-only	Secure Sockets Layer.
gslb	<Boolean>	Read-only	Global Server Load Balancing.
gslbp	<Boolean>	Read-only	GSLB Proximity.
hdosp	<Boolean>	Read-only	DOS Protection.
routing	<Boolean>	Read-only	Routing.
cf	<Boolean>	Read-only	Content Filter.
contentaccelerator	<Boolean>	Read-only	transparent Integrated Caching.
ic	<Boolean>	Read-only	Integrated Caching.
sslvpn	<Boolean>	Read-only	SSL VPN.

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
f_sslvpn_users	<Double>	Read-only	Number of licensed users allowed by this license.
f_ica_users	<Double>	Read-only	Number of licensed users allowed by ICAONLY license. As long as the AG Feature is licensed, unlimited number of ICA connections are accepted. (In API, 0 value for this parameter means unlimited when AG license in ON).
aaa	<Boolean>	Read-only	AAA.
ospf	<Boolean>	Read-only	OSPF Routing.
rip	<Boolean>	Read-only	RIP Routing.
bgp	<Boolean>	Read-only	BGP Routing.
rewrite	<Boolean>	Read-only	Rewrite.
ipv6pt	<Boolean>	Read-only	IPv6 protocol translation.
appfw	<Boolean>	Read-only	Application Firewall.
responder	<Boolean>	Read-only	Responder.
agee	<Boolean>	Read-only	.
nsxn	<Boolean>	Read-only	.
htmlinjection	<Boolean>	Read-only	HTML Injection.

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
modelid	<Double>	Read-only	Model Number ID.
push	<Boolean>	Read-only	Citrix ADC Push.
wionns	<Boolean>	Read-only	WI on NS.
appflow	<Boolean>	Read-only	AppFlow.
cloudbridge	<Boolean>	Read-only	CloudBridge.
cloudbridgeappliance	<Boolean>	Read-only	.
cloudextenderappliance	<Boolean>	Read-only	.
isis	<Boolean>	Read-only	ISIS Routing.
cluster	<Boolean>	Read-only	Clustering.
ch	<Boolean>	Read-only	Call Home.
appqoe	<Boolean>	Read-only	AppQoS.
appflowica	<Boolean>	Read-only	Appflow for ICA.
isstandardlic	<Boolean>	Read-only	Standard License.
isenterpriselic	<Boolean>	Read-only	Enterprise License.
isplatinumlic	<Boolean>	Read-only	Platinum License.
issgwylic	<Boolean>	Read-only	Simple Gateway License.
isswglic	<Boolean>	Read-only	Secure Web Gateway License.
rise	<Boolean>	Read-only	RISE.
feo	<Boolean>	Read-only	Front End Optimization.

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
lsn	<Boolean>	Read-only	Large Scale NAT.
licensingmode	<String>	Read-only	Pooled Licensed. Default value: Local Possible values = Local, Pooled, CICO, EXPRESS, CPU, CPU-Local
daystoexpiration	<Double>	Read-only	Days to expire. Minimum value = 0 Maximum value = 4294967294
rdpproxy	<Boolean>	Read-only	RDPPROXY.
rep	<Boolean>	Read-only	Reputation Services.
urlfiltering	<Boolean>	Read-only	URL Filtering.
videooptimization	<Boolean>	Read-only	Video Optimization.
forwardproxy	<Boolean>	Read-only	Forward Proxy.
sslinterception	<Boolean>	Read-only	SSL Interception.
remotecontentinspection	<Boolean>	Read-only	Remote Content Inspection.
adaptivetcp	<Boolean>	Read-only	Adaptive TCP.
cqa	<Boolean>	Read-only	Connection Quality Analytics.

Operations

(click to see Properties)

GET (ALL)

Some options that you can use for each operations:

- **Getting warnings in response:**NITRO allows you to get warnings in an operation by specifying the "warning" query parameter as "yes". For example, to get warnings while connecting to the Citrix ADC appliance, the URL is as follows:

  http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/login?warning=yes

  If any, the warnings are displayed in the response payload with the HTTP code "209 X-NITRO-WARNING".

- **Authenticated access for individual NITRO operations:**NITRO allows you to logon to the Citrix ADC appliance to perform individual operations. You can use this option instead of creating a NITRO session (using the login object) and then using that session to perform all operations,

  To do this, you must specify the username and password in the request header of the NITRO request as follows:

  X-NITRO-USER:<username>

  X-NITRO-PASS:<password>

  Note: In such cases, make sure that the request header DOES not include the following:

  Cookie:NITRO_AUTH_TOKEN=<tokenvalue>

`*Note:`*

Mandatory parameters are marked in redand placeholder content is marked in <green>.

get (all)

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/nslicense

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "nslicense": [ {

"wl":<Boolean_value>,

"sp":<Boolean_value>,

"lb":<Boolean_value>,

"cs":<Boolean_value>,

"cr":<Boolean_value>,

"sc":<Boolean_value>,

"cmp":<Boolean_value>,

"delta":<Boolean_value>,

"pq":<Boolean_value>,

"ssl":<Boolean_value>,

"gslb":<Boolean_value>,

"gslbp":<Boolean_value>,

"hdosp":<Boolean_value>,

"routing":<Boolean_value>,

"cf":<Boolean_value>,

"contentaccelerator":<Boolean_value>,

"ic":<Boolean_value>,

"sslvpn":<Boolean_value>,
```

```
"f_sslvpn_users":<Double_value>,

"f_ica_users":<Double_value>,

"aaa":<Boolean_value>,

"ospf":<Boolean_value>,

"rip":<Boolean_value>,

"bgp":<Boolean_value>,

"rewrite":<Boolean_value>,

"ipv6pt":<Boolean_value>,

"appfw":<Boolean_value>,

"responder":<Boolean_value>,

"agee":<Boolean_value>,

"nsxn":<Boolean_value>,

"htmlinjection":<Boolean_value>,

"modelid":<Double_value>,

"push":<Boolean_value>,

"wionns":<Boolean_value>,

"appflow":<Boolean_value>,

"cloudbridge":<Boolean_value>,

"cloudbridgeappliance":<Boolean_value>,

"cloudextenderappliance":<Boolean_value>,

"isis":<Boolean_value>,

"cluster":<Boolean_value>,
```

```
"ch":<Boolean_value>,

"appqoe":<Boolean_value>,

"appflowica":<Boolean_value>,

"isstandardlic":<Boolean_value>,

"isenterpriselic":<Boolean_value>,

"isplatinumlic":<Boolean_value>,

"issgwylic":<Boolean_value>,

"isswglic":<Boolean_value>,

"rise":<Boolean_value>,

"feo":<Boolean_value>,

"lsn":<Boolean_value>,

"licensingmode":<String_value>,

"daystoexpiration":<Double_value>,

"rdpproxy":<Boolean_value>,

"rep":<Boolean_value>,

"urlfiltering":<Boolean_value>,

"videooptimization":<Boolean_value>,

"forwardproxy":<Boolean_value>,

"sslinterception":<Boolean_value>,

"remotecontentinspection":<Boolean_value>,

"adaptivetcp":<Boolean_value>,

"cqa":<Boolean_value>
```

```
}]}
```

nslicenseproxyserver

Configuration for licenseproxyserver resource.

Properties

(click to see Operations)

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
serverip	<String>	Read-write	IP address of the License proxy server. Minimum length = 1
servername	<String>	Read-write	Fully qualified domain name of the License proxy server.
port	<Double>	Read-write	License proxy server port.
__count	<Double>	Read-only	count parameter

Operations

(click to see Properties)

ADD| DELETE| UPDATE| GET (ALL)| COUNT

Some options that you can use for each operations:

- **Getting warnings in response:**NITRO allows you to get warnings in an operation by specifying the "warning" query parameter as "yes". For example, to get warnings while connecting to the Citrix ADC appliance, the URL is as follows:

  http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/login?warning=yes

  If any, the warnings are displayed in the response payload with the HTTP code "209 X-NITRO-WARNING".

- **Authenticated access for individual NITRO operations:**NITRO allows you to logon to the Citrix ADC appliance to perform individual operations. You can use this option instead of creating a NITRO session (using the login object) and then using that session to perform all operations,

  To do this, you must specify the username and password in the request header of the NITRO request as follows:

  X-NITRO-USER:<username>

  X-NITRO-PASS:<password>

  Note: In such cases, make sure that the request header DOES not include the following:

  Cookie:NITRO_AUTH_TOKEN=<tokenvalue>

`*Note:`*

Mandatory parameters are marked in redand placeholder content is marked in <green>.

add

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/nslicenseproxyserver

HTTP Method:POST

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Content-Type:application/json

Request Payload:

```
{"nslicenseproxyserver":{

"serverip":<String_value>,

"servername":<String_value>,

<b>"port":<Double_value>

</b>}}
```

Response:

HTTP Status Code on Success: 201 Created HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error

delete

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/nslicenseproxyserver/serverip_value<String>

HTTP Method:DELETE

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue>

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error

update

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/nslicenseproxyserver

HTTP Method:PUT

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Content-Type:application/json

Request Payload:

```
{"nslicenseproxyserver":{

"serverip":<String_value>,

"servername":<String_value>,

<b>"port":<Double_value>

</b>}}
```

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error

get (all)

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/nslicenseproxyserver

Query-parameters:

attrs

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/nslicenseproxyserver?**attrs=property-name1,property-name2**

Use this query parameter to specify the resource details that you want to retrieve.

filter

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/nslicenseproxyserver?**filter=property-name1:property-val1,property-name2:property-val2**

Use this query-parameter to get the filtered set of nslicenseproxyserver resources configured on Citrix ADC. Filtering can be done on any of the properties of the resource.

view

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/nslicenseproxyserver?**view=summary**

Note: By default, the retrieved results are displayed in detail view (?view=detail).

pagination

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/nslicenseproxyserver?**pagesize=#no;pageno=#no**

Use this query-parameter to get the nslicenseproxyserver resources in chunks.

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "nslicenseproxyserver": [ {

"serverip":<String_value>,

"servername":<String_value>,

"port":<Double_value>

}]}
```

count

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/nslicenseproxyserver?**count=yes**

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "nslicenseproxyserver": [ { "__count": "#no"} ] }
```

nslicenseserver

Configuration for licenseserver resource.

Properties

(click to see Operations)

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
licenseserverip	<String>	Read-write	IP address of the License server. Minimum length = 1
servername	<String>	Read-write	Fully qualified domain name of the License server.
port	<Double>	Read-write	License server port.
nodeid	<Double>	Read-write	Unique number that identifies the cluster node. Minimum value = 0 Maximum value = 31
status	<Double>	Read-only	Status of license server.
grace	<Double>	Read-only	Grace status of server.
gptimeleft	<Double>	Read-only	Grace time left.
__count	<Double>	Read-only	count parameter

Operations

(click to see Properties)

ADD| DELETE| UPDATE| GET (ALL)| COUNT

Some options that you can use for each operations:

- **Getting warnings in response:**NITRO allows you to get warnings in an operation by specifying the "warning" query parameter as "yes". For example, to get warnings while connecting to the Citrix ADC appliance, the URL is as follows:

  http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/login?warning=yes

  If any, the warnings are displayed in the response payload with the HTTP code "209 X-NITRO-WARNING".

- **Authenticated access for individual NITRO operations:**NITRO allows you to logon to the Citrix ADC appliance to perform individual operations. You can use this option instead of creating a NITRO session (using the login object) and then using that session to perform all operations,

  To do this, you must specify the username and password in the request header of the NITRO request as follows:

  X-NITRO-USER:<username>

  X-NITRO-PASS:<password>

  Note: In such cases, make sure that the request header DOES not include the following:

  Cookie:NITRO_AUTH_TOKEN=<tokenvalue>

`*Note:`*

Mandatory parameters are marked in redand placeholder content is marked in <green>.

add

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/nslicenseserver

HTTP Method:POST

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Content-Type:application/json

Request Payload:

```
{"nslicenseserver":{

"licenseserverip":<String_value>,

"servername":<String_value>,

"port":<Double_value>

}}
```

Response:

HTTP Status Code on Success: 201 Created HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error

delete

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/nslicenseserver/licenseserverip_value<String>

HTTP Method:DELETE

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue>

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error

update

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/nslicenseserver

HTTP Method:PUT

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Content-Type:application/json

Request Payload:

```
{"nslicenseserver":{

"licenseserverip":<String_value>,

"servername":<String_value>,

"port":<Double_value>

}}
```

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error

get (all)

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/nslicenseserver

Query-parameters:

args

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/nslicenseserver?**args=licenseserverip:<String_value>,servername:<String_value>,nodeid:<Double_value>**

Use this query-parameter to get nslicenseserver resources based on additional properties.

attrs

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/nslicenseserver?**attrs=property-name1,property-name2**

Use this query parameter to specify the resource details that you want to retrieve.

filter

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/nslicenseserver?**filter=property-name1:property-val1,property-name2:property-val2**

Use this query-parameter to get the filtered set of nslicenseserver resources configured on Citrix ADC. Filtering can be done on any of the properties of the resource.

view

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/nslicenseserver?**view=summary**

Note: By default, the retrieved results are displayed in detail view (?view=detail).

pagination

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/nslicenseserver?**pagesize=#no;pageno=#no**

Use this query-parameter to get the nslicenseserver resources in chunks.

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "nslicenseserver": [ {

licenseserverip:<String_value>,servername:<String_value>,nodeid:<Double_va
lue>"port":<Double_value>,

"status":<Double_value>,

"grace":<Double_value>,
```

```
"gptimeleft":<Double_value>

}]}
```

count

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/nslicenseserver?**count=yes**

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "nslicenseserver": [ { "__count": "#no"} ] }
```

nslicenseserverpool

Configuration for licenseserver resource.

Properties

(click to see Operations)

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
instancetotal	<Integer>	Read-only	Instance Total.
instanceavailable	<Integer>	Read-only	Instance Available.

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
standardbandwidthtotal	<Integer>	Read-only	Standard Bandwidth Total.
standardbandwidthavailable	<Integer>	Read-only	Standard Bandwidth Available.
enterprisebandwidthtotal	<Integer>	Read-only	Enterprise Bandwidth Total.
enterprisebandwidthavailable	<Integer>	Read-only	Enterprise Bandwidth Available.
platinumbandwidthtotal	<Integer>	Read-only	Platinum Bandwidth Total.
platinumbandwidthavailable	<Integer>	Read-only	Platinum Bandwidth Available.
standardcputotal	<Integer>	Read-only	Standard CPU Total.
standardcpuavailable	<Integer>	Read-only	Standard CPU Available.
enterprisecputotal	<Integer>	Read-only	Enterprise CPU Total.
enterprisecpuavailable	<Integer>	Read-only	Enterprise CPU Available.
platinumcputotal	<Integer>	Read-only	Platinum CPU Total.
platinumcpuavailable	<Integer>	Read-only	Platinum CPU Available.
cpxinstancetotal	<Integer>	Read-only	CP1000 Instance Total.
cpxinstanceavailable	<Integer>	Read-only	CP1000 Instance Available.
vpx1stotal	<Integer>	Read-only	VPX1S Total.

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
vpx1savailable	<Integer>	Read-only	VPX1S Available.
vpx1ptotal	<Integer>	Read-only	VPX1P Total.
vpx1pavailable	<Integer>	Read-only	VPX1P Available.
vpx5stotal	<Integer>	Read-only	VPX5S Total.
vpx5savailable	<Integer>	Read-only	VPX5S Available.
vpx5ptotal	<Integer>	Read-only	VPX5P Total.
vpx5pavailable	<Integer>	Read-only	VPX5P Available.
vpx10stotal	<Integer>	Read-only	VPX10S Total.
vpx10savailable	<Integer>	Read-only	VPX10S Available.
vpx10etotal	<Integer>	Read-only	VPX10E Total.
vpx10eavailable	<Integer>	Read-only	VPX10E Available.
vpx10ptotal	<Integer>	Read-only	VPX10P Total.
vpx10pavailable	<Integer>	Read-only	VPX10P Available.
vpx25stotal	<Integer>	Read-only	VPX25S Total.
vpx25savailable	<Integer>	Read-only	VPX25S Available.
vpx25etotal	<Integer>	Read-only	VPX25E Total.
vpx25eavailable	<Integer>	Read-only	VPX25E Available.
vpx25ptotal	<Integer>	Read-only	VPX25P Total.
vpx25pavailable	<Integer>	Read-only	VPX25P Available.
vpx50stotal	<Integer>	Read-only	VPX50S Total.
vpx50savailable	<Integer>	Read-only	VPX50S Available.

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
vpx50etotal	<Integer>	Read-only	VPX50E Total.
vpx50eavailable	<Integer>	Read-only	VPX50E Available.
vpx50ptotal	<Integer>	Read-only	VPX50P Total.
vpx50pavailable	<Integer>	Read-only	VPX50P Available.
vpx100stotal	<Integer>	Read-only	VPX100S Total.
vpx100savailable	<Integer>	Read-only	VPX100S Available.
vpx100etotal	<Integer>	Read-only	VPX100E Total.
vpx100eavailable	<Integer>	Read-only	VPX100E Available.
vpx100ptotal	<Integer>	Read-only	VPX100P Total.
vpx100pavailable	<Integer>	Read-only	VPX100P Available.
vpx200stotal	<Integer>	Read-only	VPX200S Total.
vpx200savailable	<Integer>	Read-only	VPX200S Available.
vpx200etotal	<Integer>	Read-only	VPX200E Total.
vpx200eavailable	<Integer>	Read-only	VPX200E Available.
vpx200ptotal	<Integer>	Read-only	VPX200P Total.
vpx200pavailable	<Integer>	Read-only	VPX200P Available.
vpx500stotal	<Integer>	Read-only	VPX500S Total.
vpx500savailable	<Integer>	Read-only	VPX500S Available.
vpx500etotal	<Integer>	Read-only	VPX500E Total.
vpx500eavailable	<Integer>	Read-only	VPX500E Available.
vpx500ptotal	<Integer>	Read-only	VPX500P Total.

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
vpx500pavailable	<Integer>	Read-only	VPX500P Available.
vpx1000stotal	<Integer>	Read-only	VPX1000S Total.
vpx1000savailable	<Integer>	Read-only	VPX1000S Available.
vpx1000etotal	<Integer>	Read-only	VPX1000E Total.
vpx1000eavailable	<Integer>	Read-only	VPX1000E Available.
vpx1000ptotal	<Integer>	Read-only	VPX1000P Total.
vpx1000pavailable	<Integer>	Read-only	VPX1000P Available.
vpx2000ptotal	<Integer>	Read-only	VPX2000P Total.
vpx2000pavailable	<Integer>	Read-only	VPX2000P Available.
vpx3000stotal	<Integer>	Read-only	VPX3000S Total.
vpx3000savailable	<Integer>	Read-only	VPX3000S Available.
vpx3000etotal	<Integer>	Read-only	VPX3000E Total.
vpx3000eavailable	<Integer>	Read-only	VPX3000E Available.
vpx3000ptotal	<Integer>	Read-only	VPX3000P Total.
vpx3000pavailable	<Integer>	Read-only	VPX3000P Available.
vpx4000ptotal	<Integer>	Read-only	VPX4000P Total.
vpx4000pavailable	<Integer>	Read-only	VPX4000P Available.
vpx5000stotal	<Integer>	Read-only	VPX5000S Total.
vpx5000savailable	<Integer>	Read-only	VPX5000S Available.
vpx5000etotal	<Integer>	Read-only	VPX5000E Total.
vpx5000eavailable	<Integer>	Read-only	VPX5000E Available.

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
vpx5000ptotal	<Integer>	Read-only	VPX5000P Total.
vpx5000pavailable	<Integer>	Read-only	VPX5000P Available.
vpx8000stotal	<Integer>	Read-only	VPX8000S Total.
vpx8000savailable	<Integer>	Read-only	VPX8000S Available.
vpx8000etotal	<Integer>	Read-only	VPX8000E Total.
vpx8000eavailable	<Integer>	Read-only	VPX8000E Available.
vpx8000ptotal	<Integer>	Read-only	VPX8000P Total.
vpx8000pavailable	<Integer>	Read-only	VPX8000P Available.
vpx10000stotal	<Integer>	Read-only	VPX10000S Total.
vpx10000savailable	<Integer>	Read-only	VPX10000S Available.
vpx10000etotal	<Integer>	Read-only	VPX10000E Total.
vpx10000eavailable	<Integer>	Read-only	VPX10000E Available.
vpx10000ptotal	<Integer>	Read-only	VPX10000P Total.
vpx10000pavailable	<Integer>	Read-only	VPX10000P Available.
vpx15000stotal	<Integer>	Read-only	VPX15000S Total.
vpx15000savailable	<Integer>	Read-only	VPX15000S Available.
vpx15000etotal	<Integer>	Read-only	VPX15000E Total.
vpx15000eavailable	<Integer>	Read-only	VPX15000E Available.
vpx15000ptotal	<Integer>	Read-only	VPX15000P Total.

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
vpx15000pavailable	<Integer>	Read-only	VPX15000P Available.
vpx25000stotal	<Integer>	Read-only	VPX25000S Total.
vpx25000savailable	<Integer>	Read-only	VPX25000S Available.
vpx25000etotal	<Integer>	Read-only	VPX25000E Total.
vpx25000eavailable	<Integer>	Read-only	VPX25000E Available.
vpx25000ptotal	<Integer>	Read-only	VPX25000P Total.
vpx25000pavailable	<Integer>	Read-only	VPX25000P Available.
vpx40000stotal	<Integer>	Read-only	VPX40000S Total.
vpx40000savailable	<Integer>	Read-only	VPX40000S Available.
vpx40000etotal	<Integer>	Read-only	VPX40000E Total.
vpx40000eavailable	<Integer>	Read-only	VPX40000E Available.
vpx40000ptotal	<Integer>	Read-only	VPX40000P Total.
vpx40000pavailable	<Integer>	Read-only	VPX40000P Available.
vpx100000stotal	<Integer>	Read-only	VPX100000S Total.
vpx100000savailable	<Integer>	Read-only	VPX100000S Available.
vpx100000etotal	<Integer>	Read-only	VPX100000E Total.
vpx100000eavailable	<Integer>	Read-only	VPX100000E Available.

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
vpx100000ptotal	<Integer>	Read-only	VPX100000P Total.
vpx100000pavailable	<Integer>	Read-only	VPX100000P Available.

Operations

(click to see Properties)

GET (ALL)

Some options that you can use for each operations:

- **Getting warnings in response:**NITRO allows you to get warnings in an operation by specifying the "warning" query parameter as "yes". For example, to get warnings while connecting to the Citrix ADC appliance, the URL is as follows:

  http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/login?warning=yes

  If any, the warnings are displayed in the response payload with the HTTP code "209 X-NITRO-WARNING".

- **Authenticated access for individual NITRO operations:**NITRO allows you to logon to the Citrix ADC appliance to perform individual operations. You can use this option instead of creating a NITRO session (using the login object) and then using that session to perform all operations,

  To do this, you must specify the username and password in the request header of the NITRO request as follows:

  X-NITRO-USER:<username>

  X-NITRO-PASS:<password>

  Note: In such cases, make sure that the request header DOES not include the following:

  Cookie:NITRO_AUTH_TOKEN=<tokenvalue>

`*Note:`*

Mandatory parameters are marked in redand placeholder content is marked in <green>.

get (all)

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/nslicenseserverpool

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "nslicenseserverpool": [ {

"instancetotal":<Integer_value>,

"instanceavailable":<Integer_value>,

"standardbandwidthtotal":<Integer_value>,

"standardbandwidthavailable":<Integer_value>,

"enterprisebandwidthtotal":<Integer_value>,

"enterprisebandwidthavailable":<Integer_value>,

"platinumbandwidthtotal":<Integer_value>,

"platinumbandwidthavailable":<Integer_value>,

"standardcputotal":<Integer_value>,

"standardcpuavailable":<Integer_value>,

"enterprisecputotal":<Integer_value>,
```

```
"enterprisecpuavailable":<Integer_value>,

"platinumcputotal":<Integer_value>,

"platinumcpuavailable":<Integer_value>,

"cpxinstancetotal":<Integer_value>,

"cpxinstanceavailable":<Integer_value>,

"vpx1stotal":<Integer_value>,

"vpx1savailable":<Integer_value>,

"vpx1ptotal":<Integer_value>,

"vpx1pavailable":<Integer_value>,

"vpx5stotal":<Integer_value>,

"vpx5savailable":<Integer_value>,

"vpx5ptotal":<Integer_value>,

"vpx5pavailable":<Integer_value>,

"vpx10stotal":<Integer_value>,

"vpx10savailable":<Integer_value>,

"vpx10etotal":<Integer_value>,

"vpx10eavailable":<Integer_value>,

"vpx10ptotal":<Integer_value>,

"vpx10pavailable":<Integer_value>,

"vpx25stotal":<Integer_value>,

"vpx25savailable":<Integer_value>,

"vpx25etotal":<Integer_value>,
```

```
"vpx25eavailable":<Integer_value>,

"vpx25ptotal":<Integer_value>,

"vpx25pavailable":<Integer_value>,

"vpx50stotal":<Integer_value>,

"vpx50savailable":<Integer_value>,

"vpx50etotal":<Integer_value>,

"vpx50eavailable":<Integer_value>,

"vpx50ptotal":<Integer_value>,

"vpx50pavailable":<Integer_value>,

"vpx100stotal":<Integer_value>,

"vpx100savailable":<Integer_value>,

"vpx100etotal":<Integer_value>,

"vpx100eavailable":<Integer_value>,

"vpx100ptotal":<Integer_value>,

"vpx100pavailable":<Integer_value>,

"vpx200stotal":<Integer_value>,

"vpx200savailable":<Integer_value>,

"vpx200etotal":<Integer_value>,

"vpx200eavailable":<Integer_value>,

"vpx200ptotal":<Integer_value>,

"vpx200pavailable":<Integer_value>,

"vpx500stotal":<Integer_value>,
```

```
"vpx500savailable":<Integer_value>,

"vpx500etotal":<Integer_value>,

"vpx500eavailable":<Integer_value>,

"vpx500ptotal":<Integer_value>,

"vpx500pavailable":<Integer_value>,

"vpx1000stotal":<Integer_value>,

"vpx1000savailable":<Integer_value>,

"vpx1000etotal":<Integer_value>,

"vpx1000eavailable":<Integer_value>,

"vpx1000ptotal":<Integer_value>,

"vpx1000pavailable":<Integer_value>,

"vpx2000ptotal":<Integer_value>,

"vpx2000pavailable":<Integer_value>,

"vpx3000stotal":<Integer_value>,

"vpx3000savailable":<Integer_value>,

"vpx3000etotal":<Integer_value>,

"vpx3000eavailable":<Integer_value>,

"vpx3000ptotal":<Integer_value>,

"vpx3000pavailable":<Integer_value>,

"vpx4000ptotal":<Integer_value>,

"vpx4000pavailable":<Integer_value>,

"vpx5000stotal":<Integer_value>,
```

```
"vpx5000savailable":<Integer_value>,

"vpx5000etotal":<Integer_value>,

"vpx5000eavailable":<Integer_value>,

"vpx5000ptotal":<Integer_value>,

"vpx5000pavailable":<Integer_value>,

"vpx8000stotal":<Integer_value>,

"vpx8000savailable":<Integer_value>,

"vpx8000etotal":<Integer_value>,

"vpx8000eavailable":<Integer_value>,

"vpx8000ptotal":<Integer_value>,

"vpx8000pavailable":<Integer_value>,

"vpx10000stotal":<Integer_value>,

"vpx10000savailable":<Integer_value>,

"vpx10000etotal":<Integer_value>,

"vpx10000eavailable":<Integer_value>,

"vpx10000ptotal":<Integer_value>,

"vpx10000pavailable":<Integer_value>,

"vpx15000stotal":<Integer_value>,

"vpx15000savailable":<Integer_value>,

"vpx15000etotal":<Integer_value>,

"vpx15000eavailable":<Integer_value>,

"vpx15000ptotal":<Integer_value>,
```

```
"vpx15000pavailable":<Integer_value>,

"vpx25000stotal":<Integer_value>,

"vpx25000savailable":<Integer_value>,

"vpx25000etotal":<Integer_value>,

"vpx25000eavailable":<Integer_value>,

"vpx25000ptotal":<Integer_value>,

"vpx25000pavailable":<Integer_value>,

"vpx40000stotal":<Integer_value>,

"vpx40000savailable":<Integer_value>,

"vpx40000etotal":<Integer_value>,

"vpx40000eavailable":<Integer_value>,

"vpx40000ptotal":<Integer_value>,

"vpx40000pavailable":<Integer_value>,

"vpx100000stotal":<Integer_value>,

"vpx100000savailable":<Integer_value>,

"vpx100000etotal":<Integer_value>,

"vpx100000eavailable":<Integer_value>,

"vpx100000ptotal":<Integer_value>,

"vpx100000pavailable":<Integer_value>

}]}
```

nslimitidentifier

Configuration for limit Indetifier resource.

Properties

(click to see Operations)

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
limitidentifier	<String>	Read-write	Name for a rate limit identifier. Must begin with an ASCII letter or underscore (_) character, and must consist only of ASCII alphanumeric or underscore characters. Reserved words must not be used.
threshold	<Double>	Read-write	Maximum number of requests that are allowed in the given timeslice when requests (mode is set as REQUEST_RATE) are tracked per timeslice. When connections (mode is set as CONNECTION) are tracked, it is the total number of connections that would be let through. Default value: 1 Minimum value = 1

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
timeslice	<Double>	Read-write	Time interval, in milliseconds, specified in multiples of 10, during which requests are tracked to check if they cross the threshold. This argument is needed only when the mode is set to REQUEST_RATE. Default value: 1000 Minimum value = 10
mode	<String>	Read-write	Defines the type of traffic to be tracked. * REQUEST_RATE - Tracks requests/ timeslice. * CONNECTION - Tracks active transactions. Examples 1. To permit 20 requests in 10 ms and 2 traps in 10 ms: add limitidentifier limit_req -mode request_rate -limitType smooth -timeslice 1000 -Threshold 2000

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
			-trapsInTimeSlice 200 2. To permit 50 requests in 10 ms: set limitidentifier limit_req -mode request_rate -timeslice 1000 -Threshold 5000 -limitType smooth 3. To permit 1 request in 40 ms: set limitidentifier limit_req -mode request_rate -timeslice 2000 -Threshold 50 -limitType smooth 4. To permit 1 request in 200 ms and 1 trap in 130 ms: set limitidentifier limit_req -mode request_rate -timeslice 1000 -Threshold 5 -limitType smooth -trapsInTimeSlice 8 5. To permit 5000 requests in 1000 ms

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
			and 200 traps in 1000 ms: set limitidentifier limit_req -mode request_rate -timeslice 1000 -Threshold 5000 -limitType BURSTY. Default value: REQUEST_RATE Possible values = CONNECTION, REQUEST_RATE, NONE
limittype	<String>	Read-write	Smooth or bursty request type. * SMOOTH - When you want the permitted number of requests in a given interval of time to be spread evenly across the timeslice * BURSTY - When you want the permitted number of requests to exhaust the quota anytime within the timeslice. This argument is needed only when

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
			the mode is set to REQUEST_RATE. Default value: BURSTY Possible values = BURSTY, SMOOTH
selectorname	<String>	Read-write	Name of the rate limit selector. If this argument is NULL, rate limiting will be applied on all traffic received by the virtual server or the Citrix ADC (depending on whether the limit identifier is bound to a virtual server or globally) without any filtering. Minimum length = 1
maxbandwidth	<Double>	Read-write	Maximum bandwidth permitted, in kbps. Minimum value = 0 Maximum value = 4294967287
trapsintimeslice	<Double>	Read-write	Number of traps to be sent in the timeslice configured. A value of 0 indicates that traps are disabled. Minimum value = 0

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
			Maximum value = 65535
ngname	<String>	Read-only	Nodegroup name to which this identifier belongs to.
hits	<Double>	Read-only	The number of times this identifier was evaluated.
drop	<Double>	Read-only	The number of times action was taken.
rule	<String[]>	Read-only	Rule.
time	<Double>	Read-only	Time interval considered for rate limiting.
total	<Double>	Read-only	Maximum number of requests permitted in the computed timeslice.
trapscomputedintimeslice	<Double>	Read-only	The number of traps that would be sent in the timeslice configured. .
computedtraptimeslice	<Double>	Read-only	The time interval computed for sending traps.
referencecount	<Double>	Read-only	Total number of transactions pointing to this entry.
__count	<Double>	Read-only	count parameter

Operations

(click to see Properties)

ADD| DELETE| UPDATE| UNSET| GET (ALL)| GET| COUNT

Some options that you can use for each operations:

- **Getting warnings in response:**NITRO allows you to get warnings in an operation by specifying the "warning" query parameter as "yes". For example, to get warnings while connecting to the Citrix ADC appliance, the URL is as follows:

  http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/login?warning=yes

  If any, the warnings are displayed in the response payload with the HTTP code "209 X-NITRO-WARNING".

- **Authenticated access for individual NITRO operations:**NITRO allows you to logon to the Citrix ADC appliance to perform individual operations. You can use this option instead of creating a NITRO session (using the login object) and then using that session to perform all operations,

  To do this, you must specify the username and password in the request header of the NITRO request as follows:

  X-NITRO-USER:<username>

  X-NITRO-PASS:<password>

  Note: In such cases, make sure that the request header DOES not include the following:

  Cookie:NITRO_AUTH_TOKEN=<tokenvalue>

`*Note:`*

Mandatory parameters are marked in redand placeholder content is marked in <green>.

add

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/nslimitidentifier

HTTP Method:POST

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Content-Type:application/json

Request Payload:

```
{"nslimitidentifier":{

<b>"limitidentifier":<String_value>,

</b>"threshold":<Double_value>,

"timeslice":<Double_value>,

"mode":<String_value>,

"limittype":<String_value>,

"selectorname":<String_value>,

"maxbandwidth":<Double_value>,

"trapsintimeslice":<Double_value>

}}
```

Response:

HTTP Status Code on Success: 201 Created HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error

delete

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/nslimitidentifier/limitidentifier_value<String>

HTTP Method:DELETE

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue>

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error

update

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/nslimitidentifier

HTTP Method:PUT

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Content-Type:application/json

Request Payload:

```
{"nslimitidentifier":{

<b>"limitidentifier":<String_value>,

</b>"threshold":<Double_value>,

"timeslice":<Double_value>,

"mode":<String_value>,

"limittype":<String_value>,

"selectorname":<String_value>,

"maxbandwidth":<Double_value>,

"trapsintimeslice":<Double_value>

}}
```

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error

unset

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/nslimitidentifier?**action=unset**

HTTP Method:POST

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Content-Type:application/json

Request Payload:

```
{"nslimitidentifier":{

<b>"limitidentifier":<String_value>,

</b>"selectorname":true,

"threshold":true,

"timeslice":true,

"mode":true,

"limittype":true,

"maxbandwidth":true,

"trapsintimeslice":true

}}
```

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error

get (all)

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/nslimitidentifier

Query-parameters:

attrs

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/nslimitidentifier?**attrs=property-name1,property-name2**

Use this query parameter to specify the resource details that you want to retrieve.

filter

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/nslimitidentifier?**filter=property-name1:property-val1,property-name2:property-val2**

Use this query-parameter to get the filtered set of nslimitidentifier resources configured on Citrix ADC. Filtering can be done on any of the properties of the resource.

view

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/nslimitidentifier?**view=summary**

Note: By default, the retrieved results are displayed in detail view (?view=detail).

pagination

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/nslimitidentifier?**pagesize=#no;pageno=#no**

Use this query-parameter to get the nslimitidentifier resources in chunks.

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "nslimitidentifier": [ {
"limitidentifier":<String_value>,
"ngname":<String_value>,
"threshold":<Double_value>,
"timeslice":<Double_value>,
"mode":<String_value>,
"limittype":<String_value>,
"selectorname":<String_value>,
"hits":<Double_value>,
"drop":<Double_value>,
"rule":<String[]_value>,
"time":<Double_value>,
"total":<Double_value>,
"maxbandwidth":<Double_value>,
"trapsintimeslice":<Double_value>,
"trapscomputedintimeslice":<Double_value>,
"computedtraptimeslice":<Double_value>,
"referencecount":<Double_value>
}]}
```

get

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/nslimitidentifier/limitidentifier_value<String>

Query-parameters:

attrs

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/nslimitidentifier/limitidentifier_value<String>?**attrs=property-name1,property-name2**

Use this query parameter to specify the resource details that you want to retrieve.

view

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/nslimitidentifier/limitidentifier_value<String>?**view=summary**

Note: By default, the retrieved results are displayed in detail view (?view=detail).

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "nslimitidentifier": [ {

"limitidentifier":<String_value>,

"ngname":<String_value>,
```

```
"threshold":<Double_value>,

"timeslice":<Double_value>,

"mode":<String_value>,

"limittype":<String_value>,

"selectorname":<String_value>,

"hits":<Double_value>,

"drop":<Double_value>,

"rule":<String[]_value>,

"time":<Double_value>,

"total":<Double_value>,

"maxbandwidth":<Double_value>,

"trapsintimeslice":<Double_value>,

"trapscomputedintimeslice":<Double_value>,

"computedtraptimeslice":<Double_value>,

"referencecount":<Double_value>

}]}
```

count

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/nslimitidentifier?**count=yes**

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "nslimitidentifier": [ { "__count": "#no"} ] }
```

nslimitidentifier_binding

Binding object which returns the resources bound to nslimitidentifier.

Properties

(click to see Operations)

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
limitidentifier	<String>	Read-write	Name of the rate limit identifier about which to display information. If a name is not provided, information about all rate limit identifiers is shown.
nslimitidentifier_nslimitsessions_binding	<nslimitidentifier_nslimitsessions_binding[]>	Read-only	nslimitsessions that can be bound to nslimitidentifier.

Operations

(click to see Properties)

GET| GET (ALL)

Some options that you can use for each operations:

- **Getting warnings in response:**NITRO allows you to get warnings in an operation by specifying the "warning" query parameter as "yes". For example, to get warnings while connecting to the Citrix ADC appliance, the URL is as follows:

  http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/login?warning=yes

  If any, the warnings are displayed in the response payload with the HTTP code "209 X-NITRO-WARNING".

- **Authenticated access for individual NITRO operations:**NITRO allows you to logon to the Citrix ADC appliance to perform individual operations. You can use this option instead of creating a NITRO session (using the login object) and then using that session to perform all operations,

  To do this, you must specify the username and password in the request header of the NITRO request as follows:

  X-NITRO-USER:<username>

  X-NITRO-PASS:<password>

  Note: In such cases, make sure that the request header DOES not include the following:

  Cookie:NITRO_AUTH_TOKEN=<tokenvalue>

`*Note:`*

Mandatory parameters are marked in redand placeholder content is marked in <green>.

get

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/nslimitidentifier_binding/limitidentifier_value<String>

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "nslimitidentifier_binding": [ {

"limitidentifier":<String_value>,

"nslimitidentifier_nslimitsessions_binding":<nslimitidentifier_nslimitsess
ions_binding[]_value>

}]}
```

get (all)

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/nslimitidentifier_binding

Query-parameters:

bulkbindings

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/nslimitidentifier_binding?**bulkbindings=yes**

NITRO allows you to fetch bindings in bulk.

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "nslimitidentifier_binding": [ {

"limitidentifier":<String_value>,

"nslimitidentifier_nslimitsessions_binding":<nslimitidentifier_nslimitsessions_binding[]_value>

}]}
```

nslimitidentifier_nslimitsessions_binding

Binding object showing the nslimitsessions that can be bound to nslimitidentifier.

Properties

(click to see Operations)

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
limitidentifier	<String>	Read-write	Name of the rate limit identifier about which to display information. If a name is not provided, information about all rate limit identifiers is shown.
__count	<Double>	Read-write	count parameter

Operations

(click to see Properties)

GET| GET (ALL)| COUNT

Some options that you can use for each operations:

- **Getting warnings in response:**NITRO allows you to get warnings in an operation by specifying the "warning" query parameter as "yes". For example, to get warnings while connecting to the Citrix ADC appliance, the URL is as follows:

  http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/login?warning=yes

  If any, the warnings are displayed in the response payload with the HTTP code "209 X-NITRO-WARNING".

- **Authenticated access for individual NITRO operations:**NITRO allows you to logon to the Citrix ADC appliance to perform individual operations. You can use this option instead of creating a NITRO session (using the login object) and then using that session to perform all operations,

  To do this, you must specify the username and password in the request header of the NITRO request as follows:

  X-NITRO-USER:<username>

  X-NITRO-PASS:<password>

  Note: In such cases, make sure that the request header DOES not include the following:

  Cookie:NITRO_AUTH_TOKEN=<tokenvalue>

`*Note:`*

Mandatory parameters are marked in redand placeholder content is marked in <green>.

get

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/nslimitidentifier_nslimitsessions_binding/limitidentifier_value<String>

Query-parameters:

filter

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/nslimitidentifier_nslimitsessions_binding/limitidentifier_value<String>?**filter=property-name1:property-value1,property-name2:property-value2**

Use this query-parameter to get the filtered set of nslimitidentifier_nslimitsessions_binding resources configured on Citrix ADC. Filtering can be done on any of the properties of the resource.

pagination

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/nslimitidentifier_nslimitsessions_binding/limitidentifier_value<String>?**pagesize=#no;pageno=#no**

Use this query-parameter to get the nslimitidentifier_nslimitsessions_binding resources in chunks.

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "nslimitidentifier_nslimitsessions_binding": [ {

"limitidentifier":<String_value>

}]}
```

get (all)

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/nslimitidentifier_nslimitsessions_binding

Query-parameters:

bulkbindings

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/nslimitidentifier_nslimitsessions_binding?**bulkbindings=yes**

NITRO allows you to fetch bindings in bulk.

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "nslimitidentifier_nslimitsessions_binding": [ {

"limitidentifier":<String_value>

}]}
```

count

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/nslimitidentifier_nslimitsessions_binding/limitidentifier_value<String>?**count=yes**

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{"nslimitidentifier_nslimitsessions_binding": [ { "__count": "#no"} ] }
```


nslimitselector

Configuration for limit selector resource.

Properties

(click to see Operations)

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
selectorname	<String>	Read-write	.
rule	<String[]>	Read-write	. Minimum length = 1
__count	<Double>	Read-only	count parameter

Operations

(click to see Properties)

ADD| DELETE| UPDATE| UNSET| GET (ALL)| GET| COUNT

Some options that you can use for each operations:

- **Getting warnings in response:**NITRO allows you to get warnings in an operation by specifying the "warning" query parameter as "yes". For example, to get warnings while connecting to the Citrix ADC appliance, the URL is as follows:

  http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/login?warning=yes

  If any, the warnings are displayed in the response payload with the HTTP code "209 X-NITRO-WARNING".
- **Authenticated access for individual NITRO operations:**NITRO allows you to logon to the Citrix ADC appliance to perform individual operations. You can use this option instead of creating a NITRO session (using the login object) and then using that session to perform all operations,

To do this, you must specify the username and password in the request header of the NITRO request as follows:

X-NITRO-USER:<username>

X-NITRO-PASS:<password>

Note: In such cases, make sure that the request header DOES not include the following:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue>

`*Note:`*

Mandatory parameters are marked in redand placeholder content is marked in <green>.

add

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/nslimitselector

HTTP Method:POST

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Content-Type:application/json

Request Payload:

```
{"nslimitselector":{

<b>"selectorname":<String_value>,

</b><b>"rule":<String[]_value>

</b>}}
```

Response:

HTTP Status Code on Success: 201 Created HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error

delete

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/nslimitselector/selectorname_value<String>

HTTP Method:DELETE

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue>

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error

update

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/nslimitselector

HTTP Method:PUT

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Content-Type:application/json

Request Payload:

```
{"nslimitselector":{

<b>"selectorname":<String_value>,

</b>"rule":<String[]_value>

}}
```

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error

unset

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/nslimitselector?**action=unset**

HTTP Method:POST

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Content-Type:application/json

Request Payload:

```
{"nslimitselector":{

<b>"selectorname":<String_value>,

</b>"rule":true

}}
```

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error

get (all)

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/nslimitselector

Query-parameters:

attrs

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/nslimitselector?**attrs=property-name1,property-name2**

Use this query parameter to specify the resource details that you want to retrieve.

filter

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/nslimitselector?**filter=property-name1:property-val1,property-name2:property-val2**

Use this query-parameter to get the filtered set of nslimitselector resources configured on Citrix ADC. Filtering can be done on any of the properties of the resource.

view

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/nslimitselector?**view=summary**

Note: By default, the retrieved results are displayed in detail view (?view=detail).

pagination

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/nslimitselector?**pagesize=#no;pageno=#no**

Use this query-parameter to get the nslimitselector resources in chunks.

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "nslimitselector": [ {

"selectorname":<String_value>,

"rule":<String[]_value>

}]}
```

get

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/nslimitselector/selectorname_value<String>

Query-parameters:

attrs

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/nslimitselector/selectorname_value<String>?**attrs=property-name1,property-name2**

Use this query parameter to specify the resource details that you want to retrieve.

view

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/nslimitselector/selectorname_value<String>?**view=summary**

Note: By default, the retrieved results are displayed in detail view (?view=detail).

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "nslimitselector": [ {

"selectorname":<String_value>,

"rule":<String[]_value>
```

```
}]}
```

count

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/nslimitselector?**count=yes**

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "nslimitselector": [ { "__count": "#no"} ] }
```

nslimitsessions

Configuration for limit sessions resource.

Properties

(click to see Operations)

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
limitidentifier	<String>	Read-write	Name of the rate limit identifier for which to display the sessions. Minimum length = 1

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
detail	<Boolean>	Read-write	Show the individual hash values.
timeout	<Double>	Read-only	The time remaining on the session before a flush can be attempted. If active transactions are present the session will not be flushed.
hits	<Double>	Read-only	The number of times this entry was hit.
drop	<Double>	Read-only	The number of times action was taken.
number	<Double[]>	Read-only	The hash of the matched selectlets.
name	<String>	Read-only	The string formed by gathering selectlet values.
unit	<Double>	Read-only	Total computed hash of the matched selectlets.
flags	<Double>	Read-only	Used internally to identify ip addresses.
referencecount	<Double>	Read-only	Total number of transactions pointing to this entry. Its the sum total of the connection and bandwidth references.
maxbandwidth	<Double>	Read-only	The current bandwidth.

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
selectoripv61	<String>	Read-only	First IPV6 address gathered.
selectoripv62	<String>	Read-only	Second IPV6 address gathered.
flag	<Double>	Read-only	Used internally to identify ipv6 addresses.
__count	<Double>	Read-only	count parameter

Operations

(click to see Properties)

CLEAR| GET (ALL)| COUNT

Some options that you can use for each operations:

- **Getting warnings in response:**NITRO allows you to get warnings in an operation by specifying the "warning" query parameter as "yes". For example, to get warnings while connecting to the Citrix ADC appliance, the URL is as follows:

  http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/login?warning=yes

  If any, the warnings are displayed in the response payload with the HTTP code "209 X-NITRO-WARNING".

- **Authenticated access for individual NITRO operations:**NITRO allows you to logon to the Citrix ADC appliance to perform individual operations. You can use this option instead of creating a NITRO session (using the login object) and then using that session to perform all operations,

  To do this, you must specify the username and password in the request header of the NITRO request as follows:

  X-NITRO-USER:<username>

  X-NITRO-PASS:<password>

  Note: In such cases, make sure that the request header DOES not include the following:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue>

`*Note:` *

Mandatory parameters are marked in redand placeholder content is marked in <green>.

clear

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/nslimitsessions?**action=clear**

HTTP Method:POST

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Content-Type:application/json

Request Payload:

```
{"nslimitsessions":{

<b>"limitidentifier":<String_value>

</b>}}
```

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error

get (all)

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/nslimitsessions

Query-parameters:

args

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/nslimitsessions?**args=limitidentifier:<String_value>,detail:<Boolean_value>**

Use this query-parameter to get nslimitsessions resources based on additional properties.

attrs

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/nslimitsessions?**attrs=property-name1,property-name2**

Use this query parameter to specify the resource details that you want to retrieve.

filter

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/nslimitsessions?**filter=property-name1:property-val1,property-name2:property-val2**

Use this query-parameter to get the filtered set of nslimitsessions resources configured on Citrix ADC. Filtering can be done on any of the properties of the resource.

view

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/nslimitsessions?**view=summary**

Note: By default, the retrieved results are displayed in detail view (?view=detail).

pagination

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/nslimitsessions?**pagesize=#no;pageno=#no**

Use this query-parameter to get the nslimitsessions resources in chunks.

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "nslimitsessions": [ {

<b>limitidentifier:<String_value>,</
b>detail:<Boolean_value>"timeout":<Double_value>,

"hits":<Double_value>,

"drop":<Double_value>,

"number":<Double[]_value>,

"name":<String_value>,

"unit":<Double_value>,

"flags":<Double_value>,

"referencecount":<Double_value>,

"maxbandwidth":<Double_value>,

"selectoripv61":<String_value>,

"selectoripv62":<String_value>,

"flag":<Double_value>

}]}
```

count

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/nslimitsessions?**args=limitidentifier:<String_value>,detail:<Boolean_value>;count=yes**

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "nslimitsessions": [ { "__count": "#no"} ] }
```

nsmode

Configuration for ns mode resource.

Properties

(click to see Operations)

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
mode	<String[]>	Read-write	Mode to be enabled. Multiple modes can be specified by providing a blank space between each mode. Possible values = FR, FastRamp, L2, L2mode, L3, L3mode, USIP, UseSourceIP, CKA, ClientKeepAlive, TCPB, TCPBuffering, MBF, MACbasedforwarding, Edge, USNIP, SRADV, DRADV, IRADV, SRADV6,

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
			DRADV6, PMTUD, RISE_APBR, RISE_RHI, BridgeBPDUs, ULFD
fr	<Boolean>	Read-only	Fast Ramp.
l2	<Boolean>	Read-only	Layer 2 mode.
usip	<Boolean>	Read-only	Use Source IP.
cka	<Boolean>	Read-only	Client Keep-alive.
tcpb	<Boolean>	Read-only	TCP Buffering.
mbf	<Boolean>	Read-only	MAC-based forwarding.
edge	<Boolean>	Read-only	Edge configuration.
usnip	<Boolean>	Read-only	Use Subnet IP.
l3	<Boolean>	Read-only	Layer 3 mode (ip forwarding).
pmtud	<Boolean>	Read-only	Path MTU Discovery.
mediaclassification	<Boolean>	Read-only	mediaclassification.
sradv	<Boolean>	Read-only	Static Route Advertisement.
dradv	<Boolean>	Read-only	Direct Route Advertisement.
iradv	<Boolean>	Read-only	Intranet Route Advertisement.
sradv6	<Boolean>	Read-only	Ipv6 Static Route Advertisement.

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
dradv6	<Boolean>	Read-only	Ipv6 Direct Route Advertisement.
bridgebpdus	<Boolean>	Read-only	BPDU Bridging Mode.
rise_apbr	<Boolean>	Read-only	APBR Publishing Mode.
rise_rhi	<Boolean>	Read-only	RHI Publishing Mode.
ulfd	<Boolean>	Read-only	Unified Logging Framework Mode for adding/removing ULF services.

Operations

(click to see Properties)

ENABLE| DISABLE| GET (ALL)

Some options that you can use for each operations:

- **Getting warnings in response:**NITRO allows you to get warnings in an operation by specifying the "warning" query parameter as "yes". For example, to get warnings while connecting to the Citrix ADC appliance, the URL is as follows:

  http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/login?warning=yes

  If any, the warnings are displayed in the response payload with the HTTP code "209 X-NITRO-WARNING".

- **Authenticated access for individual NITRO operations:**NITRO allows you to logon to the Citrix ADC appliance to perform individual operations. You can use this option instead of creating a NITRO session (using the login object) and then using that session to perform all operations,

  To do this, you must specify the username and password in the request header of the NITRO request as follows:

X-NITRO-USER:<username>

X-NITRO-PASS:<password>

Note: In such cases, make sure that the request header DOES not include the following:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue>

`*Note:`*

Mandatory parameters are marked in redand placeholder content is marked in <green>.

enable

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/nsmode?**action=enable**

HTTP Method:POST

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Content-Type:application/json

Request Payload:

```
{"nsmode":{

"mode":<String[]_value>

}}
```

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error

disable

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/nsmode?**action=disable**

HTTP Method:POST

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Content-Type:application/json

Request Payload:

```
{"nsmode":{

"mode":<String[]_value>

}}
```

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error

get (all)

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/nsmode

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "nsmode": [ {

"mode":<String[]_value>,
```

```
"fr":<Boolean_value>,

"l2":<Boolean_value>,

"usip":<Boolean_value>,

"cka":<Boolean_value>,

"tcpb":<Boolean_value>,

"mbf":<Boolean_value>,

"edge":<Boolean_value>,

"usnip":<Boolean_value>,

"l3":<Boolean_value>,

"pmtud":<Boolean_value>,

"mediaclassification":<Boolean_value>,

"sradv":<Boolean_value>,

"dradv":<Boolean_value>,

"iradv":<Boolean_value>,

"sradv6":<Boolean_value>,

"dradv6":<Boolean_value>,

"bridgebpdus":<Boolean_value>,

"rise_apbr":<Boolean_value>,

"rise_rhi":<Boolean_value>,

"ulfd":<Boolean_value>

}]}
```

nsparam

Configuration for Citrix ADC parameters resource.

Properties

(click to see Operations)

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
httpport	<Integer[]>	Read-write	HTTP ports on the web server. This allows the system to perform connection off-load for any client request that has a destination port matching one of these configured ports. Minimum value = 1 Maximum value = 65535
maxconn	<Double>	Read-write	Maximum number of connections that will be made from the appliance to the web server(s) attached to it. The value entered here is applied globally to all attached servers. Default value: 0 Minimum value = 0 Maximum value = 4294967294

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
maxreq	<Double>	Read-write	Maximum number of requests that the system can pass on a particular connection between the appliance and a server attached to it. Setting this value to 0 allows an unlimited number of requests to be passed. This value is overridden by the maximum number of requests configured on the individual service. Minimum value = 0 Maximum value = 65535
cip	<String>	Read-write	Enable or disable the insertion of the actual client IP address into the HTTP header request passed from the client to one, some, or all servers attached to the system. The passed address can then be accessed through a minor modification to the server. * If the CIP header is specified, it will be used as the client IP header.

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
			* If the CIP header is not specified, the value that has been set will be used as the client IP header. Possible values = ENABLED, DISABLED
cipheader	<String>	Read-write	Text that will be used as the client IP address header. Minimum length = 1
cookieversion	<String>	Read-write	Version of the cookie inserted by the system. Possible values = 0, 1
securecookie	<String>	Read-write	Enable or disable secure flag for persistence cookie. Default value: ENABLED Possible values = ENABLED, DISABLED
pmtumin	<Double>	Read-write	Minimum path MTU value that Citrix ADC will process in the ICMP fragmentation needed message. If the ICMP message contains a value less than this value, then

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
			this value is used instead. Default value: 576 Minimum value = 168 Maximum value = 1500
pmtutimeout	<Double>	Read-write	Interval, in minutes, for flushing the PMTU entries. Default value: 10 Minimum value = 1 Maximum value = 1440
ftpportrange	<String>	Read-write	Minimum and maximum port (port range) that FTP services are allowed to use. Minimum length = 1024 Maximum length = 64000
crportrange	<String>	Read-write	Port range for cache redirection services. Minimum length = 1 Maximum length = 65535
timezone	<String>	Read-write	Time zone for the Citrix ADC. Name of the time zone should

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
			be specified as argument. Default value: CoordinatedUniversalTime Minimum length = 1 Maximum length = 63
grantquotamaxclient	<Double>	Read-write	Percentage of shared quota to be granted at a time for maxClient. Default value: 10 Minimum value = 0 Maximum value = 100
exclusivequotamaxclient	<Double>	Read-write	Percentage of maxClient to be given to PEs. Default value: 80 Minimum value = 0 Maximum value = 100
grantquotaspillover	<Double>	Read-write	Percentage of shared quota to be granted at a time for spillover. Default value: 10 Minimum value = 0

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
			Maximum value = 100
exclusivequotaspillover	<Double>	Read-write	Percentage of maximum limit to be given to PEs. Default value: 80 Minimum value = 0 Maximum value = 100
useproxyport	<String>	Read-write	Enable/Disable use_proxy_port setting. Default value: ENABLED Possible values = ENABLED, DISABLED
internaluserlogin	<String>	Read-write	Enables/disables the internal user from logging in to the appliance. Before disabling internal user login, you must have key-based authentication set up on the appliance. The file name for the key pair must be "ns_comm_key". Default value: ENABLED

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
			Possible values = ENABLED, DISABLED
aftpallowrandomsourceport	<String>	Read-write	Allow the FTP server to come from a random source port for active FTP data connections. Default value: DISABLED Possible values = ENABLED, DISABLED
icaports	<Integer[]>	Read-write	The ICA ports on the Web server. This allows the system to perform connection off-load for any client request that has a destination port matching one of these configured ports. Minimum value = 1
tcpcip	<String>	Read-write	Enable or disable the insertion of the client TCP/IP header in TCP payload passed from the client to one, some, or all servers attached to the system. The passed address can then be accessed through a minor

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
			modification to the server. Default value: DISABLED Possible values = ENABLED, DISABLED
servicepathingressvlan	<Double>	Read-write	VLAN on which the subscriber traffic arrives on the appliance. Minimum value = 1
secureicaports	<Integer[]>	Read-write	The Secure ICA ports on the Web server. This allows the system to perform connection off-load for any client request that has a destination port matching one of these configured ports. Minimum value = 1
mgmthttpport	<Integer>	Read-write	This allow the configuration of management HTTP port. Default value: 80 Minimum value = 1 Maximum value = 65534

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
mgmthttpsport	<Integer>	Read-write	This allows the configuration of management HTTPS port. Default value: 443 Minimum value = 1 Maximum value = 65534
proxyprotocol	<String>	Read-write	Disable/Enable v1 or v2 proxy protocol header for client info insertion. Default value: DISABLED Possible values = ENABLED, DISABLED

Operations

(click to see Properties)

UPDATE| UNSET| GET (ALL)

Some options that you can use for each operations:

- **Getting warnings in response:**NITRO allows you to get warnings in an operation by specifying the "warning" query parameter as "yes". For example, to get warnings while connecting to the Citrix ADC appliance, the URL is as follows:

  http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/login?warning=yes

  If any, the warnings are displayed in the response payload with the HTTP code "209 X-NITRO-WARNING".

- **Authenticated access for individual NITRO operations:**NITRO allows you to logon to the Citrix ADC appliance to perform individual operations. You can use this option instead of

creating a NITRO session (using the login object) and then using that session to perform all operations,

To do this, you must specify the username and password in the request header of the NITRO request as follows:

X-NITRO-USER:<username>

X-NITRO-PASS:<password>

Note: In such cases, make sure that the request header DOES not include the following:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue>

`*Note:`*

Mandatory parameters are marked in redand placeholder content is marked in <green>.

update

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/nsparam

HTTP Method:PUT

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Content-Type:application/json

Request Payload:

```
{"nsparam":{

"httpport":<Integer[]_value>,

"maxconn":<Double_value>,

"maxreq":<Double_value>,

"cip":<String_value>,

"cipheader":<String_value>,

"cookieversion":<String_value>,
```

```
"securecookie":<String_value>,

"pmtumin":<Double_value>,

"pmtutimeout":<Double_value>,

"ftpportrange":<String_value>,

"crportrange":<String_value>,

"timezone":<String_value>,

"grantquotamaxclient":<Double_value>,

"exclusivequotamaxclient":<Double_value>,

"grantquotaspillover":<Double_value>,

"exclusivequotaspillover":<Double_value>,

"useproxyport":<String_value>,

"internaluserlogin":<String_value>,

"aftpallowrandomsourceport":<String_value>,

"icaports":<Integer[]_value>,

"tcpcip":<String_value>,

"servicepathingressvlan":<Double_value>,

"secureicaports":<Integer[]_value>,

"mgmthttpport":<Integer_value>,

"mgmthttpsport":<Integer_value>,

"proxyprotocol":<String_value>

}}
```

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error

unset

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/nsparam?**action=unset**

HTTP Method:POST

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Content-Type:application/json

Request Payload:

```
{"nsparam":{

"ftpportrange":true,

"crportrange":true,

"aftpallowrandomsourceport":true,

"httpport":true,

"maxconn":true,

"maxreq":true,

"cip":true,

"cipheader":true,

"cookieversion":true,

"securecookie":true,
```

```
"pmtumin":true,

"pmtutimeout":true,

"timezone":true,

"grantquotamaxclient":true,

"exclusivequotamaxclient":true,

"grantquotaspillover":true,

"exclusivequotaspillover":true,

"useproxyport":true,

"internaluserlogin":true,

"icaports":true,

"tcpcip":true,

"servicepathingressvlan":true,

"secureicaports":true,

"mgmthttpport":true,

"mgmthttpsport":true,

"proxyprotocol":true

}}
```

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error

get (all)

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/nsparam

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "nsparam": [ {

"httpport":<Integer[]_value>,

"maxconn":<Double_value>,

"maxreq":<Double_value>,

"cip":<String_value>,

"cipheader":<String_value>,

"cookieversion":<String_value>,

"securecookie":<String_value>,

"pmtumin":<Double_value>,

"pmtutimeout":<Double_value>,

"ftpportrange":<String_value>,

"crportrange":<String_value>,
```

```
"timezone":<String_value>,

"grantquotamaxclient":<Double_value>,

"exclusivequotamaxclient":<Double_value>,

"grantquotaspillover":<Double_value>,

"exclusivequotaspillover":<Double_value>,

"useproxyport":<String_value>,

"internaluserlogin":<String_value>,

"aftpallowrandomsourceport":<String_value>,

"icaports":<Integer[]_value>,

"tcpcip":<String_value>,

"servicepathingressvlan":<Double_value>,

"secureicaports":<Integer[]_value>,

"mgmthttpport":<Integer_value>,

"mgmthttpsport":<Integer_value>,

"proxyprotocol":<String_value>

}]}
```

nspartition

Configuration for admin partition resource.

Properties

(click to see Operations)

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
partitionname	<String>	Read-write	Name of the Partition. Must begin with an ASCII alphanumeric or underscore (_) character, and must contain only ASCII alphanumeric, underscore, hash (#), period (.), space, colon (:), at (@), equals (=), and hyphen (-) characters. Minimum length = 1
maxbandwidth	<Double>	Read-write	Maximum bandwidth, in Kbps, that the partition can consume. A zero value indicates the bandwidth is unrestricted on the partition and it can consume up to the system limits. Default value: 10240
minbandwidth	<Double>	Read-write	Minimum bandwidth, in Kbps, that the partition can consume. A zero value indicates the bandwidth is unrestricted on the partition and it can consume up to the system limits.

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
			Default value: 10240
maxconn	<Double>	Read-write	Maximum number of concurrent connections that can be open in the partition. A zero value indicates no limit on number of open connections. Default value: 1024
maxmemlimit	<Double>	Read-write	Maximum memory, in megabytes, allocated to the partition. A zero value indicates the memory is unlimited on the partition and it can consume up to the system limits. Default value: 10 Minimum value = 0 Maximum value = 1048576
partitionmac	<String>	Read-write	Special MAC address for the partition which is used for communication over shared vlans in this partition. If not specified, the MAC address is auto-generated.
partitionid	<Double>	Read-only	Partition Id.

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
			Minimum value = 1
partitiontype	<String>	Read-only	Type of the Partition. Possible values = Default Partition, Current Partition
pmacinternal	<Boolean>	Read-only	Partition MAC is generated internally.
__count	<Double>	Read-only	count parameter

Operations

(click to see Properties)

ADD| DELETE| UPDATE| UNSET| SWITCH| GET (ALL)| GET| COUNT

Some options that you can use for each operations:

- **Getting warnings in response:**NITRO allows you to get warnings in an operation by specifying the "warning" query parameter as "yes". For example, to get warnings while connecting to the Citrix ADC appliance, the URL is as follows:

  http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/login?warning=yes

  If any, the warnings are displayed in the response payload with the HTTP code "209 X-NITRO-WARNING".

- **Authenticated access for individual NITRO operations:**NITRO allows you to logon to the Citrix ADC appliance to perform individual operations. You can use this option instead of creating a NITRO session (using the login object) and then using that session to perform all operations,

  To do this, you must specify the username and password in the request header of the NITRO request as follows:

  X-NITRO-USER:<username>

  X-NITRO-PASS:<password>

  Note: In such cases, make sure that the request header DOES not include the following:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue>

`*Note:` *

Mandatory parameters are marked in redand placeholder content is marked in <green>.

add

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/nspartition

HTTP Method:POST

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Content-Type:application/json

Request Payload:

```
{"nspartition":{

<b>"partitionname":<String_value>,

</b>"maxbandwidth":<Double_value>,

"minbandwidth":<Double_value>,

"maxconn":<Double_value>,

"maxmemlimit":<Double_value>,

"partitionmac":<String_value>

}}
```

Response:

HTTP Status Code on Success: 201 Created HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error

delete

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/nspartition/partitionname_value<String>

HTTP Method:DELETE

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue>

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error

update

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/nspartition

HTTP Method:PUT

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Content-Type:application/json

Request Payload:

```
{"nspartition":{

<b>"partitionname":<String_value>,

</b>"maxbandwidth":<Double_value>,

"minbandwidth":<Double_value>,

"maxconn":<Double_value>,

"maxmemlimit":<Double_value>,

"partitionmac":<String_value>

}}
```

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error

unset

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/nspartition?**action=unset**

HTTP Method:POST

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Content-Type:application/json

Request Payload:

```
{"nspartition":{

<b>"partitionname":<String_value>,

</b>"maxbandwidth":true,

"minbandwidth":true,

"maxconn":true,

"maxmemlimit":true,

"partitionmac":true

}}
```

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error

Switch

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/nspartition?**action=Switch**

HTTP Method:POST

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Content-Type:application/json

Request Payload:

```
{"nspartition":{

<b>"partitionname":<String_value>

</b>}}
```

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error

get (all)

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/nspartition

Query-parameters:

attrs

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/nspartition?**attrs=property-name1,property-name2**

Use this query parameter to specify the resource details that you want to retrieve.

filter

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/nspartition?**filter=property-name1:property-val1,property-name2:property-val2**

Use this query-parameter to get the filtered set of nspartition resources configured on Citrix ADC. Filtering can be done on any of the properties of the resource.

view

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/nspartition?**view=summary**

Note: By default, the retrieved results are displayed in detail view (?view=detail).

pagination

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/nspartition?**pagesize=#no;pageno=#no**

Use this query-parameter to get the nspartition resources in chunks.

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "nspartition": [ {

"partitionname":<String_value>,

"partitionid":<Double_value>,

"partitiontype":<String_value>,

"maxbandwidth":<Double_value>,

"minbandwidth":<Double_value>,
```

```
"maxconn":<Double_value>,

"maxmemlimit":<Double_value>,

"partitionmac":<String_value>,

"pmacinternal":<Boolean_value>

}]}
```

get

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/nspartition/partitionname_value<String>

Query-parameters:

attrs

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/nspartition/partitionname_value<String>?**attrs=property-name1,property-name2**

Use this query parameter to specify the resource details that you want to retrieve.

view

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/nspartition/partitionname_value<String>?**view=summary**

Note: By default, the retrieved results are displayed in detail view (?view=detail).

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "nspartition": [ {
"partitionname":<String_value>,
"partitionid":<Double_value>,
"partitiontype":<String_value>,
"maxbandwidth":<Double_value>,
"minbandwidth":<Double_value>,
"maxconn":<Double_value>,
"maxmemlimit":<Double_value>,
"partitionmac":<String_value>,
"pmacinternal":<Boolean_value>
}]}
```

count

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/nspartition?**count=yes**

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "nspartition": [ { "__count": "#no"} ] }
```

nspartitionmac

Configuration for partition MACs resource.

Properties

(click to see Operations)

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
partitionmac	<String>	Read-only	A configured virtual MAC address that can be used by a partition for communication over shared VLANs.
partitionname	<String>	Read-only	Name of the partition to which the MAC has been assigned. Minimum length = 1
__count	<Double>	Read-only	count parameter

Operations

(click to see Properties)

GET (ALL)| COUNT

Some options that you can use for each operations:

- **Getting warnings in response:**NITRO allows you to get warnings in an operation by specifying the "warning" query parameter as "yes". For example, to get warnings while connecting to the Citrix ADC appliance, the URL is as follows:

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/login?warning=yes

If any, the warnings are displayed in the response payload with the HTTP code "209 X-NITRO-WARNING".

- **Authenticated access for individual NITRO operations:**NITRO allows you to logon to the Citrix ADC appliance to perform individual operations. You can use this option instead of creating a NITRO session (using the login object) and then using that session to perform all operations,

  To do this, you must specify the username and password in the request header of the NITRO request as follows:

  X-NITRO-USER:<username>

  X-NITRO-PASS:<password>

  Note: In such cases, make sure that the request header DOES not include the following:

  Cookie:NITRO_AUTH_TOKEN=<tokenvalue>

`*Note:`*

Mandatory parameters are marked in redand placeholder content is marked in <green>.

get (all)

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/nspartitionmac

Query-parameters:

attrs

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/nspartitionmac?**attrs=property-name1,property-name2**

Use this query parameter to specify the resource details that you want to retrieve.

filter

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/nspartitionmac?**filter=property-name1:property-val1,property-name2:property-val2**

Use this query-parameter to get the filtered set of nspartitionmac resources configured on Citrix ADC. Filtering can be done on any of the properties of the resource.

view

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/nspartitionmac?**view=summary**

Note: By default, the retrieved results are displayed in detail view (?view=detail).

pagination

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/nspartitionmac?**pagesize=#no;pageno=#no**

Use this query-parameter to get the nspartitionmac resources in chunks.

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "nspartitionmac": [ {

"partitionmac":<String_value>,

"partitionname":<String_value>

}]}
```

count

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/nspartitionmac?**count=yes**

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "nspartitionmac": [ { "__count": "#no"} ] }
```

nspartition_binding

Binding object which returns the resources bound to nspartition.

Properties

(click to see Operations)

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
partitionname	<String>	Read-write	Name of partition for which to display parameters. Minimum length = 1
nspartition_bridgegroup_binding	<nspartition_bridgegroup_binding[]>	Read-only	bridgegroup that can be bound to nspartition.
nspartition_vlan_binding	<nspartition_vlan_binding[]>	Read-only	vlan that can be bound to nspartition.
nspartition_vxlan_binding	<nspartition_vxlan_binding[]>	Read-only	vxlan that can be bound to nspartition.

Operations

(click to see Properties)

GET| GET (ALL)

Some options that you can use for each operations:

- **Getting warnings in response:**NITRO allows you to get warnings in an operation by specifying the "warning" query parameter as "yes". For example, to get warnings while connecting to the Citrix ADC appliance, the URL is as follows:

  http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/login?warning=yes

  If any, the warnings are displayed in the response payload with the HTTP code "209 X-NITRO-WARNING".

- **Authenticated access for individual NITRO operations:**NITRO allows you to logon to the Citrix ADC appliance to perform individual operations. You can use this option instead of creating a NITRO session (using the login object) and then using that session to perform all operations,

  To do this, you must specify the username and password in the request header of the NITRO request as follows:

  X-NITRO-USER:<username>

  X-NITRO-PASS:<password>

  Note: In such cases, make sure that the request header DOES not include the following:

  Cookie:NITRO_AUTH_TOKEN=<tokenvalue>

`*Note:`*

Mandatory parameters are marked in redand placeholder content is marked in <green>.

get

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/nspartition_binding/partitionname_value<String>

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "nspartition_binding": [ {

"partitionname":<String_value>,

"nspartition_vxlan_binding":<nspartition_vxlan_binding[]_value>,

"nspartition_bridgegroup_binding":<nspartition_bridgegroup_binding[]_value
>,

"nspartition_vlan_binding":<nspartition_vlan_binding[]_value>

}]}
```

get (all)

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/nspartition_binding

Query-parameters:

bulkbindings

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/nspartition_binding?**bulkbindings=yes**

NITRO allows you to fetch bindings in bulk.

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "nspartition_binding": [ {

"partitionname":<String_value>,

"nspartition_vxlan_binding":<nspartition_vxlan_binding[]_value>,

"nspartition_bridgegroup_binding":<nspartition_bridgegroup_binding[]_value
>,

"nspartition_vlan_binding":<nspartition_vlan_binding[]_value>

}]}
```

nspartition_bridgegroup_binding

Binding object showing the bridgegroup that can be bound to nspartition.

Properties

(click to see Operations)

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
partitionname	<String>	Read-write	Name of the Partition. Must begin with an ASCII alphanumeric or underscore (_) character, and must contain only ASCII

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
			alphanumeric, underscore, hash (#), period (.), space, colon (:), at (@), equals (=), and hyphen (-) characters. Minimum length = 1
bridgegroup	<Double>	Read-write	Identifier of the bridge group that is assigned to this partition. Minimum value = 1 Maximum value = 1000
__count	<Double>	Read-write	count parameter

Operations

(click to see Properties)

ADD:| DELETE:| GET| GET (ALL)| COUNT

Some options that you can use for each operations:

- **Getting warnings in response:**NITRO allows you to get warnings in an operation by specifying the "warning" query parameter as "yes". For example, to get warnings while connecting to the Citrix ADC appliance, the URL is as follows:

  http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/login?warning=yes

  If any, the warnings are displayed in the response payload with the HTTP code "209 X-NITRO-WARNING".

- **Authenticated access for individual NITRO operations:**NITRO allows you to logon to the Citrix ADC appliance to perform individual operations. You can use this option instead of creating a NITRO session (using the login object) and then using that session to perform all operations,

To do this, you must specify the username and password in the request header of the NITRO request as follows:

X-NITRO-USER:<username>

X-NITRO-PASS:<password>

Note: In such cases, make sure that the request header DOES not include the following:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue>

`*Note:`*

Mandatory parameters are marked in redand placeholder content is marked in <green>.

add

URL:http://<Citrix-ADC-IP-address(NSIP)/nitro/v1/config/nspartition_bridgegroup_binding

HTTP Method:PUT

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Content-Type:application/json

Request Payload:

```
{

"nspartition_bridgegroup_binding":{

<b>"partitionname":<String_value>,

</b>"bridgegroup":<Double_value>

}}
```

Response:

HTTP Status Code on Success: 201 Created HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error

delete

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/nspartition_bridgegroup_binding/partitionname_value<String>

Query-parameters:

args

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/nspartition_bridgegroup_binding/partitionname_value<String>?**args=bridgegroup:<Double_value>**

HTTP Method:DELETE

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue>

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error

get

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/nspartition_bridgegroup_binding/partitionname_value<String>

Query-parameters:

filter

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/nspartition_bridgegroup_binding/partitionname_value<String>?**filter=property-name1:property-value1,property-name2:property-value2**

Use this query-parameter to get the filtered set of nspartition_bridgegroup_binding resources configured on Citrix ADC. Filtering can be done on any of the properties of the resource.

pagination

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/nspartition_bridgegroup_binding/partitionname_value<String>?**pagesize=#no;pageno=#no**

Use this query-parameter to get the nspartition_bridgegroup_binding resources in chunks.

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "nspartition_bridgegroup_binding": [ {

"partitionname":<String_value>,

"bridgegroup":<Double_value>

}]}
```

get (all)

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/nspartition_bridgegroup_binding

Query-parameters:

bulkbindings

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/nspartition_bridgegroup_binding?**bulkbindings=yes**

NITRO allows you to fetch bindings in bulk.

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "nspartition_bridgegroup_binding": [ {

"partitionname":<String_value>,

"bridgegroup":<Double_value>

}]}
```

count

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/nspartition_bridgegroup_binding/partitionname_value<String>?**count=yes**

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{"nspartition_bridgegroup_binding": [ { "__count": "#no"} ] }
```

nspartition_vlan_binding

Binding object showing the vlan that can be bound to nspartition.

Properties

(click to see Operations)

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
partitionname	<String>	Read-write	Name of the Partition. Must begin with an ASCII alphanumeric or underscore (_) character, and must contain only ASCII alphanumeric, underscore, hash (#), period (.), space, colon (:), at (@), equals (=), and hyphen (-) characters. Minimum length = 1
vlan	<Double>	Read-write	Identifier of the vlan that is assigned to this partition. Minimum value = 1 Maximum value = 4094
__count	<Double>	Read-write	count parameter

Operations

(click to see Properties)

ADD:| DELETE:| GET| GET (ALL)| COUNT

Some options that you can use for each operations:

- **Getting warnings in response:**NITRO allows you to get warnings in an operation by specifying the "warning" query parameter as "yes". For example, to get warnings while connecting to the Citrix ADC appliance, the URL is as follows:

  http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/login?warning=yes

  If any, the warnings are displayed in the response payload with the HTTP code "209 X-NITRO-WARNING".

- **Authenticated access for individual NITRO operations:**NITRO allows you to logon to the Citrix ADC appliance to perform individual operations. You can use this option instead of creating a NITRO session (using the login object) and then using that session to perform all operations,

  To do this, you must specify the username and password in the request header of the NITRO request as follows:

  X-NITRO-USER:<username>

  X-NITRO-PASS:<password>

  Note: In such cases, make sure that the request header DOES not include the following:

  Cookie:NITRO_AUTH_TOKEN=<tokenvalue>

`*Note:`*

Mandatory parameters are marked in redand placeholder content is marked in <green>.

add

URL:http://<Citrix-ADC-IP-address(NSIP)/nitro/v1/config/nspartition_vlan_binding

HTTP Method:PUT

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Content-Type:application/json

Request Payload:

```
{

"nspartition_vlan_binding":{

<b>"partitionname":<String_value>,

</b>"vlan":<Double_value>

}}
```

Response:

HTTP Status Code on Success: 201 Created HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error

delete

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/nspartition_vlan_binding/partitionname_value<String>

Query-parameters:

args

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/nspartition_vlan_binding/partitionname_value<String>?**args=vlan:<Double_value>**

HTTP Method:DELETE

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue>

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error

get

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/nspartition_vlan_binding/partitionname_value<String>

Query-parameters:

filter

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/nspartition_vlan_binding/partitionname_value<String>?**filter=property-name1:property-value1,property-name2:property-value2**

Use this query-parameter to get the filtered set of nspartition_vlan_binding resources configured on Citrix ADC. Filtering can be done on any of the properties of the resource.

pagination

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/nspartition_vlan_binding/partitionname_value<String>?**pagesize=#no;pageno=#no**

Use this query-parameter to get the nspartition_vlan_binding resources in chunks.

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "nspartition_vlan_binding": [ {

"partitionname":<String_value>,
```

```
"vlan":<Double_value>

}]}
```

get (all)

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/nspartition_vlan_binding

Query-parameters:

bulkbindings

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/nspartition_vlan_binding?**bulkbindings=yes**

NITRO allows you to fetch bindings in bulk.

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "nspartition_vlan_binding": [ {

"partitionname":<String_value>,

"vlan":<Double_value>

}]}
```

count

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/nspartition_vlan_binding/partitionname_value<String>?**count=yes**

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{"nspartition_vlan_binding": [ { "__count": "#no"} ] }
```

nspartition_vxlan_binding

Binding object showing the vxlan that can be bound to nspartition.

Properties

(click to see Operations)

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
partitionname	<String>	Read-write	Name of the Partition. Must begin with an ASCII alphanumeric or underscore (_) character, and must contain only ASCII alphanumeric,

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
			underscore, hash (#), period (.), space, colon (:), at (@), equals (=), and hyphen (-) characters. Minimum length = 1
vxlan	<Double>	Read-write	Identifier of the vxlan that is assigned to this partition. Minimum value = 1 Maximum value = 16777215
__count	<Double>	Read-write	count parameter

Operations

(click to see Properties)

ADD:| DELETE:| GET| GET (ALL)| COUNT

Some options that you can use for each operations:

- **Getting warnings in response:**NITRO allows you to get warnings in an operation by specifying the "warning" query parameter as "yes". For example, to get warnings while connecting to the Citrix ADC appliance, the URL is as follows:

  http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/login?warning=yes

  If any, the warnings are displayed in the response payload with the HTTP code "209 X-NITRO-WARNING".

- **Authenticated access for individual NITRO operations:**NITRO allows you to logon to the Citrix ADC appliance to perform individual operations. You can use this option instead of creating a NITRO session (using the login object) and then using that session to perform all operations,

  To do this, you must specify the username and password in the request header of the NITRO request as follows:

X-NITRO-USER:<username>

X-NITRO-PASS:<password>

Note: In such cases, make sure that the request header DOES not include the following:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue>

`*Note:`*

Mandatory parameters are marked in redand placeholder content is marked in <green>.

add

URL:http://<Citrix-ADC-IP-address(NSIP)/nitro/v1/config/nspartition_vxlan_binding

HTTP Method:PUT

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Content-Type:application/json

Request Payload:

```
{

"nspartition_vxlan_binding":{

<b>"partitionname":<String_value>,

</b>"vxlan":<Double_value>

}}
```

Response:

HTTP Status Code on Success: 201 Created HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error

delete

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/nspartition_vxlan_binding/partitionname_value<String>

Query-parameters:

args

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/nspartition_vxlan_binding/partitionname_value<String>?**args=vxlan:<Double_value>**

HTTP Method:DELETE

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue>

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error

get

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/nspartition_vxlan_binding/partitionname_value<String>

Query-parameters:

filter

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/nspartition_vxlan_binding/partitionname_value<String>?**filter=property-name1:property-value1,property-name2:property-value2**

Use this query-parameter to get the filtered set of nspartition_vxlan_binding resources configured on Citrix ADC. Filtering can be done on any of the properties of the resource.

pagination

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/nspartition_vxlan_binding/partitionname_value<String>?**pagesize=#no;pageno=#no**

Use this query-parameter to get the nspartition_vxlan_binding resources in chunks.

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "nspartition_vxlan_binding": [ {

"partitionname":<String_value>,

"vxlan":<Double_value>

}]}
```

get (all)

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/nspartition_vxlan_binding

Query-parameters:

bulkbindings

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/nspartition_vxlan_binding?**bulkbindings=yes**

NITRO allows you to fetch bindings in bulk.

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "nspartition_vxlan_binding": [ {

"partitionname":<String_value>,

"vxlan":<Double_value>

}]}
```

count

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/nspartition_vxlan_binding/partitionname_value<String>?**count=yes**

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{"nspartition_vxlan_binding": [ { "__count": "#no"} ] }
```

nspbr

Configuration for Policy Based Routing(PBR) entry resource.

Properties

(click to see Operations)

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
name	<String>	Read-write	Name for the PBR. Must begin with an ASCII alphabetic or underscore \(_\) character, and must contain only ASCII alphanumeric, underscore, hash \(#\), period \(.\), space, colon \(:\), at \(@\), equals \(=\), and hyphen \(-\) characters. Cannot be changed after the PBR is created. Minimum length = 1
action	<String>	Read-write	Action to perform on the outgoing IPv4 packets that match the PBR. Available settings function as follows: * ALLOW - The Citrix ADC sends the packet to the

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
			designated next-hop router. * DENY - The Citrix ADC applies the routing table for normal destination-based routing. Possible values = ALLOW, DENY
td	<Double>	Read-write	Integer value that uniquely identifies the traffic domain in which you want to configure the entity. If you do not specify an ID, the entity becomes part of the default traffic domain, which has an ID of 0. Minimum value = 0 Maximum value = 4094
srcip	<Boolean>	Read-write	IP address or range of IP addresses to match against the source IP address of an outgoing IPv4 packet. In the command line interface, separate the range with a hyphen. For example:

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
			10.102.29.30-10.102.29.189.
srcipop	<String>	Read-write	Either the equals (=) or does not equal (!=) logical operator. Possible values = =, !=, EQ, NEQ
srcipval	<String>	Read-write	IP address or range of IP addresses to match against the source IP address of an outgoing IPv4 packet. In the command line interface, separate the range with a hyphen. For example: 10.102.29.30-10.102.29.189.
srcport	<Boolean>	Read-write	Port number or range of port numbers to match against the source port number of an outgoing IPv4 packet. In the command line interface, separate the range with a hyphen. For example: 40-90. Note: The destination port can be specified only for TCP and UDP protocols.

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
srcportop	<String>	Read-write	Either the equals (=) or does not equal (!=) logical operator. Possible values = =, !=, EQ, NEQ
srcportval	<String>	Read-write	Port number or range of port numbers to match against the source port number of an outgoing IPv4 packet. In the command line interface, separate the range with a hyphen. For example: 40-90. Note: The destination port can be specified only for TCP and UDP protocols. Maximum length = 65535
destip	<Boolean>	Read-write	IP address or range of IP addresses to match against the destination IP address of an outgoing IPv4 packet. In the command line interface, separate the range with a hyphen. For example:

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
			10.102.29.30-10.102.29.189.
destipop	<String>	Read-write	Either the equals (=) or does not equal (!=) logical operator. Possible values = =, !=, EQ, NEQ
destipval	<String>	Read-write	IP address or range of IP addresses to match against the destination IP address of an outgoing IPv4 packet. In the command line interface, separate the range with a hyphen. For example: 10.102.29.30-10.102.29.189.
destport	<Boolean>	Read-write	Port number or range of port numbers to match against the destination port number of an outgoing IPv4 packet. In the command line interface, separate the range with a hyphen. For example: 40-90. Note: The destination port can be specified only for

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
			TCP and UDP protocols.
destportop	<String>	Read-write	Either the equals (=) or does not equal (!=) logical operator. Possible values = =, !=, EQ, NEQ
destportval	<String>	Read-write	Port number or range of port numbers to match against the destination port number of an outgoing IPv4 packet. In the command line interface, separate the range with a hyphen. For example: 40-90. Note: The destination port can be specified only for TCP and UDP protocols. Maximum length = 65535
nexthop	<Boolean>	Read-write	IP address of the next hop router or the name of the link load balancing virtual server to which to send matching packets if action is set to ALLOW.

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
			If you specify a link load balancing (LLB) virtual server, which can provide a backup if a next hop link fails, first make sure that the next hops bound to the LLB virtual server are actually next hops that are directly connected to the Citrix ADC. Otherwise, the Citrix ADC throws an error when you attempt to create the PBR. The next hop can be null to represent null routes.
nexthopval	<String>	Read-write	The Next Hop IP address or gateway name.
iptunnel	<Boolean>	Read-write	The Tunnel name.
iptunnelname	<String>	Read-write	The iptunnel name where packets need to be forwarded upon.
vxlanvlanmap	<String>	Read-write	The vlan to vxlan mapping to be applied for incoming packets over this pbr tunnel.

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
srcmac	<String>	Read-write	MAC address to match against the source MAC address of an outgoing IPv4 packet.
srcmacmask	<String>	Read-write	Used to define range of Source MAC address. It takes string of 0 and 1, 0s are for exact match and 1s for wildcard. For matching first 3 bytes of MAC address, srcMacMask value "000000111111". . Default value: "000000000000"
protocol	<String>	Read-write	Protocol, identified by protocol name, to match against the protocol of an outgoing IPv4 packet. Possible values = ICMP, IGMP, TCP, EGP, IGP, ARGUS, UDP, RDP, RSVP, EIGRP, L2TP, ISIS, GGP, IPoverIP, ST, CBT, BBN-RCC-M, NVP-II, PUP, EMCON, XNET, CHAOS, MUX, DCN-MEAS, HMP, PRM, XNS-IDP, TRUNK-1,

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
			TRUNK-2, LEAF-1, LEAF-2, IRTP, ISO-TP4, NETBLT, MFE-NSP, MERIT-INP, SEP, 3PC, IDPR, XTP, DDP, IDPR-CMTP, TP++, IL, IPv6, SDRP, IPv6-Route, IPv6-Frag, IDRP, GRE, MHRP, BNA, ESP, AH, I-NLSP, SWIPE, NARP, MOBILE, TLSP, SKIP, ICMPV6, IPv6-NoNx, IPv6-Opts, Any-Host-Internal-Protocol, CFTP, Any-Local-Network, SAT-EXPAK, KRYPTOLAN, RVD, IPPC, Any-Distributed-File-System, TFTP, VISA, IPCV, CPNX, CPHB, WSN, PVP, BR-SAT-MO, SUN-ND, WB-MON, WB-EXPAK, ISO-IP, VMTP, SECURE-VM, VINES, TTP, NSFNET-IG, DGP, TCF, OSPFIGP, Sprite-RP, LARP, MTP, AX.25, IPIP,

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
			MICP, SCC-SP, ETHERIP, Any-Private-Encryption-Scheme, GMTP, IFMP, PNNI, PIM, ARIS, SCPS, QNX, A/N, IPComp, SNP, Compaq-Pe, IPX-in-IP, VRRP, PGM, Any-0-Hop-Protocol, ENCAP, DDX, IATP, STP, SRP, UTI, SMP, SM, PTP, FIRE, CRTP, CRUDP, SSCOPMCE, IPLT, SPS, PIPE, SCTP, FC, RSVP-E2E-IGNORE, Mobility-Header, UDPLite
protocolnumber	<Double>	Read-write	Protocol, identified by protocol number, to match against the protocol of an outgoing IPv4 packet. Minimum value = 1 Maximum value = 255
vlan	<Double>	Read-write	ID of the VLAN. The Citrix ADC compares the PBR only to the outgoing packets on the specified VLAN. If you do not specify

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
			any interface ID, the appliance compares the PBR to the outgoing packets on all VLANs. Minimum value = 1 Maximum value = 4094
vxlan	<Double>	Read-write	ID of the VXLAN. The Citrix ADC compares the PBR only to the outgoing packets on the specified VXLAN. If you do not specify any interface ID, the appliance compares the PBR to the outgoing packets on all VXLANs. Minimum value = 1 Maximum value = 16777215
Interface	<String>	Read-write	ID of an interface. The Citrix ADC compares the PBR only to the outgoing packets on the specified interface. If you do not specify any value, the appliance compares the PBR to the outgoing packets on all interfaces.

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
priority	<Double>	Read-write	Priority of the PBR, which determines the order in which it is evaluated relative to the other PBRs. If you do not specify priorities while creating PBRs, the PBRs are evaluated in the order in which they are created. Minimum value = 1 Maximum value = 81920
msr	<String>	Read-write	Monitor the route specified byte Next Hop parameter. This parameter is not applicable if you specify a link load balancing (LLB) virtual server name with the Next Hop parameter. Default value: DISABLED Possible values = ENABLED, DISABLED
monitor	<String>	Read-write	The name of the monitor.(Can be only of type ping or ARP). Minimum length = 1

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
state	<String>	Read-write	Enable or disable the PBR. After you apply the PBRs, the Citrix ADC compares outgoing packets to the enabled PBRs. Default value: ENABLED Possible values = ENABLED, DISABLED
ownergroup	<String>	Read-write	The owner node group in a Cluster for this pbr rule. If ownernode is not specified then the pbr rule is treated as Striped pbr rule. Default value: DEFAULT_NG Minimum length = 1
detail	<Boolean>	Read-write	To get a detailed view.
hits	<Double>	Read-only	The hits of this PBR.
kernelstate	<String>	Read-only	The commit status of the PBR. Default value: NOTAPPLIED Possible values = APPLIED, NOTAPPLIED, RE-APPLY, SFAPPLIED,

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
			SFNOTAPPLIED, SFAPPLIED61, SFNOTAPPLIED61
curstate	<Double>	Read-only	If this route is UP/DOWN.
totalprobes	<Double>	Read-only	The total number of probes sent.
totalfailedprobes	<Double>	Read-only	The total number of failed probes.
failedprobes	<Double>	Read-only	Number of the current failed monitoring probes.
monstatcode	<Integer>	Read-only	The code indicating the monitor response.
monstatparam1	<Integer>	Read-only	First parameter for use with message code.
monstatparam2	<Integer>	Read-only	Second parameter for use with message code.
monstatparam3	<Integer>	Read-only	Third parameter for use with message code.
data	<Boolean>	Read-only	Internal data of this route.
__count	<Double>	Read-only	count parameter

Operations

(click to see Properties)

ADD| DELETE| UPDATE| UNSET| ENABLE| DISABLE| GET (ALL)| GET| COUNT

Some options that you can use for each operations:

- **Getting warnings in response:**NITRO allows you to get warnings in an operation by specifying the "warning" query parameter as "yes". For example, to get warnings while connecting to the Citrix ADC appliance, the URL is as follows:

  http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/login?warning=yes

  If any, the warnings are displayed in the response payload with the HTTP code "209 X-NITRO-WARNING".

- **Authenticated access for individual NITRO operations:**NITRO allows you to logon to the Citrix ADC appliance to perform individual operations. You can use this option instead of creating a NITRO session (using the login object) and then using that session to perform all operations,

  To do this, you must specify the username and password in the request header of the NITRO request as follows:

  X-NITRO-USER:<username>

  X-NITRO-PASS:<password>

  Note: In such cases, make sure that the request header DOES not include the following:

  Cookie:NITRO_AUTH_TOKEN=<tokenvalue>

`*Note:`*

Mandatory parameters are marked in redand placeholder content is marked in <green>.

add

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/nspbr

HTTP Method:POST

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Content-Type:application/json

Request Payload:

```
{"nspbr":{
<b>"name":<String_value>,
</b><b>"action":<String_value>,
</b>"td":<Double_value>,
"srcip":<Boolean_value>,
"srcipop":<String_value>,
"srcipval":<String_value>,
"srcport":<Boolean_value>,
"srcportop":<String_value>,
"srcportval":<String_value>,
"destip":<Boolean_value>,
"destipop":<String_value>,
"destipval":<String_value>,
"destport":<Boolean_value>,
"destportop":<String_value>,
"destportval":<String_value>,
"nexthop":<Boolean_value>,
"nexthopval":<String_value>,
"iptunnel":<Boolean_value>,
"iptunnelname":<String_value>,
"vxlanvlanmap":<String_value>,
```

```
"srcmac":<String_value>,

"srcmacmask":<String_value>,

"protocol":<String_value>,

"protocolnumber":<Double_value>,

"vlan":<Double_value>,

"vxlan":<Double_value>,

"Interface":<String_value>,

"priority":<Double_value>,

"msr":<String_value>,

"monitor":<String_value>,

"state":<String_value>,

"ownergroup":<String_value>

}}
```

Response:

HTTP Status Code on Success: 201 Created HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error

delete

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/nspbr/name_value<String>

HTTP Method:DELETE

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue>

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error

update

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/nspbr

HTTP Method:PUT

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Content-Type:application/json

Request Payload:

```
{"nspbr":{

<b>"name":<String_value>,

</b>"action":<String_value>,

"srcip":<Boolean_value>,

"srcipop":<String_value>,

"srcipval":<String_value>,

"srcport":<Boolean_value>,

"srcportop":<String_value>,

"srcportval":<String_value>,

"destip":<Boolean_value>,

"destipop":<String_value>,

"destipval":<String_value>,
```

```
"destport":<Boolean_value>,

"destportop":<String_value>,

"destportval":<String_value>,

"nexthop":<Boolean_value>,

"nexthopval":<String_value>,

"iptunnel":<Boolean_value>,

"iptunnelname":<String_value>,

"vxlanvlanmap":<String_value>,

"srcmac":<String_value>,

"srcmacmask":<String_value>,

"protocol":<String_value>,

"protocolnumber":<Double_value>,

"vlan":<Double_value>,

"vxlan":<Double_value>,

"Interface":<String_value>,

"priority":<Double_value>,

"msr":<String_value>,

"monitor":<String_value>

}}
```

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error

unset

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/nspbr?**action=unset**

HTTP Method:POST

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Content-Type:application/json

Request Payload:

```
{"nspbr":{

<b>"name":<String_value>,

</b>"srcip":true,

"srcport":true,

"destip":true,

"destport":true,

"nexthop":true,

"iptunnel":true,

"vxlanvlanmap":true,

"srcmac":true,

"srcmacmask":true,

"protocol":true,

"vlan":true,
```

```
"vxlan":true,

"Interface":true,

"msr":true,

"monitor":true

}}
```

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error

enable

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/nspbr?**action=enable**

HTTP Method:POST

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Content-Type:application/json

Request Payload:

```
{"nspbr":{

<b>"name":<String_value>

</b>}}
```

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error

disable

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/nspbr?**action=disable**

HTTP Method:POST

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Content-Type:application/json

Request Payload:

```
{"nspbr":{

<b>"name":<String_value>

</b>}}
```

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error

get (all)

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/nspbr

Query-parameters:

args

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/nspbr?**args=name:<String_value>,detail:<Boolean_value>**

Use this query-parameter to get nspbr resources based on additional properties.

attrs

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/nspbr?**attrs=property-name1,property-name2**

Use this query parameter to specify the resource details that you want to retrieve.

filter

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/nspbr?**filter=property-name1:property-val1,property-name2:property-val2**

Use this query-parameter to get the filtered set of nspbr resources configured on Citrix ADC. Filtering can be done on any of the properties of the resource.

view

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/nspbr?**view=summary**

Note: By default, the retrieved results are displayed in detail view (?view=detail).

pagination

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/nspbr?**pagesize=#no;pageno=#no**

Use this query-parameter to get the nspbr resources in chunks.

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "nspbr": [ {

name:<String_value>,detail:<Boolean_value>"action":<String_value>,

"srcmac":<String_value>,
```

```
"srcmacmask":<String_value>,

"protocol":<String_value>,

"protocolnumber":<Double_value>,

"srcportval":<String_value>,

"srcportop":<String_value>,

"td":<Double_value>,

"destportval":<String_value>,

"destportop":<String_value>,

"srcipval":<String_value>,

"srcipop":<String_value>,

"destipval":<String_value>,

"destipop":<String_value>,

"vlan":<Double_value>,

"vxlan":<Double_value>,

"state":<String_value>,

"Interface":<String_value>,

"hits":<Double_value>,

"priority":<Double_value>,

"kernelstate":<String_value>,

"nexthopval":<String_value>,

"iptunnelname":<String_value>,

"vxlanvlanmap":<String_value>,
```

```
"msr":<String_value>,

"monitor":<String_value>,

"curstate":<Double_value>,

"totalprobes":<Double_value>,

"totalfailedprobes":<Double_value>,

"failedprobes":<Double_value>,

"monstatcode":<Integer_value>,

"monstatparam1":<Integer_value>,

"monstatparam2":<Integer_value>,

"monstatparam3":<Integer_value>,

"data":<Boolean_value>,

"ownergroup":<String_value>

}]}
```

get

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/nspbr/name_value<String>

Query-parameters:

attrs

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/nspbr/name_value<String>?**attrs=property-name1,property-name2**

Use this query parameter to specify the resource details that you want to retrieve.

view

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/nspbr/name_value<String>?**view=summary**

Note: By default, the retrieved results are displayed in detail view (?view=detail).

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "nspbr": [ {

name:<String_value>,detail:<Boolean_value>"action":<String_value>,

"srcmac":<String_value>,

"srcmacmask":<String_value>,

"protocol":<String_value>,

"protocolnumber":<Double_value>,

"srcportval":<String_value>,

"srcportop":<String_value>,

"td":<Double_value>,

"destportval":<String_value>,

"destportop":<String_value>,

"srcipval":<String_value>,
```

```
"srcipop":<String_value>,

"destipval":<String_value>,

"destipop":<String_value>,

"vlan":<Double_value>,

"vxlan":<Double_value>,

"state":<String_value>,

"Interface":<String_value>,

"hits":<Double_value>,

"priority":<Double_value>,

"kernelstate":<String_value>,

"nexthopval":<String_value>,

"iptunnelname":<String_value>,

"vxlanvlanmap":<String_value>,

"msr":<String_value>,

"monitor":<String_value>,

"curstate":<Double_value>,

"totalprobes":<Double_value>,

"totalfailedprobes":<Double_value>,

"failedprobes":<Double_value>,

"monstatcode":<Integer_value>,

"monstatparam1":<Integer_value>,

"monstatparam2":<Integer_value>,
```

```
"monstatparam3":<Integer_value>,

"data":<Boolean_value>,

"ownergroup":<String_value>

}]}
```

count

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/nspbr?**count=yes**

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "nspbr": [ { "__count": "#no"} ] }
```

nspbr6

Configuration for PBR6 entry resource.

Properties

(click to see Operations)

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
name	<String>	Read-write	Name for the PBR6. Must begin with an ASCII alphabetic or underscore \(_\) character, and must contain only ASCII alphanumeric, underscore, hash \(#\), period \(.\), space, colon \(:\), at \(@\), equals \(=\), and hyphen \(-\) characters. Cannot be changed after the PBR6 is created. Minimum length = 1
td	<Double>	Read-write	Integer value that uniquely identifies the traffic domain in which you want to configure the entity. If you do not specify an ID, the entity becomes part of the default traffic domain, which has an ID of 0. Minimum value = 0 Maximum value = 4094
action	<String>	Read-write	Action to perform on the outgoing IPv6 packets that match the PBR6. Available settings function as follows:

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
			* ALLOW - The Citrix ADC sends the packet to the designated next-hop router. * DENY - The Citrix ADC applies the routing table for normal destination-based routing. Possible values = ALLOW, DENY
srcipv6	<Boolean>	Read-write	IP address or range of IP addresses to match against the source IP address of an outgoing IPv6 packet. In the command line interface, separate the range with a hyphen.
srcipop	<String>	Read-write	Either the equals (=) or does not equal (!=) logical operator. Possible values = =, !=, EQ, NEQ
srcipv6val	<String>	Read-write	IP address or range of IP addresses to match against the source IP address of an outgoing IPv6 packet. In the command line interface, separate

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
			the range with a hyphen.
srcport	<Boolean>	Read-write	Port number or range of port numbers to match against the source port number of an outgoing IPv6 packet. In the command line interface, separate the range with a hyphen. For example: 40-90.
srcportop	<String>	Read-write	Either the equals (=) or does not equal (!=) logical operator. Possible values = =, !=, EQ, NEQ
srcportval	<String>	Read-write	Source port (range). Maximum length = 65535
destipv6	<Boolean>	Read-write	IP address or range of IP addresses to match against the destination IP address of an outgoing IPv6 packet. In the command line interface, separate the range with a hyphen.
destipop	<String>	Read-write	Either the equals (=) or does not equal (!=) logical operator.

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
			Possible values = =, !=, EQ, NEQ
destipv6val	<String>	Read-write	IP address or range of IP addresses to match against the destination IP address of an outgoing IPv6 packet. In the command line interface, separate the range with a hyphen.
destport	<Boolean>	Read-write	Port number or range of port numbers to match against the destination port number of an outgoing IPv6 packet. In the command line interface, separate the range with a hyphen. For example: 40-90. Note: The destination port can be specified only for TCP and UDP protocols.
destportop	<String>	Read-write	Either the equals (=) or does not equal (!=) logical operator. Possible values = =, !=, EQ, NEQ

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
destportval	<String>	Read-write	Destination port (range). Maximum length = 65535
srcmac	<String>	Read-write	MAC address to match against the source MAC address of an outgoing IPv6 packet.
srcmacmask	<String>	Read-write	Used to define range of Source MAC address. It takes string of 0 and 1, 0s are for exact match and 1s for wildcard. For matching first 3 bytes of MAC address, srcMacMask value "000000111111". . Default value: "000000000000"
protocol	<String>	Read-write	Protocol, identified by protocol name, to match against the protocol of an outgoing IPv6 packet. Possible values = ICMPV6, TCP, UDP, ICMP, IGMP, EGP, IGP, ARGUS, RDP, RSVP, EIGRP, L2TP, ISIS, GGP, IPoverIP, ST, CBT, BBN-RCC-M, NVP-

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
			II, PUP, EMCON, XNET, CHAOS, MUX, DCN-MEAS, HMP, PRM, XNS-IDP, TRUNK-1, TRUNK-2, LEAF-1, LEAF-2, IRTP, ISO-TP4, NETBLT, MFE-NSP, MERIT-INP, SEP, 3PC, IDPR, XTP, DDP, IDPR-CMTP, TP++, IL, IPv6, SDRP, IPv6-Route, IPv6-Frag, IDRP, GRE, MHRP, BNA, ESP, AH, I-NLSP, SWIPE, NARP, MOBILE, TLSP, SKIP, IPv6-NoNx, IPv6-Opts, Any-Host-Internal-Protocol, CFTP, Any-Local-Network, SAT-EXPAK, KRYPTOLAN, RVD, IPPC, Any-Distributed-File-System, TFTP, VISA, IPCV, CPNX, CPHB, WSN, PVP, BR-SAT-MO, SUN-ND, WB-MON, WB-EXPAK, ISO-IP, VMTP, SECURE-VM, VINES, TTP,

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
			NSFNET-IG, DGP, TCF, OSPFIGP, Sprite-RP, LARP, MTP, AX.25, IPIP, MICP, SCC-SP, ETHERIP, Any-Private-Encryption-Scheme, GMTP, IFMP, PNNI, PIM, ARIS, SCPS, QNX, A/N, IPComp, SNP, Compaq-Pe, IPX-in-IP, VRRP, PGM, Any-0-Hop-Protocol, ENCAP, DDX, IATP, STP, SRP, UTI, SMP, SM, PTP, FIRE, CRTP, CRUDP, SSCOPMCE, IPLT, SPS, PIPE, SCTP, FC, RSVP-E2E-IGNORE, Mobility-Header, UDPLite
protocolnumber	<Double>	Read-write	Protocol, identified by protocol number, to match against the protocol of an outgoing IPv6 packet. Minimum value = 1 Maximum value = 255

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
vlan	<Double>	Read-write	ID of the VLAN. The Citrix ADC compares the PBR6 only to the outgoing packets on the specified VLAN. If you do not specify an interface ID, the appliance compares the PBR6 to the outgoing packets on all VLANs. Minimum value = 1 Maximum value = 4094
vxlan	<Double>	Read-write	ID of the VXLAN. The Citrix ADC compares the PBR6 only to the outgoing packets on the specified VXLAN. If you do not specify an interface ID, the appliance compares the PBR6 to the outgoing packets on all VXLANs. Minimum value = 1 Maximum value = 16777215
Interface	<String>	Read-write	ID of an interface. The Citrix ADC compares the PBR6 only to the outgoing packets on the specified interface. If you do not specify a value, the appliance

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
			compares the PBR6 to the outgoing packets on all interfaces.
priority	<Double>	Read-write	Priority of the PBR6, which determines the order in which it is evaluated relative to the other PBR6s. If you do not specify priorities while creating PBR6s, the PBR6s are evaluated in the order in which they are created. Minimum value = 1 Maximum value = 81920
state	<String>	Read-write	Enable or disable the PBR6. After you apply the PBR6s, the Citrix ADC compares outgoing packets to the enabled PBR6s. Default value: ENABLED Possible values = ENABLED, DISABLED
msr	<String>	Read-write	Monitor the route specified by the Next Hop parameter. Default value: DISABLED

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
			Possible values = ENABLED, DISABLED
monitor	<String>	Read-write	The name of the monitor.(Can be only of type ping or ARP). Minimum length = 1
nexthop	<Boolean>	Read-write	IP address of the next hop router to which to send matching packets if action is set to ALLOW. This next hop should be directly reachable from the appliance.
nexthopval	<String>	Read-write	The Next Hop IPv6 address.
iptunnel	<String>	Read-write	The iptunnel name where packets need to be forwarded upon.
vxlanvlanmap	<String>	Read-write	The vlan to vxlan mapping to be applied for incoming packets over this pbr tunnel.
nexthopvlan	<Double>	Read-write	VLAN number to be used for link local nexthop . Minimum value = 1 Maximum value = 4094

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
ownergroup	<String>	Read-write	The owner node group in a Cluster for this pbr rule. If owner node group is not specified then the pbr rule is treated as Striped pbr rule. Default value: DEFAULT_NG Minimum length = 1
detail	<Boolean>	Read-write	To get a detailed view.
kernelstate	<String>	Read-only	Commit status of the PBR6. Default value: NOTAPPLIED Possible values = APPLIED, NOTAPPLIED, RE-APPLY, SFAPPLIED, SFNOTAPPLIED
hits	<Double>	Read-only	Number of hits of this PBR6.
curstate	<Double>	Read-only	If this route is UP/DOWN.
totalprobes	<Double>	Read-only	The total number of probes sent.
totalfailedprobes	<Double>	Read-only	The total number of failed probes.

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
failedprobes	<Double>	Read-only	Number of the current failed monitoring probes.
monstatcode	<Integer>	Read-only	The code indicating the monitor response.
monstatparam1	<Integer>	Read-only	First parameter for use with message code.
monstatparam2	<Integer>	Read-only	Second parameter for use with message code.
monstatparam3	<Integer>	Read-only	Third parameter for use with message code.
data	<Boolean>	Read-only	Internal data of this route.
__count	<Double>	Read-only	count parameter

Operations

(click to see Properties)

ADD| RENUMBER| DELETE| UPDATE| UNSET| ENABLE| DISABLE| CLEAR| APPLY| GET (ALL)| GET| COUNT

Some options that you can use for each operations:

- **Getting warnings in response:**NITRO allows you to get warnings in an operation by specifying the "warning" query parameter as "yes". For example, to get warnings while connecting to the Citrix ADC appliance, the URL is as follows:

  http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/login?warning=yes

  If any, the warnings are displayed in the response payload with the HTTP code "209 X-NITRO-WARNING".

- **Authenticated access for individual NITRO operations:**NITRO allows you to logon to the Citrix ADC appliance to perform individual operations. You can use this option instead of creating a NITRO session (using the login object) and then using that session to perform all operations,

  To do this, you must specify the username and password in the request header of the NITRO request as follows:

  X-NITRO-USER:<username>

  X-NITRO-PASS:<password>

  Note: In such cases, make sure that the request header DOES not include the following:

  Cookie:NITRO_AUTH_TOKEN=<tokenvalue>

`*Note:`*

Mandatory parameters are marked in redand placeholder content is marked in <green>.

add

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/nspbr6

HTTP Method:POST

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Content-Type:application/json

Request Payload:

```
{"nspbr6":{

<b>"name":<String_value>,

</b>"td":<Double_value>,

<b>"action":<String_value>,

</b>"srcipv6":<Boolean_value>,

"srcipop":<String_value>,
```

```
"srcipv6val":<String_value>,

"srcport":<Boolean_value>,

"srcportop":<String_value>,

"srcportval":<String_value>,

"destipv6":<Boolean_value>,

"destipop":<String_value>,

"destipv6val":<String_value>,

"destport":<Boolean_value>,

"destportop":<String_value>,

"destportval":<String_value>,

"srcmac":<String_value>,

"srcmacmask":<String_value>,

"protocol":<String_value>,

"protocolnumber":<Double_value>,

"vlan":<Double_value>,

"vxlan":<Double_value>,

"Interface":<String_value>,

"priority":<Double_value>,

"state":<String_value>,

"msr":<String_value>,

"monitor":<String_value>,

"nexthop":<Boolean_value>,
```

```
"nexthopval":<String_value>,

"iptunnel":<String_value>,

"vxlanvlanmap":<String_value>,

"nexthopvlan":<Double_value>,

"ownergroup":<String_value>

}}
```

Response:

HTTP Status Code on Success: 201 Created HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error

renumber

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/nspbr6?**action=renumber**

HTTP Method:POST

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Content-Type:application/json

Request Payload:

```
{"nspbr6":{

}}
```

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error

delete

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/nspbr6/name_value<String>

HTTP Method:DELETE

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue>

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error

update

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/nspbr6

HTTP Method:PUT

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Content-Type:application/json

Request Payload:

```
{"nspbr6":{

<b>"name":<String_value>,

</b>"action":<String_value>,

"srcipv6":<Boolean_value>,

"srcipop":<String_value>,

"srcipv6val":<String_value>,

"srcport":<Boolean_value>,

"srcportop":<String_value>,
```

```
"srcportval":<String_value>,

"destipv6":<Boolean_value>,

"destipop":<String_value>,

"destipv6val":<String_value>,

"destport":<Boolean_value>,

"destportop":<String_value>,

"destportval":<String_value>,

"srcmac":<String_value>,

"srcmacmask":<String_value>,

"protocol":<String_value>,

"protocolnumber":<Double_value>,

"vlan":<Double_value>,

"vxlan":<Double_value>,

"Interface":<String_value>,

"priority":<Double_value>,

"msr":<String_value>,

"monitor":<String_value>,

"nexthop":<Boolean_value>,

"nexthopval":<String_value>,

"iptunnel":<String_value>,

"vxlanvlanmap":<String_value>,

"nexthopvlan":<Double_value>
```

```
}}
```

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error

unset

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/nspbr6?**action=unset**

HTTP Method:POST

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Content-Type:application/json

Request Payload:

```
{"nspbr6":{

<b>"name":<String_value>,

</b>"srcipv6":true,

"srcport":true,

"destipv6":true,

"destport":true,

"srcmac":true,

"srcmacmask":true,

"protocol":true,

"Interface":true,
```

```
"vlan":true,

"vxlan":true,

"msr":true,

"monitor":true,

"nexthop":true,

"iptunnel":true,

"nexthopvlan":true,

"vxlanvlanmap":true

}}
```

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error

enable

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/nspbr6?**action=enable**

HTTP Method:POST

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Content-Type:application/json

Request Payload:

```
{"nspbr6":{

<b>"name":<String_value>
```

```
</b>}}
```

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error

disable

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/nspbr6?**action=disable**

HTTP Method:POST

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Content-Type:application/json

Request Payload:

```
{"nspbr6":{

<b>"name":<String_value>

</b>}}
```

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error

clear

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/nspbr6?**action=clear**

HTTP Method:POST

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Content-Type:application/json

Request Payload:

```
{"nspbr6":{

}}
```

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error

apply

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/nspbr6?**action=apply**

HTTP Method:POST

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Content-Type:application/json

Request Payload:

```
{"nspbr6":{

}}
```

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error

get (all)

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/nspbr6

Query-parameters:

args

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/nspbr6?**args=name:<String_value>,detail:<Boolean_value>**

Use this query-parameter to get nspbr6 resources based on additional properties.

attrs

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/nspbr6?**attrs=property-name1,property-name2**

Use this query parameter to specify the resource details that you want to retrieve.

filter

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/nspbr6?**filter=property-name1:property-val1,property-name2:property-val2**

Use this query-parameter to get the filtered set of nspbr6 resources configured on Citrix ADC. Filtering can be done on any of the properties of the resource.

view

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/nspbr6?**view=summary**

Note: By default, the retrieved results are displayed in detail view (?view=detail).

pagination

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/nspbr6?**pagesize=#no;pageno=#no**

Use this query-parameter to get the nspbr6 resources in chunks.

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "nspbr6": [ {

name:<String_value>,detail:<Boolean_value>"td":<Double_value>,

"action":<String_value>,

"srcmac":<String_value>,

"srcmacmask":<String_value>,

"protocol":<String_value>,

"protocolnumber":<Double_value>,

"srcportval":<String_value>,

"srcportop":<String_value>,

"destportval":<String_value>,

"destportop":<String_value>,

"srcipv6val":<String_value>,

"srcipop":<String_value>,

"destipv6val":<String_value>,

"destipop":<String_value>,

"vlan":<Double_value>,

"vxlan":<Double_value>,
```

```
"state":<String_value>,

"kernelstate":<String_value>,

"Interface":<String_value>,

"hits":<Double_value>,

"priority":<Double_value>,

"msr":<String_value>,

"monitor":<String_value>,

"curstate":<Double_value>,

"totalprobes":<Double_value>,

"totalfailedprobes":<Double_value>,

"failedprobes":<Double_value>,

"monstatcode":<Integer_value>,

"monstatparam1":<Integer_value>,

"monstatparam2":<Integer_value>,

"monstatparam3":<Integer_value>,

"nexthopval":<String_value>,

"iptunnel":<String_value>,

"vxlanvlanmap":<String_value>,

"nexthopvlan":<Double_value>,

"data":<Boolean_value>,

"ownergroup":<String_value>

}]}
```

get

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/nspbr6/name_value<String>

Query-parameters:

attrs

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/nspbr6/name_value<String>?**attrs=property-name1,property-name2**

Use this query parameter to specify the resource details that you want to retrieve.

view

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/nspbr6/name_value<String>?**view=summary**

Note: By default, the retrieved results are displayed in detail view (?view=detail).

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "nspbr6": [ {

name:<String_value>,detail:<Boolean_value>"td":<Double_value>,

"action":<String_value>,
```

```
"srcmac":<String_value>,

"srcmacmask":<String_value>,

"protocol":<String_value>,

"protocolnumber":<Double_value>,

"srcportval":<String_value>,

"srcportop":<String_value>,

"destportval":<String_value>,

"destportop":<String_value>,

"srcipv6val":<String_value>,

"srcipop":<String_value>,

"destipv6val":<String_value>,

"destipop":<String_value>,

"vlan":<Double_value>,

"vxlan":<Double_value>,

"state":<String_value>,

"kernelstate":<String_value>,

"Interface":<String_value>,

"hits":<Double_value>,

"priority":<Double_value>,

"msr":<String_value>,

"monitor":<String_value>,

"curstate":<Double_value>,
```

```
"totalprobes":<Double_value>,

"totalfailedprobes":<Double_value>,

"failedprobes":<Double_value>,

"monstatcode":<Integer_value>,

"monstatparam1":<Integer_value>,

"monstatparam2":<Integer_value>,

"monstatparam3":<Integer_value>,

"nexthopval":<String_value>,

"iptunnel":<String_value>,

"vxlanvlanmap":<String_value>,

"nexthopvlan":<Double_value>,

"data":<Boolean_value>,

"ownergroup":<String_value>

}]}
```

count

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/nspbr6?**count=yes**

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "nspbr6": [ { "__count": "#no"} ] }
```

nspbrs

Configuration for Policy based routing resource.

Operations

(click to see Properties)

RENUMBER| CLEAR| APPLY

Some options that you can use for each operations:

- **Getting warnings in response:**NITRO allows you to get warnings in an operation by specifying the "warning" query parameter as "yes". For example, to get warnings while connecting to the Citrix ADC appliance, the URL is as follows:

  http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/login?warning=yes

  If any, the warnings are displayed in the response payload with the HTTP code "209 X-NITRO-WARNING".

- **Authenticated access for individual NITRO operations:**NITRO allows you to logon to the Citrix ADC appliance to perform individual operations. You can use this option instead of creating a NITRO session (using the login object) and then using that session to perform all operations,

  To do this, you must specify the username and password in the request header of the NITRO request as follows:

  X-NITRO-USER:<username>

X-NITRO-PASS:<password>

Note: In such cases, make sure that the request header DOES not include the following:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue>

`*Note:`*

Mandatory parameters are marked in redand placeholder content is marked in <green>.

renumber

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/nspbrs?**action=renumber**

HTTP Method:POST

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Content-Type:application/json

Request Payload:

```
{"nspbrs":{

}}
```

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error

clear

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/nspbrs?**action=clear**

HTTP Method:POST

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Content-Type:application/json

Request Payload:

```
{"nspbrs":{

}}
```

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error

apply

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/nspbrs?**action=apply**

HTTP Method:POST

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Content-Type:application/json

Request Payload:

```
{"nspbrs":{

}}
```

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error

nsratecontrol

Configuration for rate control resource.

Properties

(click to see Operations)

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
tcpthreshold	<Double>	Read-write	Number of SYNs permitted per 10 milliseconds.
udpthreshold	<Double>	Read-write	Number of UDP packets permitted per 10 milliseconds.
icmpthreshold	<Double>	Read-write	Number of ICMP packets permitted per 10 milliseconds. Default value: 100
tcprstthreshold	<Double>	Read-write	The number of TCP RST packets permitted per 10 milli second. zero means rate control is disabled and 0xffffffff means every thing is rate controlled. Default value: 100

Operations

(click to see Properties)

UPDATE| UNSET| GET (ALL)

Some options that you can use for each operations:

- **Getting warnings in response:**NITRO allows you to get warnings in an operation by specifying the "warning" query parameter as "yes". For example, to get warnings while connecting to the Citrix ADC appliance, the URL is as follows:

  http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/login?warning=yes

If any, the warnings are displayed in the response payload with the HTTP code "209 X-NITRO-WARNING".

- **Authenticated access for individual NITRO operations:**NITRO allows you to logon to the Citrix ADC appliance to perform individual operations. You can use this option instead of creating a NITRO session (using the login object) and then using that session to perform all operations,

  To do this, you must specify the username and password in the request header of the NITRO request as follows:

  X-NITRO-USER:<username>

  X-NITRO-PASS:<password>

  Note: In such cases, make sure that the request header DOES not include the following:

  Cookie:NITRO_AUTH_TOKEN=<tokenvalue>

`*Note:`*

Mandatory parameters are marked in redand placeholder content is marked in <green>.

update

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/nsratecontrol

HTTP Method:PUT

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Content-Type:application/json

Request Payload:

```
{"nsratecontrol":{

"tcpthreshold":<Double_value>,

"udpthreshold":<Double_value>,

"icmpthreshold":<Double_value>,
```

```
"tcprstthreshold":<Double_value>

}}
```

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error

unset

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/nsratecontrol?**action=unset**

HTTP Method:POST

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Content-Type:application/json

Request Payload:

```
{"nsratecontrol":{

"tcpthreshold":true,

"udpthreshold":true,

"icmpthreshold":true,

"tcprstthreshold":true

}}
```

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error

get (all)

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/nsratecontrol

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "nsratecontrol": [ {

"tcpthreshold":<Double_value>,

"udpthreshold":<Double_value>,

"icmpthreshold":<Double_value>,

"tcprstthreshold":<Double_value>

}]}
```

nsrollbackcmd

Configuration for Generate rollback commands resource.

Properties

(click to see Operations)

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
filename	<String>	Read-write	File that contains the commands for which the rollback commands must be generated. Specify the full path of the file name.
outtype	<String>	Read-write	Format in which the rollback commands must be generated. Possible values = cli, xml

Operations

(click to see Properties)

GET (ALL)

Some options that you can use for each operations:

- **Getting warnings in response:**NITRO allows you to get warnings in an operation by specifying the "warning" query parameter as "yes". For example, to get warnings while connecting to the Citrix ADC appliance, the URL is as follows:

  http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/login?warning=yes

  If any, the warnings are displayed in the response payload with the HTTP code "209 X-NITRO-WARNING".

- **Authenticated access for individual NITRO operations:**NITRO allows you to logon to the Citrix ADC appliance to perform individual operations. You can use this option instead of creating a NITRO session (using the login object) and then using that session to perform all operations,

  To do this, you must specify the username and password in the request header of the NITRO request as follows:

  X-NITRO-USER:<username>

  X-NITRO-PASS:<password>

Note: In such cases, make sure that the request header DOES not include the following:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue>

`*Note:` *

Mandatory parameters are marked in redand placeholder content is marked in <green>.

get (all)

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/nsrollbackcmd

Query-parameters:

args

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/nsrollbackcmd?
args=filename:<String_value>,outtype:<String_value>

Use this query-parameter to get nsrollbackcmd resources based on additional properties.

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "nsrollbackcmd": [ {

filename:<String_value>,outtype:<String_value>

}]}
```

nsrpcnode

Configuration for rpc node resource.

Properties

(click to see Operations)

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
ipaddress	<String>	Read-write	IP address of the node. This has to be in the same subnet as the NSIP address. Minimum length = 1
password	<String>	Read-write	Password to be used in authentication with the peer system node. Minimum length = 1
srcip	<String>	Read-write	Source IP address to be used to communicate with the peer system node. The default value is 0, which means that the appliance uses the NSIP address as the source IP address.
secure	<String>	Read-write	State of the channel when talking to the node. Possible values = YES, NO
__count	<Double>	Read-only	count parameter

Operations

(click to see Properties)

UPDATE| UNSET| GET (ALL)| GET| COUNT

Some options that you can use for each operations:

- **Getting warnings in response:**NITRO allows you to get warnings in an operation by specifying the "warning" query parameter as "yes". For example, to get warnings while connecting to the Citrix ADC appliance, the URL is as follows:

  http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/login?warning=yes

  If any, the warnings are displayed in the response payload with the HTTP code "209 X-NITRO-WARNING".

- **Authenticated access for individual NITRO operations:**NITRO allows you to logon to the Citrix ADC appliance to perform individual operations. You can use this option instead of creating a NITRO session (using the login object) and then using that session to perform all operations,

  To do this, you must specify the username and password in the request header of the NITRO request as follows:

  X-NITRO-USER:<username>

  X-NITRO-PASS:<password>

  Note: In such cases, make sure that the request header DOES not include the following:

  Cookie:NITRO_AUTH_TOKEN=<tokenvalue>

`*Note:`*

Mandatory parameters are marked in redand placeholder content is marked in <green>.

update

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/nsrpcnode

HTTP Method:PUT

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Content-Type:application/json

Request Payload:

```
{"nsrpcnode":{

<b>"ipaddress":<String_value>,

</b>"password":<String_value>,

"srcip":<String_value>,

"secure":<String_value>

}}
```

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error

unset

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/nsrpcnode?**action=unset**

HTTP Method:POST

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Content-Type:application/json

Request Payload:

```
{"nsrpcnode":{

<b>"ipaddress":<String_value>,

</b>"password":true,

"srcip":true,
```

```
"secure":true

}}
```

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error

get (all)

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/nsrpcnode

Query-parameters:

attrs

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/nsrpcnode?**attrs=property-name1,property-name2**

Use this query parameter to specify the resource details that you want to retrieve.

filter

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/nsrpcnode?**filter=property-name1:property-val1,property-name2:property-val2**

Use this query-parameter to get the filtered set of nsrpcnode resources configured on Citrix ADC. Filtering can be done on any of the properties of the resource.

view

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/nsrpcnode?**view=summary**

Note: By default, the retrieved results are displayed in detail view (?view=detail).

pagination

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/nsrpcnode?**pagesize=#no;pageno=#no**

Use this query-parameter to get the nsrpcnode resources in chunks.

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "nsrpcnode": [ {

"ipaddress":<String_value>,

"password":<String_value>,

"srcip":<String_value>,

"secure":<String_value>

}]}
```

get

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/nsrpcnode/ipaddress_value<String>

Query-parameters:

attrs

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/nsrpcnode/ipaddress_value<String>?**attrs=property-name1,property-name2**

Use this query parameter to specify the resource details that you want to retrieve.

view

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/nsrpcnode/ipaddress_value<String>?**view=summary**

Note: By default, the retrieved results are displayed in detail view (?view=detail).

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "nsrpcnode": [ {

"ipaddress":<String_value>,

"password":<String_value>,

"srcip":<String_value>,

"secure":<String_value>

}]}
```

count

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/nsrpcnode?**count=yes**

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "nsrpcnode": [ { "__count": "#no"} ] }
```

nsrunningconfig

Configuration for running configuration resource.

Properties

(click to see Operations)

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
withdefaults	<Boolean>	Read-write	Include default values of parameters that have not been explicitly configured. If this argument is disabled, such parameters are not included.
response	<String>	Read-only	running config data as text blob.

Operations

(click to see Properties)

GET (ALL)

Some options that you can use for each operations:

- **Getting warnings in response:**NITRO allows you to get warnings in an operation by specifying the "warning" query parameter as "yes". For example, to get warnings while connecting to the Citrix ADC appliance, the URL is as follows:

  http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/login?warning=yes

  If any, the warnings are displayed in the response payload with the HTTP code "209 X-NITRO-WARNING".

- **Authenticated access for individual NITRO operations:**NITRO allows you to logon to the Citrix ADC appliance to perform individual operations. You can use this option instead of creating a NITRO session (using the login object) and then using that session to perform all operations,

  To do this, you must specify the username and password in the request header of the NITRO request as follows:

  X-NITRO-USER:<username>

  X-NITRO-PASS:<password>

  Note: In such cases, make sure that the request header DOES not include the following:

  Cookie:NITRO_AUTH_TOKEN=<tokenvalue>

`*Note:`*

Mandatory parameters are marked in redand placeholder content is marked in <green>.

get (all)

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/nsrunningconfig

Query-parameters:

args

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/nsrunningconfig?**args=withdefaults:<Boolean_value>**

Use this query-parameter to get nsrunningconfig resources based on additional properties.

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "nsrunningconfig": [ {

withdefaults:<Boolean_value>"response":<String_value>

}]}
```

nssavedconfig

Configuration for last saved configuration resource.

Properties

(click to see Operations)

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
textblob	<String>	Read-only	Text of the last saved configuration.

Operations

(click to see Properties)

GET (ALL)

Some options that you can use for each operations:

- **Getting warnings in response:**NITRO allows you to get warnings in an operation by specifying the "warning" query parameter as "yes". For example, to get warnings while connecting to the Citrix ADC appliance, the URL is as follows:

  http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/login?warning=yes

  If any, the warnings are displayed in the response payload with the HTTP code "209 X-NITRO-WARNING".

- **Authenticated access for individual NITRO operations:**NITRO allows you to logon to the Citrix ADC appliance to perform individual operations. You can use this option instead of creating a NITRO session (using the login object) and then using that session to perform all operations,

  To do this, you must specify the username and password in the request header of the NITRO request as follows:

  X-NITRO-USER:<username>

  X-NITRO-PASS:<password>

  Note: In such cases, make sure that the request header DOES not include the following:

  Cookie:NITRO_AUTH_TOKEN=<tokenvalue>

`*Note:`*

Mandatory parameters are marked in redand placeholder content is marked in <green>.

get (all)

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/nssavedconfig

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "nssavedconfig": [ {

"textblob":<String_value>

}]}
```

nsservicefunction

Configuration for service Function resource.

Properties

(click to see Operations)

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
servicefunctionname	<String>	Read-write	Name of the service function to be created. Leading character must be a number or letter. Other characters allowed, after the first character, are @ _ - . (period) : (colon) # and space ( ). Minimum length = 1
ingressvlan	<Double>	Read-write	VLAN ID on which the traffic from service function reaches Citrix ADC. Minimum value = 1 Maximum value = 4094

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
__count	<Double>	Read-only	count parameter

Operations

(click to see Properties)

ADD| UPDATE| DELETE| GET (ALL)| GET| COUNT

Some options that you can use for each operations:

- **Getting warnings in response:**NITRO allows you to get warnings in an operation by specifying the "warning" query parameter as "yes". For example, to get warnings while connecting to the Citrix ADC appliance, the URL is as follows:

  http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/login?warning=yes

  If any, the warnings are displayed in the response payload with the HTTP code "209 X-NITRO-WARNING".
- **Authenticated access for individual NITRO operations:**NITRO allows you to logon to the Citrix ADC appliance to perform individual operations. You can use this option instead of creating a NITRO session (using the login object) and then using that session to perform all operations,

  To do this, you must specify the username and password in the request header of the NITRO request as follows:

  X-NITRO-USER:<username>

  X-NITRO-PASS:<password>

  Note: In such cases, make sure that the request header DOES not include the following:

  Cookie:NITRO_AUTH_TOKEN=<tokenvalue>

`*Note:`*

Mandatory parameters are marked in redand placeholder content is marked in <green>.

add

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/nsservicefunction

HTTP Method:POST

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Content-Type:application/json

Request Payload:

```
{"nsservicefunction":{

<b>"servicefunctionname":<String_value>,

</b><b>"ingressvlan":<Double_value>

</b>}}
```

Response:

HTTP Status Code on Success: 201 Created HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error

update

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/nsservicefunction

HTTP Method:PUT

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Content-Type:application/json

Request Payload:

```
{"nsservicefunction":{

<b>"servicefunctionname":<String_value>,

</b>"ingressvlan":<Double_value>

}}
```

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error

delete

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/nsservicefunction/servicefunctionname_value<String>

HTTP Method:DELETE

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue>

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error

get (all)

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/nsservicefunction

Query-parameters:

attrs

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/nsservicefunction?**attrs=property-name1,property-name2**

Use this query parameter to specify the resource details that you want to retrieve.

filter

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/nsservicefunction?**filter=property-name1:property-val1,property-name2:property-val2**

Use this query-parameter to get the filtered set of nsservicefunction resources configured on Citrix ADC. Filtering can be done on any of the properties of the resource.

view

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/nsservicefunction?**view=summary**

Note: By default, the retrieved results are displayed in detail view (?view=detail).

pagination

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/nsservicefunction?**pagesize=#no;pageno=#no**

Use this query-parameter to get the nsservicefunction resources in chunks.

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "nsservicefunction": [ {

"servicefunctionname":<String_value>,

"ingressvlan":<Double_value>

}]}
```

get

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/nsservicefunction/servicefunctionname_value<String>

Query-parameters:

attrs

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/nsservicefunction/servicefunctionname_value<String>?**attrs=property-name1,property-name2**

Use this query parameter to specify the resource details that you want to retrieve.

view

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/nsservicefunction/servicefunctionname_value<String>?**view=summary**

Note: By default, the retrieved results are displayed in detail view (?view=detail).

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "nsservicefunction": [ {

"servicefunctionname":<String_value>,

"ingressvlan":<Double_value>

}]}
```

count

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/nservicefunction?**count=yes**

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "nservicefunction": [ { "__count": "#no"} ] }
```

nservicepath

Configuration for service Chain resource.

Properties

(click to see Operations)

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
servicepathname	<String>	Read-write	Name for the Service path. Must begin with an ASCII alphanumeric or underscore (_) character, and must contain only ASCII alphanumeric, underscore, hash

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
			(#), period (.), space, colon (:), at (@), equals (=), and hyphen (-) characters. Minimum length = 1
__count	<Double>	Read-only	count parameter

Operations

(click to see Properties)

ADD| DELETE| GET (ALL)| GET| COUNT

Some options that you can use for each operations:

- **Getting warnings in response:**NITRO allows you to get warnings in an operation by specifying the "warning" query parameter as "yes". For example, to get warnings while connecting to the Citrix ADC appliance, the URL is as follows:

  http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/login?warning=yes

  If any, the warnings are displayed in the response payload with the HTTP code "209 X-NITRO-WARNING".

- **Authenticated access for individual NITRO operations:**NITRO allows you to logon to the Citrix ADC appliance to perform individual operations. You can use this option instead of creating a NITRO session (using the login object) and then using that session to perform all operations,

  To do this, you must specify the username and password in the request header of the NITRO request as follows:

  X-NITRO-USER:<username>

  X-NITRO-PASS:<password>

  Note: In such cases, make sure that the request header DOES not include the following:

  Cookie:NITRO_AUTH_TOKEN=<tokenvalue>

`*Note:` *

Mandatory parameters are marked in redand placeholder content is marked in <green>.

add

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/nsservicepath

HTTP Method:POST

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Content-Type:application/json

Request Payload:

```
{"nsservicepath":{

<b>"servicepathname":<String_value>

</b>}}
```

Response:

HTTP Status Code on Success: 201 Created HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error

delete

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/nsservicepath/servicepathname_value<String>

HTTP Method:DELETE

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue>

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error

get (all)

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/nsservicepath

Query-parameters:

attrs

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/nsservicepath?**attrs=property-name1,property-name2**

Use this query parameter to specify the resource details that you want to retrieve.

filter

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/nsservicepath?**filter=property-name1:property-val1,property-name2:property-val2**

Use this query-parameter to get the filtered set of nsservicepath resources configured on Citrix ADC. Filtering can be done on any of the properties of the resource.

view

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/nsservicepath?**view=summary**

Note: By default, the retrieved results are displayed in detail view (?view=detail).

pagination

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/nsservicepath?**pagesize=#no;pageno=#no**

Use this query-parameter to get the nsservicepath resources in chunks.

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "nsservicepath": [ {

"servicepathname":<String_value>,



}]}
```

get

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/nsservicepath/servicepathname_value<String>

Query-parameters:

attrs

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/nsservicepath/servicepathname_value<String>?**attrs=property-name1,property-name2**

Use this query parameter to specify the resource details that you want to retrieve.

view

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/nsservicepath/servicepathname_value<String>?**view=summary**

Note: By default, the retrieved results are displayed in detail view (?view=detail).

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "nsservicepath": [ {

"servicepathname":<String_value>,



}]}
```

count

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/nsservicepath?**count=yes**

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "nsservicepath": [ { "__count": "#no"} ] }
```

nsservicepath_binding

Binding object which returns the resources bound to nsservicepath.

Properties

(click to see Operations)

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
servicepathname	<String>	Read-write	Name for the Service path. Must begin with an ASCII alphanumeric or underscore (_) character, and must contain only ASCII alphanumeric, underscore, hash (#), period (.), space, colon (:), at (@), equals (=), and hyphen (-) characters. Minimum length = 1
nsservicepath_nsservicefunction_binding	<nsservicepath_nsservicefunction_binding []>	Read-only	nsservicefunction that can be bound to nsservicepath.

Operations

(click to see Properties)

GET| GET (ALL)

Some options that you can use for each operations:

- **Getting warnings in response:**NITRO allows you to get warnings in an operation by specifying the "warning" query parameter as "yes". For example, to get warnings while connecting to the Citrix ADC appliance, the URL is as follows:

  http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/login?warning=yes

  If any, the warnings are displayed in the response payload with the HTTP code "209 X-NITRO-WARNING".

- **Authenticated access for individual NITRO operations:**NITRO allows you to logon to the Citrix ADC appliance to perform individual operations. You can use this option instead of creating a NITRO session (using the login object) and then using that session to perform all operations,

  To do this, you must specify the username and password in the request header of the NITRO request as follows:

  X-NITRO-USER:<username>

  X-NITRO-PASS:<password>

  Note: In such cases, make sure that the request header DOES not include the following:

  Cookie:NITRO_AUTH_TOKEN=<tokenvalue>

`*Note:`*

Mandatory parameters are marked in redand placeholder content is marked in <green>.

get

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/nsservicepath_binding/servicepathname_value<String>

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "nsservicepath_binding": [ {

"servicepathname":<String_value>,

"nsservicepath_nsservicefunction_binding":<nsservicepath_nsservicefunction
_binding[]_value>

}]}
```

get (all)

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/nsservicepath_binding

Query-parameters:

bulkbindings

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/nsservicepath_binding?**bulkbindings=yes**

NITRO allows you to fetch bindings in bulk.

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "nsservicepath_binding": [ {

"servicepathname":<String_value>,

"nsservicepath_nsservicefunction_binding":<nsservicepath_nsservicefunction
_binding[]_value>

}]}
```

nsservicepath_nsservicefunction_binding

Binding object showing the nsservicefunction that can be bound to nsservicepath.

Properties

(click to see Operations)

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
servicepathname	<String>	Read-write	Name for the Service path. Must begin with an ASCII alphanumeric or underscore (_) character, and must contain only ASCII alphanumeric, underscore, hash (#), period (.), space, colon (:), at (@), equals (=), and hyphen (-) characters. Minimum length = 1

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
index	<Double>	Read-write	The serviceindex of each servicefunction in path. Minimum value = 1
servicefunction	<String>	Read-write	List of service functions constituting the chain. Minimum length = 1
__count	<Double>	Read-write	count parameter

Operations

(click to see Properties)

ADD:| DELETE:| GET| GET (ALL)| COUNT

Some options that you can use for each operations:

- **Getting warnings in response:**NITRO allows you to get warnings in an operation by specifying the "warning" query parameter as "yes". For example, to get warnings while connecting to the Citrix ADC appliance, the URL is as follows:

  http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/login?warning=yes

  If any, the warnings are displayed in the response payload with the HTTP code "209 X-NITRO-WARNING".

- **Authenticated access for individual NITRO operations:**NITRO allows you to logon to the Citrix ADC appliance to perform individual operations. You can use this option instead of creating a NITRO session (using the login object) and then using that session to perform all operations,

  To do this, you must specify the username and password in the request header of the NITRO request as follows:

  X-NITRO-USER:<username>

  X-NITRO-PASS:<password>

  Note: In such cases, make sure that the request header DOES not include the following:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue>

`*Note:` *

Mandatory parameters are marked in redand placeholder content is marked in <green>.

add

URL:http://<Citrix-ADC-IP-address(NSIP)/nitro/v1/config/nsservicepath_nsservicefunction_binding

HTTP Method:PUT

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Content-Type:application/json

Request Payload:

```
{

"nsservicepath_nsservicefunction_binding":{

<b>"servicepathname":<String_value>,

</b><b>"servicefunction":<String_value>,

</b><b>"index":<Double_value></b>

}}
```

Response:

HTTP Status Code on Success: 201 Created HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error

delete

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/nsservicepath_nsservicefunction_binding/servicepathname_value<String>

Query-parameters:

args

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/nsservicepath_nsservicefunction_binding/servicepathname_value<String>?**args=servicefunction:<String_value>**

HTTP Method:DELETE

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue>

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error

get

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/nsservicepath_nsservicefunction_binding/servicepathname_value<String>

Query-parameters:

filter

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/nsservicepath_nsservicefunction_binding/servicepathname_value<String>?**filter=property-name1:property-value1,property-name2:property-value2**

Use this query-parameter to get the filtered set of nsservicepath_nsservicefunction_binding resources configured on Citrix ADC. Filtering can be done on any of the properties of the resource.

pagination

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/nsservicepath_nsservicefunction_binding/servicepathname_value<String>?**pagesize=#no;pageno=#no**

Use this query-parameter to get the nsservicepath_nsservicefunction_binding resources in chunks.

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "nsservicepath_nsservicefunction_binding": [ {

"servicepathname":<String_value>,

"index":<Double_value>,

"servicefunction":<String_value>

}]}
```

get (all)

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/nsservicepath_nsservicefunction_binding

Query-parameters:

bulkbindings

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/nsservicepath_nsservicefunction_binding?**bulkbindings=yes**

NITRO allows you to fetch bindings in bulk.

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "nsservicepath_nsservicefunction_binding": [ {

"servicepathname":<String_value>,

"index":<Double_value>,

"servicefunction":<String_value>

}]}
```

count

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/nsservicepath_nsservicefunction_binding/servicepathname_value<String>?**count=yes**

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{"nsservicepath_nsservicefunction_binding": [ { "__count": "#no"} ] }
```

nssimpleacl

Configuration for simple ACL resource.

Properties

(click to see Operations)

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
aclname	<String>	Read-write	Name for the simple ACL rule. Must begin with an ASCII alphabetic or underscore (_) character, and must contain only ASCII alphanumeric, underscore, hash (#), period (.), space, colon (:), at (@), equals (=), and hyphen (-) characters. Cannot be changed after the simple ACL rule is created. Minimum length = 1
aclaction	<String>	Read-write	Drop incoming IPv4 packets that match the simple ACL rule. Possible values = DENY
td	<Double>	Read-write	Integer value that uniquely identifies

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
			the traffic domain in which you want to configure the entity. If you do not specify an ID, the entity becomes part of the default traffic domain, which has an ID of 0. Minimum value = 0 Maximum value = 4094
srcip	<String>	Read-write	IP address to match against the source IP address of an incoming IPv4 packet.
destport	<Integer>	Read-write	Port number to match against the destination port number of an incoming IPv4 packet. DestPort is mandatory while setting Protocol. Omitting the port number and protocol creates an all-ports and all protocols simple ACL rule, which matches any port and any protocol. In that case, you cannot create

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
			another simple ACL rule specifying a specific port and the same source IPv4 address. Minimum value = 1 Maximum value = 65535
protocol	<String>	Read-write	Protocol to match against the protocol of an incoming IPv4 packet. You must set this parameter if you have set the Destination Port parameter. Possible values = TCP, UDP
ttl	<Double>	Read-write	Number of seconds, in multiples of four, after which the simple ACL rule expires. If you do not want the simple ACL rule to expire, do not specify a TTL value. Minimum value = 4 Maximum value = 2147483647
estsessions	<Boolean>	Read-write	.
hits	<Double>	Read-only	Number of hits for this ACL rule.

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
__count	<Double>	Read-only	count parameter

Operations

(click to see Properties)

ADD| CLEAR| DELETE| FLUSH| GET (ALL)| GET| COUNT

Some options that you can use for each operations:

- **Getting warnings in response:**NITRO allows you to get warnings in an operation by specifying the "warning" query parameter as "yes". For example, to get warnings while connecting to the Citrix ADC appliance, the URL is as follows:

  http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/login?warning=yes

  If any, the warnings are displayed in the response payload with the HTTP code "209 X-NITRO-WARNING".
- **Authenticated access for individual NITRO operations:**NITRO allows you to logon to the Citrix ADC appliance to perform individual operations. You can use this option instead of creating a NITRO session (using the login object) and then using that session to perform all operations,

  To do this, you must specify the username and password in the request header of the NITRO request as follows:

  X-NITRO-USER:<username>

  X-NITRO-PASS:<password>

  Note: In such cases, make sure that the request header DOES not include the following:

  Cookie:NITRO_AUTH_TOKEN=<tokenvalue>

`*Note:`*

Mandatory parameters are marked in redand placeholder content is marked in <green>.

add

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/nssimpleacl

HTTP Method:POST

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Content-Type:application/json

Request Payload:

```
{"nssimpleacl":{

<b>"aclname":<String_value>,

</b><b>"aclaction":<String_value>,

</b>"td":<Double_value>,

<b>"srcip":<String_value>,

</b>"destport":<Integer_value>,

"protocol":<String_value>,

"ttl":<Double_value>

}}
```

Response:

HTTP Status Code on Success: 201 Created HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error

clear

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/nssimpleacl?**action=clear**

HTTP Method:POST

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Content-Type:application/json

Request Payload:

```
{"nssimpleacl":{

}}
```

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error

delete

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/nssimpleacl/aclname_value<String>

HTTP Method:DELETE

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue>

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error

flush

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/nssimpleacl?**action=flush**

HTTP Method:POST

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Content-Type:application/json

Request Payload:

```
{"nssimpleacl":{
```

```
<b>"estsessions":<Boolean_value>

</b>}}
```

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error

get (all)

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/nssimpleacl

Query-parameters:

attrs

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/nssimpleacl?**attrs=property-name1,property-name2**

Use this query parameter to specify the resource details that you want to retrieve.

filter

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/nssimpleacl?**filter=property-name1:property-val1,property-name2:property-val2**

Use this query-parameter to get the filtered set of nssimpleacl resources configured on Citrix ADC. Filtering can be done on any of the properties of the resource.

view

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/nssimpleacl?**view=summary**

Note: By default, the retrieved results are displayed in detail view (?view=detail).

pagination

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/nssimpleacl?**pagesize=#no;pageno=#no**

Use this query-parameter to get the nssimpleacl resources in chunks.

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "nssimpleacl": [ {

"aclname":<String_value>,

"aclaction":<String_value>,

"td":<Double_value>,

"srcip":<String_value>,

"destport":<Integer_value>,

"protocol":<String_value>,

"ttl":<Double_value>,

"hits":<Double_value>

}]}
```

get

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/nssimpleacl/aclname_value<String>

Query-parameters:

attrs

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/nssimpleacl/aclname_value<String>?**attrs=property-name1,property-name2**

Use this query parameter to specify the resource details that you want to retrieve.

view

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/nssimpleacl/aclname_value<String>?**view=summary**

Note: By default, the retrieved results are displayed in detail view (?view=detail).

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "nssimpleacl": [ {

"aclname":<String_value>,

"aclaction":<String_value>,

"td":<Double_value>,

"srcip":<String_value>,

"destport":<Integer_value>,

"protocol":<String_value>,
```

```
"ttl":<Double_value>,

"hits":<Double_value>

}]}
```

count

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/nssimpleacl?**count=yes**

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "nssimpleacl": [ { "__count": "#no"} ] }
```

nssimpleacl6

Configuration for simple ACL6 resource.

Properties

(click to see Operations)

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
aclname	<String>	Read-write	Name for the simple ACL6 rule. Must begin with an ASCII alphabetic or underscore (_) character, and must contain only ASCII alphanumeric, underscore, hash (#), period (.), space, colon (:), at (@), equals (=), and hyphen (-) characters. Cannot be changed after the simple ACL6 rule is created. Minimum length = 1
td	<Double>	Read-write	Integer value that uniquely identifies the traffic domain in which you want to configure the entity. If you do not specify an ID, the entity becomes part of the default traffic domain, which has an ID of 0. Minimum value = 0 Maximum value = 4094
aclaction	<String>	Read-write	Drop incoming IPv6 packets that match the simple ACL6 rule.

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
			Possible values = DENY
srcipv6	<String>	Read-write	IP address to match against the source IP address of an incoming IPv6 packet.
destport	<Integer>	Read-write	Port number to match against the destination port number of an incoming IPv6 packet. DestPort is mandatory while setting Protocol. Omitting the port number and protocol creates an all-ports and all protocol simple ACL6 rule, which matches any port and any protocol. In that case, you cannot create another simple ACL6 rule specifying a specific port and the same source IPv6 address. Minimum value = 1

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
			Maximum value = 65535
protocol	<String>	Read-write	Protocol to match against the protocol of an incoming IPv6 packet. You must set this parameter if you set the Destination Port parameter. Possible values = TCP, UDP
ttl	<Double>	Read-write	Number of seconds, in multiples of four, after which the simple ACL6 rule expires. If you do not want the simple ACL6 rule to expire, do not specify a TTL value. Minimum value = 4 Maximum value = 2147483647
estsessions	<Boolean>	Read-write	.
hits	<Double>	Read-only	Number of hits for this SACL6 rule.
__count	<Double>	Read-only	count parameter

Operations

(click to see Properties)

ADD| CLEAR| FLUSH| DELETE| GET (ALL)| GET| COUNT

Some options that you can use for each operations:

- **Getting warnings in response:**NITRO allows you to get warnings in an operation by specifying the "warning" query parameter as "yes". For example, to get warnings while connecting to the Citrix ADC appliance, the URL is as follows:

  http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/login?warning=yes

  If any, the warnings are displayed in the response payload with the HTTP code "209 X-NITRO-WARNING".

- **Authenticated access for individual NITRO operations:**NITRO allows you to logon to the Citrix ADC appliance to perform individual operations. You can use this option instead of creating a NITRO session (using the login object) and then using that session to perform all operations,

  To do this, you must specify the username and password in the request header of the NITRO request as follows:

  X-NITRO-USER:<username>

  X-NITRO-PASS:<password>

  Note: In such cases, make sure that the request header DOES not include the following:

  Cookie:NITRO_AUTH_TOKEN=<tokenvalue>

`*Note:`*

Mandatory parameters are marked in redand placeholder content is marked in <green>.

add

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/nssimpleacl6

HTTP Method:POST

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Content-Type:application/json

Request Payload:

```
{"nssimpleacl6":{
```

```
<b>"aclname":<String_value>,

</b>"td":<Double_value>,

<b>"aclaction":<String_value>,

</b><b>"srcipv6":<String_value>,

</b>"destport":<Integer_value>,

"protocol":<String_value>,

"ttl":<Double_value>

}}
```

Response:

HTTP Status Code on Success: 201 Created HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error

clear

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/nssimpleacl6?**action=clear**

HTTP Method:POST

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Content-Type:application/json

Request Payload:

```
{"nssimpleacl6":{

}}
```

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error

flush

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/nssimpleacl6?**action=flush**

HTTP Method:POST

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Content-Type:application/json

Request Payload:

```
{"nssimpleacl6":{

<b>"estsessions":<Boolean_value>

</b>}}
```

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error

delete

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/nssimpleacl6/aclname_value<String>

HTTP Method:DELETE

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue>

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error

get (all)

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/nssimpleacl6

Query-parameters:

attrs

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/nssimpleacl6?**attrs=property-name1,property-name2**

Use this query parameter to specify the resource details that you want to retrieve.

filter

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/nssimpleacl6?**filter=property-name1:property-val1,property-name2:property-val2**

Use this query-parameter to get the filtered set of nssimpleacl6 resources configured on Citrix ADC. Filtering can be done on any of the properties of the resource.

view

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/nssimpleacl6?**view=summary**

Note: By default, the retrieved results are displayed in detail view (?view=detail).

pagination

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/nssimpleacl6?**pagesize=#no;pageno=#no**

Use this query-parameter to get the nssimpleacl6 resources in chunks.

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "nssimpleacl6": [ {

"aclname":<String_value>,

"aclaction":<String_value>,

"td":<Double_value>,

"srcipv6":<String_value>,

"destport":<Integer_value>,

"protocol":<String_value>,

"ttl":<Double_value>,

"hits":<Double_value>

}]}
```

get

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/nssimpleacl6/aclname_value<String>

Query-parameters:

attrs

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/nssimpleacl6/aclname_value<String>?**attrs=property-name1,property-name2**

Use this query parameter to specify the resource details that you want to retrieve.

view

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/nssimpleacl6/aclname_value<String>?**view=summary**

Note: By default, the retrieved results are displayed in detail view (?view=detail).

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "nssimpleacl6": [ {

"aclname":<String_value>,

"aclaction":<String_value>,

"td":<Double_value>,

"srcipv6":<String_value>,

"destport":<Integer_value>,

"protocol":<String_value>,

"ttl":<Double_value>,

"hits":<Double_value>

}]}
```

count

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/nssimpleacl6?**count=yes**

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "nssimpleacl6": [ { "__count": "#no"} ] }
```

nssourceroutecachetable

Configuration for Source IP Mac Cache Table. resource.

Properties

(click to see Operations)

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
sourceip	<String>	Read-only	Source ip of the connection.
sourcemac	<String>	Read-only	source MAC address of an incoming IPv6 packet.

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
vlan	<Double>	Read-only	ID of the VLAN.
Interface	<String>	Read-only	ID of an interface.
__count	<Double>	Read-only	count parameter

Operations

(click to see Properties)

FLUSH| GET (ALL)| COUNT

Some options that you can use for each operations:

- **Getting warnings in response:**NITRO allows you to get warnings in an operation by specifying the "warning" query parameter as "yes". For example, to get warnings while connecting to the Citrix ADC appliance, the URL is as follows:

  http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/login?warning=yes

  If any, the warnings are displayed in the response payload with the HTTP code "209 X-NITRO-WARNING".

- **Authenticated access for individual NITRO operations:**NITRO allows you to logon to the Citrix ADC appliance to perform individual operations. You can use this option instead of creating a NITRO session (using the login object) and then using that session to perform all operations,

  To do this, you must specify the username and password in the request header of the NITRO request as follows:

  X-NITRO-USER:<username>

  X-NITRO-PASS:<password>

  Note: In such cases, make sure that the request header DOES not include the following:

  Cookie:NITRO_AUTH_TOKEN=<tokenvalue>

`*Note:`*

Mandatory parameters are marked in redand placeholder content is marked in <green>.

flush

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/nssourceroutecachetable?**action=flush**

HTTP Method:POST

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Content-Type:application/json

Request Payload:

```
{"nssourceroutecachetable":{

}}
```

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error

get (all)

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/nssourceroutecachetable

Query-parameters:

attrs

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/nssourceroutecachetable?**attrs=property-name1,property-name2**

Use this query parameter to specify the resource details that you want to retrieve.

filter

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/nssourceroutecachetable?**filter=property-name1:property-val1,property-name2:property-val2**

Use this query-parameter to get the filtered set of nssourceroutecachetable resources configured on Citrix ADC. Filtering can be done on any of the properties of the resource.

view

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/nssourceroutecachetable?**view=summary**

Note: By default, the retrieved results are displayed in detail view (?view=detail).

pagination

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/nssourceroutecachetable?**pagesize=#no;pageno=#no**

Use this query-parameter to get the nssourceroutecachetable resources in chunks.

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "nssourceroutecachetable": [ {

"sourceip":<String_value>,

"sourcemac":<String_value>,

"vlan":<Double_value>,

"Interface":<String_value>

}]}
```

count

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/nssourceroutecachetable?**count=yes**

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "nssourceroutecachetable": [ { "__count": "#no"} ] }
```

nsspparams

Configuration for Surge Protection parameter resource.

Properties

(click to see Operations)

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
basethreshold	<Integer>	Read-write	Maximum number of server connections that can be opened before surge protection is activated. Default value: 200

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
			Minimum value = 0 Maximum value = 32767
throttle	<String>	Read-write	Rate at which the system opens connections to the server. Default value: Normal Possible values = Aggressive, Normal, Relaxed
table0	<Integer[]>	Read-only	Table.

Operations

(click to see Properties)

UPDATE| UNSET| GET (ALL)

Some options that you can use for each operations:

- **Getting warnings in response:**NITRO allows you to get warnings in an operation by specifying the "warning" query parameter as "yes". For example, to get warnings while connecting to the Citrix ADC appliance, the URL is as follows:

  http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/login?warning=yes

  If any, the warnings are displayed in the response payload with the HTTP code "209 X-NITRO-WARNING".

- **Authenticated access for individual NITRO operations:**NITRO allows you to logon to the Citrix ADC appliance to perform individual operations. You can use this option instead of creating a NITRO session (using the login object) and then using that session to perform all operations,

  To do this, you must specify the username and password in the request header of the NITRO request as follows:

X-NITRO-USER:<username>

X-NITRO-PASS:<password>

Note: In such cases, make sure that the request header DOES not include the following:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue>

`*Note:`*

Mandatory parameters are marked in redand placeholder content is marked in <green>.

update

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/nsspparams

HTTP Method:PUT

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Content-Type:application/json

Request Payload:

```
{"nsspparams":{

"basethreshold":<Integer_value>,

"throttle":<String_value>

}}
```

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error

unset

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/nsspparams?**action=unset**

HTTP Method:POST

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Content-Type:application/json

Request Payload:

```
{"nsspparams":{

"basethreshold":true,

"throttle":true

}}
```

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error

get (all)

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/nsspparams

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "nsspparams": [ {
"basethreshold":<Integer_value>,
"throttle":<String_value>,
"table0":<Integer[]_value>
}]}
```

nsstats

Configuration for stats resource.

Properties

(click to see Operations)

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
cleanuplevel	<String>	Read-write	The level of stats to be cleared. 'global' option will clear global counters only, 'all' option will clear all device counters also along with global counters. For both the cases only 'ever incrementing counters' i.e. total counters will be cleared. Possible values = global, all

Operations

(click to see Properties)

CLEAR

Some options that you can use for each operations:

- **Getting warnings in response:**NITRO allows you to get warnings in an operation by specifying the "warning" query parameter as "yes". For example, to get warnings while connecting to the Citrix ADC appliance, the URL is as follows:

  http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/login?warning=yes

  If any, the warnings are displayed in the response payload with the HTTP code "209 X-NITRO-WARNING".

- **Authenticated access for individual NITRO operations:**NITRO allows you to logon to the Citrix ADC appliance to perform individual operations. You can use this option instead of creating a NITRO session (using the login object) and then using that session to perform all operations,

  To do this, you must specify the username and password in the request header of the NITRO request as follows:

  X-NITRO-USER:<username>

  X-NITRO-PASS:<password>

  Note: In such cases, make sure that the request header DOES not include the following:

  Cookie:NITRO_AUTH_TOKEN=<tokenvalue>

`*Note:`*

Mandatory parameters are marked in redand placeholder content is marked in <green>.

clear

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/nsstats?**action=clear**

HTTP Method:POST

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Content-Type:application/json

Request Payload:

```
{"nsstats":{

<b>"cleanuplevel":<String_value>

</b>}}
```

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error

nssurgeq

Configuration for surge queue resource.

Properties

(click to see Operations)

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
name	<String>	Read-write	Name of a virtual server, service or service group for which the SurgeQ must be flushed.
servername	<String>	Read-write	Name of a service group member. This argument is needed when you want to flush the SurgeQ of a service group.
port	<Integer>	Read-write	port on which server is bound to the entity(Servicegroup). Range 1 - 65535

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
			* in CLI is represented as 65535 in NITRO API

Operations

(click to see Properties)

FLUSH

Some options that you can use for each operations:

- **Getting warnings in response:**NITRO allows you to get warnings in an operation by specifying the "warning" query parameter as "yes". For example, to get warnings while connecting to the Citrix ADC appliance, the URL is as follows:

  http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/login?warning=yes

  If any, the warnings are displayed in the response payload with the HTTP code "209 X-NITRO-WARNING".
- **Authenticated access for individual NITRO operations:**NITRO allows you to logon to the Citrix ADC appliance to perform individual operations. You can use this option instead of creating a NITRO session (using the login object) and then using that session to perform all operations,

  To do this, you must specify the username and password in the request header of the NITRO request as follows:

  X-NITRO-USER:<username>

  X-NITRO-PASS:<password>

  Note: In such cases, make sure that the request header DOES not include the following:

  Cookie:NITRO_AUTH_TOKEN=<tokenvalue>

`*Note:`*

Mandatory parameters are marked in redand placeholder content is marked in <green>.

flush

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/nssurgeq?**action=flush**

HTTP Method:POST

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Content-Type:application/json

Request Payload:

```
{"nssurgeq":{

"name":<String_value>,

"servername":<String_value>,

"port":<Integer_value>

}}
```

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error

nstcpbufparam

Configuration for tcp buffer parameter resource.

Properties

(click to see Operations)

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
size	<Double>	Read-write	TCP buffering size per connection, in kilobytes.

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
			Default value: 64 Minimum value = 4 Maximum value = 20480
memlimit	<Double>	Read-write	Maximum memory, in megabytes, that can be used for buffering. Default value: 64

Operations

(click to see Properties)

UPDATE| UNSET| GET (ALL)

Some options that you can use for each operations:

- **Getting warnings in response:**NITRO allows you to get warnings in an operation by specifying the "warning" query parameter as "yes". For example, to get warnings while connecting to the Citrix ADC appliance, the URL is as follows:

  http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/login?warning=yes

  If any, the warnings are displayed in the response payload with the HTTP code "209 X-NITRO-WARNING".

- **Authenticated access for individual NITRO operations:**NITRO allows you to logon to the Citrix ADC appliance to perform individual operations. You can use this option instead of creating a NITRO session (using the login object) and then using that session to perform all operations,

  To do this, you must specify the username and password in the request header of the NITRO request as follows:

  X-NITRO-USER:<username>

  X-NITRO-PASS:<password>

Note: In such cases, make sure that the request header DOES not include the following:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue>

`*Note:`*

Mandatory parameters are marked in redand placeholder content is marked in <green>.

update

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/nstcpbufparam

HTTP Method:PUT

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Content-Type:application/json

Request Payload:

```
{"nstcpbufparam":{

"size":<Double_value>,

"memlimit":<Double_value>

}}
```

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error

unset

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/nstcpbufparam?**action=unset**

HTTP Method:POST

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Content-Type:application/json

Request Payload:

```
{"nstcpbufparam":{

"size":true,

"memlimit":true

}}
```

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error

get (all)

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/nstcpbufparam

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "nstcpbufparam": [ {

"size":<Double_value>,
```

```
"memlimit":<Double_value>

}]}
```

nstcpparam

Configuration for tcp parameters resource.

Properties

(click to see Operations)

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
ws	<String>	Read-write	Enable or disable window scaling. Default value: ENABLED Possible values = ENABLED, DISABLED
wsval	<Double>	Read-write	Factor used to calculate the new window size. This argument is needed only when the window scaling is enabled. Default value: 8 Minimum value = 0 Maximum value = 14
sack	<String>	Read-write	Enable or disable Selective

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
			ACKnowledgement (SACK). Default value: ENABLED Possible values = ENABLED, DISABLED
learnvsvrmss	<String>	Read-write	Enable or disable maximum segment size (MSS) learning for virtual servers. Default value: DISABLED Possible values = ENABLED, DISABLED
maxburst	<Double>	Read-write	Maximum number of TCP segments allowed in a burst. Default value: 6 Minimum value = 1 Maximum value = 255
initialcwnd	<Double>	Read-write	Initial maximum upper limit on the number of TCP packets that can be outstanding on the TCP link to the server. Default value: 10 Minimum value = 1

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
			Maximum value = 44
recvbuffsize	<Double>	Read-write	TCP Receive buffer size. Default value: 8190 Minimum value = 8190 Maximum value = 20971520
delayedack	<Double>	Read-write	Timeout for TCP delayed ACK, in milliseconds. Default value: 100 Minimum value = 10 Maximum value = 300
downstaterst	<String>	Read-write	Flag to switch on RST on down services. Default value: DISABLED Possible values = ENABLED, DISABLED
nagle	<String>	Read-write	Enable or disable the Nagle algorithm on TCP connections. Default value: DISABLED

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
			Possible values = ENABLED, DISABLED
limitedpersist	<String>	Read-write	Limit the number of persist (zero window) probes. Default value: ENABLED Possible values = ENABLED, DISABLED
oooqsize	<Double>	Read-write	Maximum size of out-of-order packets queue. A value of 0 means no limit. Default value: 300 Minimum value = 0 Maximum value = 65535
ackonpush	<String>	Read-write	Send immediate positive acknowledgement (ACK) on receipt of TCP packets with PUSH flag. Default value: ENABLED Possible values = ENABLED, DISABLED
maxpktpermss	<Double>	Read-write	Maximum number of TCP packets allowed

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
			per maximum segment size (MSS). Minimum value = 0 Maximum value = 1460
pktperretx	<Integer>	Read-write	Maximum limit on the number of packets that should be retransmitted on receiving a partial ACK. Default value: 1 Minimum value = 1 Maximum value = 100
minrto	<Integer>	Read-write	Minimum retransmission timeout, in milliseconds, specified in 10-millisecond increments (value must yield a whole number if divided by 10). Default value: 1000 Minimum value = 10 Maximum value = 64000
slowstartincr	<Integer>	Read-write	Multiplier that determines the rate at which slow start increases the size of

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
			the TCP transmission window after each acknowledgement of successful transmission. Default value: 2 Minimum value = 1 Maximum value = 100
maxdynserverprobes	<Double>	Read-write	Maximum number of probes that Citrix ADC can send out in 10 milliseconds, to dynamically learn a service. Citrix ADC probes for the existence of the origin in case of wildcard virtual server or services. Default value: 7 Minimum value = 1 Maximum value = 65535
synholdfastgiveup	<Double>	Read-write	Maximum threshold. After crossing this threshold number of outstanding probes for origin, the Citrix ADC reduces the number of connection retries for probe connections.

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
			Default value: 1024 Minimum value = 256 Maximum value = 65535
maxsynholdperprobe	<Double>	Read-write	Limit the number of client connections (SYN) waiting for status of single probe. Any new SYN packets will be dropped. Default value: 128 Minimum value = 1 Maximum value = 255
maxsynhold	<Double>	Read-write	Limit the number of client connections (SYN) waiting for status of probe system wide. Any new SYN packets will be dropped. Default value: 16384 Minimum value = 256 Maximum value = 65535
msslearninterval	<Double>	Read-write	Duration, in seconds, to sample the Maximum Segment

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
			Size (MSS) of the services. The Citrix ADC determines the best MSS to set for the virtual server based on this sampling. The argument to enable maximum segment size (MSS) for virtual servers must be enabled. Default value: 180 Minimum value = 1 Maximum value = 1048576
msslearndelay	<Double>	Read-write	Frequency, in seconds, at which the virtual servers learn the Maximum segment size (MSS) from the services. The argument to enable maximum segment size (MSS) for virtual servers must be enabled. Default value: 3600 Minimum value = 1 Maximum value = 1048576
maxtimewaitconn	<Double>	Read-write	Maximum number of connections to hold in the TCP TIME_WAIT state on

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
			a packet engine. New connections entering TIME_WAIT state are proactively cleaned up. Default value: 7000 Minimum value = 1
kaprobeupdatelastactivity	<String>	Read-write	Update last activity for KA probes. Default value: ENABLED Possible values = ENABLED, DISABLED
maxsynackretx	<Double>	Read-write	When 'syncookie' is disabled in the TCP profile that is bound to the virtual server or service, and the number of TCP SYN+ACK retransmission by Citrix ADC for that virtual server or service crosses this threshold, the Citrix ADC responds by using the TCP SYN-Cookie mechanism. Default value: 100 Minimum value = 100 Maximum value = 1048576

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
synattackdetection	<String>	Read-write	Detect TCP SYN packet flood and send an SNMP trap. Default value: ENABLED Possible values = ENABLED, DISABLED
connflushifnomem	<String>	Read-write	Flush an existing connection if no memory can be obtained for new connection. HALF_CLOSED_AND_IDLE: Flush a connection that is closed by us but not by peer, or failing that, a connection that is past configured idle time. New connection fails if no such connection can be found. FIFO: If no half-closed or idle connection can be found, flush the oldest non-management connection, even if it is active. New connection fails if

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
			the oldest few connections are management connections. Note: If you enable this setting, you should also consider lowering the zombie timeout and half-close timeout, while setting the Citrix ADC timeout. See Also: connFlushThres argument below. Default value: NSA_CONNFLUSH_NONE Possible values = NONE , HALFCLOSED_AND_IDLE, FIFO
connflushthres	<Double>	Read-write	Flush an existing connection (as configured through -connFlushIfNoMem FIFO) if the system has more than specified number of connections, and a new connection is to be established. Note: This value may be

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
			rounded down to be a whole multiple of the number of packet engines running. Minimum value = 1
mptcpconcloseonpassivesf	<String>	Read-write	Accept DATA_FIN/ FAST_CLOSE on passive subflow. Default value: ENABLED Possible values = ENABLED, DISABLED
mptcpchecksum	<String>	Read-write	Use MPTCP DSS checksum. Default value: ENABLED Possible values = ENABLED, DISABLED
mptcpsftimeout	<Double>	Read-write	The timeout value in seconds for idle mptcp subflows. If this timeout is not set, idle subflows are cleared after cltTimeout of vserver. Default value: 0 Minimum value = 0 Maximum value = 31536000

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
mptcpsfreplacetimeout	<Double>	Read-write	The minimum idle time value in seconds for idle mptcp subflows after which the sublow is replaced by new incoming subflow if maximum subflow limit is reached. The priority for replacement is given to those subflow without any transaction. Default value: 10 Minimum value = 0 Maximum value = 31536000
mptcpmaxsf	<Double>	Read-write	Maximum number of subflow connections supported in established state per mptcp connection. Default value: 4 Minimum value = 2 Maximum value = 6
mptcpmaxpendingsf	<Double>	Read-write	Maximum number of subflow connections supported in pending join state per mptcp connection. Default value: 4 Minimum value = 0

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
			Maximum value = 4
mptcppendingjointhreshold	<Double>	Read-write	Maximum system level pending join connections allowed. Default value: 0 Minimum value = 0 Maximum value = 4294967294
mptcprtostoswitchsf	<Double>	Read-write	Number of RTO's at subflow level, after which MPCTP should start using other subflow. Default value: 2 Minimum value = 1 Maximum value = 6
mptcpusebackupondss	<String>	Read-write	When enabled, if NS receives a DSS on a backup subflow, NS will start using that subflow to send data. And if disabled, NS will continue to transmit on current chosen subflow. In case there is some error on a subflow (like RTO's/RST etc.) then NS can choose a backup subflow irrespective of this tunable.

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
			Default value: ENABLED Possible values = ENABLED, DISABLED
tcpmaxretries	<Double>	Read-write	Number of RTO's after which a connection should be freed. Default value: 7 Minimum value = 1 Maximum value = 7
mptcpimmediatesfcloseonfin	<String>	Read-write	Allow subflows to close immediately on FIN before the DATA_FIN exchange is completed at mptcp level. Default value: DISABLED Possible values = ENABLED, DISABLED
mptcpclosemptcpsessiononlastsfclose	<String>	Read-write	Allow to send DATA FIN or FAST CLOSE on mptcp connection while sending FIN or RST on the last subflow. Default value: DISABLED

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
			Possible values = ENABLED, DISABLED
tcpfastopencookietimeout	<Double>	Read-write	Timeout in seconds after which a new TFO Key is computed for generating TFO Cookie. If zero, the same key is used always. If timeout is less than 120seconds, NS defaults to 120seconds timeout. Default value: 0 Minimum value = 0 Maximum value = 31536000
autosyncookietimeout	<Double>	Read-write	Timeout for the server to function in syncookie mode after the synattack. This is valid if TCP syncookie is disabled on the profile and server acts in non syncookie mode by default. Default value: 30 Minimum value = 7 Maximum value = 65535

Operations

(click to see Properties)

UPDATE| UNSET| GET (ALL)

Some options that you can use for each operations:

- **Getting warnings in response:**NITRO allows you to get warnings in an operation by specifying the "warning" query parameter as "yes". For example, to get warnings while connecting to the Citrix ADC appliance, the URL is as follows:

  http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/login?warning=yes

  If any, the warnings are displayed in the response payload with the HTTP code "209 X-NITRO-WARNING".

- **Authenticated access for individual NITRO operations:**NITRO allows you to logon to the Citrix ADC appliance to perform individual operations. You can use this option instead of creating a NITRO session (using the login object) and then using that session to perform all operations,

  To do this, you must specify the username and password in the request header of the NITRO request as follows:

  X-NITRO-USER:<username>

  X-NITRO-PASS:<password>

  Note: In such cases, make sure that the request header DOES not include the following:

  Cookie:NITRO_AUTH_TOKEN=<tokenvalue>

`*Note:`*

Mandatory parameters are marked in redand placeholder content is marked in <green>.

update

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/nstcpparam

HTTP Method:PUT

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Content-Type:application/json

Request Payload:

```
{"nstcpparam":{

"ws":<String_value>,

"wsval":<Double_value>,

"sack":<String_value>,

"learnvsvrmss":<String_value>,

"maxburst":<Double_value>,

"initialcwnd":<Double_value>,

"recvbuffsize":<Double_value>,

"delayedack":<Double_value>,

"downstaterst":<String_value>,

"nagle":<String_value>,

"limitedpersist":<String_value>,

"oooqsize":<Double_value>,

"ackonpush":<String_value>,

"maxpktpermss":<Double_value>,

"pktperretx":<Integer_value>,

"minrto":<Integer_value>,

"slowstartincr":<Integer_value>,

"maxdynserverprobes":<Double_value>,

"synholdfastgiveup":<Double_value>,
```

```
"maxsynholdperprobe":<Double_value>,

"maxsynhold":<Double_value>,

"msslearninterval":<Double_value>,

"msslearndelay":<Double_value>,

"maxtimewaitconn":<Double_value>,

"kaprobeupdatelastactivity":<String_value>,

"maxsynackretx":<Double_value>,

"synattackdetection":<String_value>,

"connflushifnomem":<String_value>,

"connflushthres":<Double_value>,

"mptcpconcloseonpassivesf":<String_value>,

"mptcpchecksum":<String_value>,

"mptcpsftimeout":<Double_value>,

"mptcpsfreplacetimeout":<Double_value>,

"mptcpmaxsf":<Double_value>,

"mptcpmaxpendingsf":<Double_value>,

"mptcppendingjointhreshold":<Double_value>,

"mptcprtostoswitchsf":<Double_value>,

"mptcpusebackupondss":<String_value>,

"tcpmaxretries":<Double_value>,

"mptcpimmediatesfcloseonfin":<String_value>,

"mptcpclosemptcpsessiononlastsfclose":<String_value>,
```

```
"tcpfastopencookietimeout":<Double_value>,

"autosyncookietimeout":<Double_value>

}}
```

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error

unset

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/nstcpparam?**action=unset**

HTTP Method:POST

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Content-Type:application/json

Request Payload:

```
{"nstcpparam":{

"ws":true,

"wsval":true,

"sack":true,

"learnvsvrmss":true,

"maxburst":true,

"initialcwnd":true,

"recvbuffsize":true,
```

```
"delayedack":true,

"downstaterst":true,

"nagle":true,

"limitedpersist":true,

"oooqsize":true,

"ackonpush":true,

"maxpktpermss":true,

"pktperretx":true,

"minrto":true,

"slowstartincr":true,

"maxdynserverprobes":true,

"synholdfastgiveup":true,

"maxsynholdperprobe":true,

"maxsynhold":true,

"msslearninterval":true,

"msslearndelay":true,

"maxtimewaitconn":true,

"kaprobeupdatelastactivity":true,

"maxsynackretx":true,

"synattackdetection":true,

"connflushifnomem":true,

"connflushthres":true,
```

```
"mptcpconcloseonpassivesf":true,

"mptcpchecksum":true,

"mptcpsftimeout":true,

"mptcpsfreplacetimeout":true,

"mptcpmaxsf":true,

"mptcpmaxpendingsf":true,

"mptcppendingjointhreshold":true,

"mptcprtostoswitchsf":true,

"mptcpusebackupondss":true,

"tcpmaxretries":true,

"mptcpimmediatesfcloseonfin":true,

"mptcpclosemptcpsessiononlastsfclose":true,

"tcpfastopencookietimeout":true,

"autosyncookietimeout":true

}}
```

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error

get (all)

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/nstcpparam

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "nstcpparam": [ {

"ws":<String_value>,

"wsval":<Double_value>,

"sack":<String_value>,

"learnvsvrmss":<String_value>,

"maxburst":<Double_value>,

"initialcwnd":<Double_value>,

"recvbuffsize":<Double_value>,

"delayedack":<Double_value>,

"downstaterst":<String_value>,

"nagle":<String_value>,

"limitedpersist":<String_value>,

"oooqsize":<Double_value>,

"ackonpush":<String_value>,
```

```
"maxpktpermss":<Double_value>,

"pktperretx":<Integer_value>,

"minrto":<Integer_value>,

"slowstartincr":<Integer_value>,

"maxdynserverprobes":<Double_value>,

"synholdfastgiveup":<Double_value>,

"maxsynholdperprobe":<Double_value>,

"maxsynhold":<Double_value>,

"msslearninterval":<Double_value>,

"msslearndelay":<Double_value>,

"maxtimewaitconn":<Double_value>,

"kaprobeupdatelastactivity":<String_value>,

"maxsynackretx":<Double_value>,

"synattackdetection":<String_value>,

"connflushifnomem":<String_value>,

"connflushthres":<Double_value>,

"mptcpconcloseonpassivesf":<String_value>,

"mptcpchecksum":<String_value>,

"mptcpsftimeout":<Double_value>,

"mptcpsfreplacetimeout":<Double_value>,

"mptcpmaxsf":<Double_value>,

"mptcpmaxpendingsf":<Double_value>,
```

```
"mptcppendingjointhreshold":<Double_value>,

"mptcprtostoswitchsf":<Double_value>,

"mptcpusebackupondss":<String_value>,

"tcpmaxretries":<Double_value>,

"mptcpimmediatesfcloseonfin":<String_value>,

"mptcpclosemptcpsessiononlastsfclose":<String_value>,

"tcpfastopencookietimeout":<Double_value>,

"autosyncookietimeout":<Double_value>

}]}
```

nstcpprofile

Configuration for TCP profile resource.

Properties

(click to see Operations)

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
name	<String>	Read-write	Name for a TCP profile. Must begin with a letter, number, or the underscore \(_\) character. Other characters allowed, after the first character, are the hyphen \(-\), period \(.\), hash \(#\), space \(\), at \(@\), colon \(:\), and equal \(=\) characters. The

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
			name of a TCP profile cannot be changed after it is created. CLI Users: If the name includes one or more spaces, enclose the name in double or single quotation marks \ (for example, "my tcp profile" or 'my tcp profile'\). Minimum length = 1 Maximum length = 127
ws	<String>	Read-write	Enable or disable window scaling. Default value: DISABLED Possible values = ENABLED, DISABLED
sack	<String>	Read-write	Enable or disable Selective ACKnowledgement (SACK). Default value: DISABLED Possible values = ENABLED, DISABLED

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
wsval	<Double>	Read-write	Factor used to calculate the new window size. This argument is needed only when window scaling is enabled. Default value: 4 Minimum value = 0 Maximum value = 14
nagle	<String>	Read-write	Enable or disable the Nagle algorithm on TCP connections. Default value: DISABLED Possible values = ENABLED, DISABLED
ackonpush	<String>	Read-write	Send immediate positive acknowledgement (ACK) on receipt of TCP packets with PUSH flag. Default value: ENABLED Possible values = ENABLED, DISABLED

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
mss	<Double>	Read-write	Maximum number of octets to allow in a TCP data segment. Minimum value = 0 Maximum value = 9176
maxburst	<Double>	Read-write	Maximum number of TCP segments allowed in a burst. Default value: 6 Minimum value = 1 Maximum value = 255
initialcwnd	<Double>	Read-write	Initial maximum upper limit on the number of TCP packets that can be outstanding on the TCP link to the server. Default value: 4 Minimum value = 1 Maximum value = 44
delayedack	<Double>	Read-write	Timeout for TCP delayed ACK, in milliseconds. Default value: 100 Minimum value = 10 Maximum value = 300

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
oooqsize	<Double>	Read-write	Maximum size of out-of-order packets queue. A value of 0 means no limit. Default value: 64 Minimum value = 0 Maximum value = 65535
maxpktpermss	<Double>	Read-write	Maximum number of TCP packets allowed per maximum segment size (MSS). Minimum value = 0 Maximum value = 1460
pktperretx	<Double>	Read-write	Maximum limit on the number of packets that should be retransmitted on receiving a partial ACK. Default value: 1 Minimum value = 1 Maximum value = 512
minrto	<Double>	Read-write	Minimum retransmission timeout, in milliseconds, specified in 10-millisecond increments (value must yield a whole

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
			number if divided by 10). Default value: 1000 Minimum value = 10 Maximum value = 64000
slowstartincr	<Double>	Read-write	Multiplier that determines the rate at which slow start increases the size of the TCP transmission window after each acknowledgement of successful transmission. Default value: 2 Minimum value = 1 Maximum value = 100
buffersize	<Double>	Read-write	TCP buffering size, in bytes. Default value: 8190 Minimum value = 8190 Maximum value = 20971520
syncookie	<String>	Read-write	Enable or disable the SYNCOOKIE mechanism for TCP handshake with clients. Disabling SYNCOOKIE

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
			prevents SYN attack protection on the Citrix ADC. Default value: ENABLED Possible values = ENABLED, DISABLED
kaprobeupdatelastactivity	<String>	Read-write	Update last activity for the connection after receiving keep-alive (KA) probes. Default value: ENABLED Possible values = ENABLED, DISABLED
flavor	<String>	Read-write	Set TCP congestion control algorithm. Default value: Default Possible values = Default, Westwood, BIC, CUBIC, Nile
dynamicreceivebuffering	<String>	Read-write	Enable or disable dynamic receive buffering. When enabled, allows the receive buffer to be adjusted dynamically based on memory and network conditions.

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
			Note: The buffer size argument must be set for dynamic adjustments to take place. Default value: DISABLED Possible values = ENABLED, DISABLED
ka	<String>	Read-write	Send periodic TCP keep-alive (KA) probes to check if peer is still up. Default value: DISABLED Possible values = ENABLED, DISABLED
kaconnidletime	<Double>	Read-write	Duration, in seconds, for the connection to be idle, before sending a keep-alive (KA) probe. Minimum value = 1 Maximum value = 4095
kamaxprobes	<Double>	Read-write	Number of keep-alive (KA) probes to be sent when not acknowledged, before assuming the peer to be down.

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
			Minimum value = 1 Maximum value = 254
kaprobeinterval	<Double>	Read-write	Time interval, in seconds, before the next keep-alive (KA) probe, if the peer does not respond. Minimum value = 1 Maximum value = 4095
sendbuffsize	<Double>	Read-write	TCP Send Buffer Size. Default value: 8190 Minimum value = 8190 Maximum value = 20971520
mptcp	<String>	Read-write	Enable or disable Multipath TCP. Default value: DISABLED Possible values = ENABLED, DISABLED
establishclientconn	<String>	Read-write	Establishing Client Client connection on First data/ Final-ACK / Automatic.

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
			Default value: AUTOMATIC Possible values = AUTOMATIC, CONN_ESTABLISHED, ON_FIRST_DATA
tcpsegoffload	<String>	Read-write	Offload TCP segmentation to the NIC. If set to AUTOMATIC, TCP segmentation will be offloaded to the NIC, if the NIC supports it. Default value: AUTOMATIC Possible values = AUTOMATIC, DISABLED
rstwindowattenuate	<String>	Read-write	Enable or disable RST window attenuation to protect against spoofing. When enabled, will reply with corrective ACK when a sequence number is invalid. Default value: DISABLED Possible values = ENABLED, DISABLED

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
rstmaxack	<String>	Read-write	Enable or disable acceptance of RST that is out of window yet echoes highest ACK sequence number. Useful only in proxy mode. Default value: DISABLED Possible values = ENABLED, DISABLED
spoofsyndrop	<String>	Read-write	Enable or disable drop of invalid SYN packets to protect against spoofing. When disabled, established connections will be reset when a SYN packet is received. Default value: ENABLED Possible values = ENABLED, DISABLED
ecn	<String>	Read-write	Enable or disable TCP Explicit Congestion Notification. Default value: DISABLED

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
			Possible values = ENABLED, DISABLED
mptcpdropdataonpreestsf	<String>	Read-write	Enable or disable silently dropping the data on Pre-Established subflow. When enabled, DSS data packets are dropped silently instead of dropping the connection when data is received on pre established subflow. Default value: DISABLED Possible values = ENABLED, DISABLED
mptcpfastopen	<String>	Read-write	Enable or disable Multipath TCP fastopen. When enabled, DSS data packets are accepted before receiving the third ack of SYN handshake. Default value: DISABLED Possible values = ENABLED, DISABLED

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
mptcpsessiontimeout	<Double>	Read-write	MPTCP session timeout in seconds. If this value is not set, idle MPTCP sessions are flushed after vserver's client idle timeout. Default value: 0 Minimum value = 0 Maximum value = 86400
timestamp	<String>	Read-write	Enable or Disable TCP Timestamp option (RFC 1323). Default value: DISABLED Possible values = ENABLED, DISABLED
dsack	<String>	Read-write	Enable or disable DSACK. Default value: ENABLED Possible values = ENABLED, DISABLED
ackaggregation	<String>	Read-write	Enable or disable ACK Aggregation. Default value: DISABLED

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
			Possible values = ENABLED, DISABLED
frto	<String>	Read-write	Enable or disable FRTO (Forward RTO-Recovery). Default value: DISABLED Possible values = ENABLED, DISABLED
maxcwnd	<Double>	Read-write	TCP Maximum Congestion Window. Default value: 524288 Minimum value = 8190 Maximum value = 20971520
fack	<String>	Read-write	Enable or disable FACK (Forward ACK). Default value: DISABLED Possible values = ENABLED, DISABLED
tcpmode	<String>	Read-write	TCP Optimization modes TRANSPARENT / ENDPOINT.

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
			Default value: TRANSPARENT Possible values = TRANSPARENT, ENDPOINT
tcpfastopen	<String>	Read-write	Enable or disable TCP Fastopen. When enabled, NS can receive or send Data in SYN or SYN-ACK packets. Default value: DISABLED Possible values = ENABLED, DISABLED
hystart	<String>	Read-write	Enable or disable CUBIC Hystart. Default value: DISABLED Possible values = ENABLED, DISABLED
dupackthresh	<Double>	Read-write	TCP dupack threshold. Default value: 3 Minimum value = 1 Maximum value = 15
burstratecontrol	<String>	Read-write	TCP Burst Rate Control DISABLED/

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
			FIXED/DYNAMIC. FIXED requires a TCP rate to be set. Default value: DISABLED Possible values = DISABLED, FIXED, DYNAMIC
tcprate	<Double>	Read-write	TCP connection payload send rate in Kb/s. Default value: 0 Minimum value = 0 Maximum value = 10000000
rateqmax	<Double>	Read-write	Maximum connection queue size in bytes, when BurstRateControl is used. Default value: 0 Minimum value = 0 Maximum value = 1000000000
drophalfclosedconnontimeout	<String>	Read-write	Silently drop tcp half closed connections on idle timeout. Default value: DISABLED

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
			Possible values = ENABLED, DISABLED
dropestconnontimeout	<String>	Read-write	Silently drop tcp established connections on idle timeout. Default value: DISABLED Possible values = ENABLED, DISABLED
applyadaptivetcp	<String>	Read-write	Apply Adaptive TCP optimizations. Default value: DISABLED Possible values = ENABLED, DISABLED
tcpfastopencookiesize	<Double>	Read-write	TCP FastOpen Cookie size. This accepts only even numbers. Odd number is trimmed down to nearest even number. Default value: 8 Minimum value = 4 Maximum value = 16
taillossprobe	<String>	Read-write	TCP tail loss probe optimizations.

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
			Default value: DISABLED Possible values = ENABLED, DISABLED
refcnt	<Double>	Read-only	Number of entities using this profile.
__count	<Double>	Read-only	count parameter

Operations

(click to see Properties)

ADD| DELETE| UPDATE| UNSET| GET (ALL)| GET| COUNT

Some options that you can use for each operations:

- **Getting warnings in response:**NITRO allows you to get warnings in an operation by specifying the "warning" query parameter as "yes". For example, to get warnings while connecting to the Citrix ADC appliance, the URL is as follows:

  http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/login?warning=yes

  If any, the warnings are displayed in the response payload with the HTTP code "209 X-NITRO-WARNING".

- **Authenticated access for individual NITRO operations:**NITRO allows you to logon to the Citrix ADC appliance to perform individual operations. You can use this option instead of creating a NITRO session (using the login object) and then using that session to perform all operations,

  To do this, you must specify the username and password in the request header of the NITRO request as follows:

  X-NITRO-USER:<username>

  X-NITRO-PASS:<password>

  Note: In such cases, make sure that the request header DOES not include the following:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue>

`*Note:` *

Mandatory parameters are marked in redand placeholder content is marked in <green>.

add

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/nstcpprofile

HTTP Method:POST

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Content-Type:application/json

Request Payload:

```
{"nstcpprofile":{

<b>"name":<String_value>,

</b>"ws":<String_value>,

"sack":<String_value>,

"wsval":<Double_value>,

"nagle":<String_value>,

"ackonpush":<String_value>,

"mss":<Double_value>,

"maxburst":<Double_value>,

"initialcwnd":<Double_value>,

"delayedack":<Double_value>,

"oooqsize":<Double_value>,

"maxpktpermss":<Double_value>,
```

```
"pktperretx":<Double_value>,

"minrto":<Double_value>,

"slowstartincr":<Double_value>,

"buffersize":<Double_value>,

"syncookie":<String_value>,

"kaprobeupdatelastactivity":<String_value>,

"flavor":<String_value>,

"dynamicreceivebuffering":<String_value>,

"ka":<String_value>,

"kaconnidletime":<Double_value>,

"kamaxprobes":<Double_value>,

"kaprobeinterval":<Double_value>,

"sendbuffsize":<Double_value>,

"mptcp":<String_value>,

"establishclientconn":<String_value>,

"tcpsegoffload":<String_value>,

"rstwindowattenuate":<String_value>,

"rstmaxack":<String_value>,

"spoofsyndrop":<String_value>,

"ecn":<String_value>,

"mptcpdropdataonpreestsf":<String_value>,

"mptcpfastopen":<String_value>,
```

```
"mptcpsessiontimeout":<Double_value>,

"timestamp":<String_value>,

"dsack":<String_value>,

"ackaggregation":<String_value>,

"frto":<String_value>,

"maxcwnd":<Double_value>,

"fack":<String_value>,

"tcpmode":<String_value>,

"tcpfastopen":<String_value>,

"hystart":<String_value>,

"dupackthresh":<Double_value>,

"burstratecontrol":<String_value>,

"tcprate":<Double_value>,

"rateqmax":<Double_value>,

"drophalfclosedconnontimeout":<String_value>,

"dropestconnontimeout":<String_value>,

"applyadaptivetcp":<String_value>,

"tcpfastopencookiesize":<Double_value>,

"taillossprobe":<String_value>

}}
```

Response:

HTTP Status Code on Success: 201 Created HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error

delete

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/nstcpprofile/name_value<String>

HTTP Method:DELETE

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue>

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error

update

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/nstcpprofile

HTTP Method:PUT

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Content-Type:application/json

Request Payload:

```
{"nstcpprofile":{

<b>"name":<String_value>,

</b>"ws":<String_value>,

"sack":<String_value>,

"wsval":<Double_value>,

"nagle":<String_value>,
```

```
"ackonpush":<String_value>,

"mss":<Double_value>,

"maxburst":<Double_value>,

"initialcwnd":<Double_value>,

"delayedack":<Double_value>,

"oooqsize":<Double_value>,

"maxpktpermss":<Double_value>,

"pktperretx":<Double_value>,

"minrto":<Double_value>,

"slowstartincr":<Double_value>,

"buffersize":<Double_value>,

"syncookie":<String_value>,

"kaprobeupdatelastactivity":<String_value>,

"flavor":<String_value>,

"dynamicreceivebuffering":<String_value>,

"ka":<String_value>,

"kaconnidletime":<Double_value>,

"kamaxprobes":<Double_value>,

"kaprobeinterval":<Double_value>,

"sendbuffsize":<Double_value>,

"mptcp":<String_value>,

"establishclientconn":<String_value>,
```

```
"tcpsegoffload":<String_value>,

"rstwindowattenuate":<String_value>,

"rstmaxack":<String_value>,

"spoofsyndrop":<String_value>,

"ecn":<String_value>,

"mptcpdropdataonpreestsf":<String_value>,

"mptcpfastopen":<String_value>,

"mptcpsessiontimeout":<Double_value>,

"timestamp":<String_value>,

"dsack":<String_value>,

"ackaggregation":<String_value>,

"frto":<String_value>,

"maxcwnd":<Double_value>,

"fack":<String_value>,

"tcpmode":<String_value>,

"tcpfastopen":<String_value>,

"hystart":<String_value>,

"dupackthresh":<Double_value>,

"burstratecontrol":<String_value>,

"tcprate":<Double_value>,

"rateqmax":<Double_value>,

"drophalfclosedconnontimeout":<String_value>,
```

```
"dropestconnontimeout":<String_value>,

"applyadaptivetcp":<String_value>,

"tcpfastopencookiesize":<Double_value>,

"taillossprobe":<String_value>

}}
```

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error

unset

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/nstcpprofile?**action=unset**

HTTP Method:POST

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Content-Type:application/json

Request Payload:

```
{"nstcpprofile":{

<b>"name":<String_value>,

</b>"ws":true,

"sack":true,

"wsval":true,

"nagle":true,
```

```
"ackonpush":true,

"mss":true,

"maxburst":true,

"initialcwnd":true,

"delayedack":true,

"oooqsize":true,

"maxpktpermss":true,

"pktperretx":true,

"minrto":true,

"slowstartincr":true,

"buffersize":true,

"syncookie":true,

"kaprobeupdatelastactivity":true,

"flavor":true,

"dynamicreceivebuffering":true,

"ka":true,

"kamaxprobes":true,

"kaconnidletime":true,

"kaprobeinterval":true,

"sendbuffsize":true,

"mptcp":true,

"establishclientconn":true,
```

```
"tcpsegoffload":true,

"rstwindowattenuate":true,

"rstmaxack":true,

"spoofsyndrop":true,

"ecn":true,

"mptcpdropdataonpreestsf":true,

"mptcpfastopen":true,

"mptcpsessiontimeout":true,

"timestamp":true,

"dsack":true,

"ackaggregation":true,

"frto":true,

"maxcwnd":true,

"fack":true,

"tcpmode":true,

"tcpfastopen":true,

"hystart":true,

"dupackthresh":true,

"burstratecontrol":true,

"tcprate":true,

"rateqmax":true,

"drophalfclosedconnontimeout":true,
```

```
"dropestconnontimeout":true,

"applyadaptivetcp":true,

"tcpfastopencookiesize":true,

"taillossprobe":true

}}
```

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error

get (all)

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/nstcpprofile

Query-parameters:

attrs

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/nstcpprofile?**attrs=property-name1,property-name2**

Use this query parameter to specify the resource details that you want to retrieve.

filter

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/nstcpprofile?**filter=property-name1:property-val1,property-name2:property-val2**

Use this query-parameter to get the filtered set of nstcpprofile resources configured on Citrix ADC. Filtering can be done on any of the properties of the resource.

view

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/nstcpprofile?**view=summary**

Note: By default, the retrieved results are displayed in detail view (?view=detail).

pagination

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/nstcpprofile?**pagesize=#no;pageno=#no**

Use this query-parameter to get the nstcpprofile resources in chunks.

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "nstcpprofile": [ {

"name":<String_value>,

"ws":<String_value>,

"sack":<String_value>,

"wsval":<Double_value>,

"nagle":<String_value>,

"ackonpush":<String_value>,

"mss":<Double_value>,

"maxburst":<Double_value>,

"initialcwnd":<Double_value>,
```

```
"delayedack":<Double_value>,

"oooqsize":<Double_value>,

"maxpktpermss":<Double_value>,

"pktperretx":<Double_value>,

"minrto":<Double_value>,

"slowstartincr":<Double_value>,

"buffersize":<Double_value>,

"flavor":<String_value>,

"refcnt":<Double_value>,

"syncookie":<String_value>,

"kaprobeupdatelastactivity":<String_value>,

"dynamicreceivebuffering":<String_value>,

"ka":<String_value>,

"kaconnidletime":<Double_value>,

"kamaxprobes":<Double_value>,

"kaprobeinterval":<Double_value>,

"sendbuffsize":<Double_value>,

"mptcp":<String_value>,

"establishclientconn":<String_value>,

"tcpsegoffload":<String_value>,

"rstwindowattenuate":<String_value>,

"rstmaxack":<String_value>,
```

```
"timestamp":<String_value>,

"spoofsyndrop":<String_value>,

"ecn":<String_value>,

"mptcpdropdataonpreestsf":<String_value>,

"mptcpfastopen":<String_value>,

"mptcpsessiontimeout":<Double_value>,

"dsack":<String_value>,

"ackaggregation":<String_value>,

"frto":<String_value>,

"maxcwnd":<Double_value>,

"fack":<String_value>,

"tcpmode":<String_value>,

"tcpfastopen":<String_value>,

"hystart":<String_value>,

"dupackthresh":<Double_value>,

"burstratecontrol":<String_value>,

"tcprate":<Double_value>,

"rateqmax":<Double_value>,

"drophalfclosedconnontimeout":<String_value>,

"dropestconnontimeout":<String_value>,

"applyadaptivetcp":<String_value>,

"tcpfastopencookiesize":<Double_value>,
```

```
"taillossprobe":<String_value>

}]}
```

get

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/nstcpprofile/name_value<String>

Query-parameters:

attrs

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/nstcpprofile/name_value<String>?**attrs=property-name1,property-name2**

Use this query parameter to specify the resource details that you want to retrieve.

view

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/nstcpprofile/name_value<String>?**view=summary**

Note: By default, the retrieved results are displayed in detail view (?view=detail).

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "nstcpprofile": [ {
```

```
"name":<String_value>,

"ws":<String_value>,

"sack":<String_value>,

"wsval":<Double_value>,

"nagle":<String_value>,

"ackonpush":<String_value>,

"mss":<Double_value>,

"maxburst":<Double_value>,

"initialcwnd":<Double_value>,

"delayedack":<Double_value>,

"oooqsize":<Double_value>,

"maxpktpermss":<Double_value>,

"pktperretx":<Double_value>,

"minrto":<Double_value>,

"slowstartincr":<Double_value>,

"buffersize":<Double_value>,

"flavor":<String_value>,

"refcnt":<Double_value>,

"syncookie":<String_value>,

"kaprobeupdatelastactivity":<String_value>,

"dynamicreceivebuffering":<String_value>,

"ka":<String_value>,
```

```
"kaconnidletime":<Double_value>,

"kamaxprobes":<Double_value>,

"kaprobeinterval":<Double_value>,

"sendbuffsize":<Double_value>,

"mptcp":<String_value>,

"establishclientconn":<String_value>,

"tcpsegoffload":<String_value>,

"rstwindowattenuate":<String_value>,

"rstmaxack":<String_value>,

"timestamp":<String_value>,

"spoofsyndrop":<String_value>,

"ecn":<String_value>,

"mptcpdropdataonpreestsf":<String_value>,

"mptcpfastopen":<String_value>,

"mptcpsessiontimeout":<Double_value>,

"dsack":<String_value>,

"ackaggregation":<String_value>,

"frto":<String_value>,

"maxcwnd":<Double_value>,

"fack":<String_value>,

"tcpmode":<String_value>,

"tcpfastopen":<String_value>,
```

```
"hystart":<String_value>,

"dupackthresh":<Double_value>,

"burstratecontrol":<String_value>,

"tcprate":<Double_value>,

"rateqmax":<Double_value>,

"drophalfclosedconnontimeout":<String_value>,

"dropestconnontimeout":<String_value>,

"applyadaptivetcp":<String_value>,

"tcpfastopencookiesize":<Double_value>,

"taillossprobe":<String_value>

}]}
```

count

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/nstcpprofile?**count=yes**

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "nstcpprofile": [ { "__count": "#no"} ] }
```

nstimeout

Configuration for timeout resource.

Properties

(click to see Operations)

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
zombie	<Double>	Read-write	Interval, in seconds, at which the Citrix ADC zombie cleanup process must run. This process cleans up inactive TCP connections. Default value: 120 Minimum value = 1 Maximum value = 600
client	<Double>	Read-write	Client idle timeout (in seconds). If zero, the service-type default value is taken when service is created. Default value: 0 Minimum value = 0 Maximum value = 18000

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
server	<Double>	Read-write	Server idle timeout (in seconds). If zero, the service-type default value is taken when service is created. Default value: 0 Minimum value = 0 Maximum value = 18000
httpclient	<Double>	Read-write	Global idle timeout, in seconds, for client connections of HTTP service type. This value is over ridden by the client timeout that is configured on individual entities. Default value: 0 Minimum value = 0 Maximum value = 18000
httpserver	<Double>	Read-write	Global idle timeout, in seconds, for server connections of HTTP service type. This value is over ridden by the server timeout that is configured on individual entities. Default value: 0 Minimum value = 0

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
			Maximum value = 18000
tcpclient	<Double>	Read-write	Global idle timeout, in seconds, for non-HTTP client connections of TCP service type. This value is over ridden by the client timeout that is configured on individual entities. Default value: 0 Minimum value = 0 Maximum value = 18000
tcpserver	<Double>	Read-write	Global idle timeout, in seconds, for non-HTTP server connections of TCP service type. This value is over ridden by the server timeout that is configured on entities. Default value: 0 Minimum value = 0 Maximum value = 18000
anyclient	<Double>	Read-write	Global idle timeout, in seconds, for non-TCP client connections. This value is over ridden by the client timeout

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
			that is configured on individual entities. Default value: 0 Minimum value = 0 Maximum value = 31536000
anyserver	<Double>	Read-write	Global idle timeout, in seconds, for non TCP server connections. This value is over ridden by the server timeout that is configured on individual entities. Default value: 0 Minimum value = 0 Maximum value = 31536000
anytcpclient	<Double>	Read-write	Global idle timeout, in seconds, for TCP client connections. This value takes precedence over entity level timeout settings (vserver/service). This is applicable only to transport protocol TCP. Default value: 0 Minimum value = 0 Maximum value = 31536000

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
anytcpserver	<Double>	Read-write	Global idle timeout, in seconds, for TCP server connections. This value takes precedence over entity level timeout settings (vserver/ service). This is applicable only to transport protocol TCP. Default value: 0 Minimum value = 0 Maximum value = 31536000
halfclose	<Double>	Read-write	Idle timeout, in seconds, for connections that are in TCP half-closed state. Default value: 10 Minimum value = 1 Maximum value = 600
nontcpzombie	<Double>	Read-write	Interval at which the zombie clean-up process for non-TCP connections should run. Inactive IP NAT connections will be cleaned up. Default value: 60 Minimum value = 1

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
			Maximum value = 600
reducedfintimeout	<Double>	Read-write	Alternative idle timeout, in seconds, for closed TCP NATPCB connections. Default value: 30 Minimum value = 1 Maximum value = 300
reducedrsttimeout	<Double>	Read-write	Timer interval, in seconds, for abruptly terminated TCP NATPCB connections. Default value: 0 Minimum value = 0 Maximum value = 300
newconnidletimeout	<Double>	Read-write	Timer interval, in seconds, for new TCP NATPCB connections on which no data was received. Default value: 4 Minimum value = 1 Maximum value = 120

Operations

(click to see Properties)

UPDATE| UNSET| GET (ALL)

Some options that you can use for each operations:

- **Getting warnings in response:**NITRO allows you to get warnings in an operation by specifying the "warning" query parameter as "yes". For example, to get warnings while connecting to the Citrix ADC appliance, the URL is as follows:

  http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/login?warning=yes

  If any, the warnings are displayed in the response payload with the HTTP code "209 X-NITRO-WARNING".

- **Authenticated access for individual NITRO operations:**NITRO allows you to logon to the Citrix ADC appliance to perform individual operations. You can use this option instead of creating a NITRO session (using the login object) and then using that session to perform all operations,

  To do this, you must specify the username and password in the request header of the NITRO request as follows:

  X-NITRO-USER:<username>

  X-NITRO-PASS:<password>

  Note: In such cases, make sure that the request header DOES not include the following:

  Cookie:NITRO_AUTH_TOKEN=<tokenvalue>

`*Note:`*

Mandatory parameters are marked in redand placeholder content is marked in <green>.

update

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/nstimeout

HTTP Method:PUT

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Content-Type:application/json

Request Payload:

```
{"nstimeout":{
"zombie":<Double_value>,
"client":<Double_value>,
"server":<Double_value>,
"httpclient":<Double_value>,
"httpserver":<Double_value>,
"tcpclient":<Double_value>,
"tcpserver":<Double_value>,
"anyclient":<Double_value>,
"anyserver":<Double_value>,
"anytcpclient":<Double_value>,
"anytcpserver":<Double_value>,
"halfclose":<Double_value>,
"nontcpzombie":<Double_value>,
"reducedfintimeout":<Double_value>,
"reducedrsttimeout":<Double_value>,
"newconnidletimeout":<Double_value>
}}
```

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error

unset

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/nstimeout?**action=unset**

HTTP Method:POST

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Content-Type:application/json

Request Payload:

```
{"nstimeout":{

"zombie":true,

"client":true,

"server":true,

"httpclient":true,

"httpserver":true,

"tcpclient":true,

"tcpserver":true,

"anyclient":true,

"anyserver":true,

"anytcpclient":true,

"anytcpserver":true,

"halfclose":true,
```

```
"nontcpzombie":true,

"reducedfintimeout":true,

"reducedrsttimeout":true,

"newconnidletimeout":true

}}
```

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error

get (all)

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/nstimeout

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "nstimeout": [ {

"zombie":<Double_value>,
```

```
"client":<Double_value>,

"server":<Double_value>,

"httpclient":<Double_value>,

"httpserver":<Double_value>,

"tcpclient":<Double_value>,

"tcpserver":<Double_value>,

"anyclient":<Double_value>,

"anyserver":<Double_value>,

"halfclose":<Double_value>,

"anytcpclient":<Double_value>,

"anytcpserver":<Double_value>,

"nontcpzombie":<Double_value>,

"reducedfintimeout":<Double_value>,

"reducedrsttimeout":<Double_value>,

"newconnidletimeout":<Double_value>

}]}
```

nstimer

Configuration for Timer resource.

Properties

(click to see Operations)

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
name	<String>	Read-write	Timer name. Minimum length = 1
interval	<Integer>	Read-write	The frequency at which the policies bound to this timer are invoked. The minimum value is 20 msec. The maximum value is 20940 in seconds and 349 in minutes. Default value: 5 Minimum value = 1 Maximum value = 20940000
unit	<String>	Read-write	Timer interval unit. Default value: SEC Possible values = SEC, MIN
comment	<String>	Read-write	Comments associated with this timer.
newname	<String>	Read-write	The new name of the timer. Minimum length = 1
__count	<Double>	Read-only	count parameter

Operations

(click to see Properties)

ADD| DELETE| UPDATE| UNSET| RENAME| GET (ALL)| GET| COUNT

Some options that you can use for each operations:

- **Getting warnings in response:**NITRO allows you to get warnings in an operation by specifying the "warning" query parameter as "yes". For example, to get warnings while connecting to the Citrix ADC appliance, the URL is as follows:

  http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/login?warning=yes

  If any, the warnings are displayed in the response payload with the HTTP code "209 X-NITRO-WARNING".

- **Authenticated access for individual NITRO operations:**NITRO allows you to logon to the Citrix ADC appliance to perform individual operations. You can use this option instead of creating a NITRO session (using the login object) and then using that session to perform all operations,

  To do this, you must specify the username and password in the request header of the NITRO request as follows:

  X-NITRO-USER:<username>

  X-NITRO-PASS:<password>

  Note: In such cases, make sure that the request header DOES not include the following:

  Cookie:NITRO_AUTH_TOKEN=<tokenvalue>

`*Note:`*

Mandatory parameters are marked in redand placeholder content is marked in <green>.

add

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/nstimer

HTTP Method:POST

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Content-Type:application/json

Request Payload:

```
{"nstimer":{

<b>"name":<String_value>,

</b><b>"interval":<Integer_value>,

</b>"unit":<String_value>,

"comment":<String_value>

}}
```

Response:

HTTP Status Code on Success: 201 Created HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error

delete

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/nstimer/name_value<String>

HTTP Method:DELETE

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue>

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error

update

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/nstimer

HTTP Method:PUT

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Content-Type:application/json

Request Payload:

```
{"nstimer":{

<b>"name":<String_value>,

</b>"interval":<Integer_value>,

"unit":<String_value>,

"comment":<String_value>

}}
```

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error

unset

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/nstimer?**action=unset**

HTTP Method:POST

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Content-Type:application/json

Request Payload:

```
{"nstimer":{

<b>"name":<String_value>,

</b>"interval":true,
```

```
"unit":true,

"comment":true

}}
```

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error

rename

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/nstimer?**action=rename**

HTTP Method:POST

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Content-Type:application/json

Request Payload:

```
{"nstimer":{

<b>"name":<String_value>,

</b><b>"newname":<String_value>

</b>}}
```

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error

get (all)

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/nstimer

Query-parameters:

attrs

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/nstimer?**attrs=property-name1,property-name2**

Use this query parameter to specify the resource details that you want to retrieve.

filter

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/nstimer?**filter=property-name1:property-val1,property-name2:property-val2**

Use this query-parameter to get the filtered set of nstimer resources configured on Citrix ADC. Filtering can be done on any of the properties of the resource.

view

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/nstimer?**view=summary**

Note: By default, the retrieved results are displayed in detail view (?view=detail).

pagination

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/nstimer?**pagesize=#no;pageno=#no**

Use this query-parameter to get the nstimer resources in chunks.

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "nstimer": [ {

"name":<String_value>,

"interval":<Integer_value>,

"unit":<String_value>,

"comment":<String_value>

}]}
```

get

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/nstimer/name_value<String>

Query-parameters:

attrs

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/nstimer/name_value<String>?**attrs=property-name1,property-name2**

Use this query parameter to specify the resource details that you want to retrieve.

view

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/nstimer/name_value<String>?**view=summary**

Note: By default, the retrieved results are displayed in detail view (?view=detail).

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "nstimer": [ {

"name":<String_value>,

"interval":<Integer_value>,

"unit":<String_value>,

"comment":<String_value>

}]}
```

count

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/nstimer?**count=yes**

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "nstimer": [ { "__count": "#no"} ] }
```

nstimer_autoscalepolicy_binding

Binding object showing the autoscalepolicy that can be bound to nstimer.

Properties

(click to see Operations)

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
priority	<Double>	Read-write	Specifies the priority of the timer policy.
gotopriorityexpression	<String>	Read-write	Expression specifying the priority of the next policy which will get evaluated if the current policy rule evaluates to TRUE.
policyname	<String>	Read-write	The timer policy associated with the timer.
name	<String>	Read-write	Timer name. Minimum length = 1
threshold	<Double>	Read-write	Denotes the threshold. If the rule of the policy in the binding relation evaluates 'threshold size' number of times in 'sample size' to true, then the corresponding action is taken. Its value

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
			needs to be less than or equal to the sample size value. Default value: 3 Minimum value = 1 Maximum value = 32
samplesize	<Double>	Read-write	Denotes the sample size. Sample size value of 'x' means that previous '(x - 1)' policy's rule evaluation results and the current evaluation result are present with the binding. For example, sample size of 10 means that there is a state of previous 9 policy evaluation results and also the current policy evaluation result. Default value: 3 Minimum value = 1 Maximum value = 32
vserver	<String>	Read-write	Name of the vserver which provides the context for the rule in timer policy. When not specified it is

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
			treated as a Global Default context.
__count	<Double>	Read-write	count parameter

Operations

(click to see Properties)

ADD:| DELETE:| GET| GET (ALL)| COUNT

Some options that you can use for each operations:

- **Getting warnings in response:**NITRO allows you to get warnings in an operation by specifying the "warning" query parameter as "yes". For example, to get warnings while connecting to the Citrix ADC appliance, the URL is as follows:

  http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/login?warning=yes

  If any, the warnings are displayed in the response payload with the HTTP code "209 X-NITRO-WARNING".
- **Authenticated access for individual NITRO operations:**NITRO allows you to logon to the Citrix ADC appliance to perform individual operations. You can use this option instead of creating a NITRO session (using the login object) and then using that session to perform all operations,

  To do this, you must specify the username and password in the request header of the NITRO request as follows:

  X-NITRO-USER:<username>

  X-NITRO-PASS:<password>

  Note: In such cases, make sure that the request header DOES not include the following:

  Cookie:NITRO_AUTH_TOKEN=<tokenvalue>

`*Note:`*

Mandatory parameters are marked in redand placeholder content is marked in <green>.

add

URL:http://<Citrix-ADC-IP-address(NSIP)/nitro/v1/config/nstimer_autoscalepolicy_binding

HTTP Method:PUT

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Content-Type:application/json

Request Payload:

```
{

"nstimer_autoscalepolicy_binding":{

<b>"name":<String_value>,

</b><b>"policyname":<String_value>,

</b><b>"priority":<Double_value>,

</b>"gotopriorityexpression":<String_value>,

"vserver":<String_value>,

"samplesize":<Double_value>,

"threshold":<Double_value>

}}
```

Response:

HTTP Status Code on Success: 201 Created HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error

delete

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/nstimer_autoscalepolicy_binding/name_value<String>

Query-parameters:

args

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/nstimer_autoscalepolicy_binding/name_value<String>?**args=policyname:<String_value>**

HTTP Method:DELETE

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue>

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error

get

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/nstimer_autoscalepolicy_binding/name_value<String>

Query-parameters:

filter

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/nstimer_autoscalepolicy_binding/name_value<String>?**filter=property-name1:property-value1,property-name2:property-value2**

Use this query-parameter to get the filtered set of nstimer_autoscalepolicy_binding resources configured on Citrix ADC. Filtering can be done on any of the properties of the resource.

pagination

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/nstimer_autoscalepolicy_binding/name_value<String>?**pagesize=#no;pageno=#no**

Use this query-parameter to get the nstimer_autoscalepolicy_binding resources in chunks.

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "nstimer_autoscalepolicy_binding": [ {

"priority":<Double_value>,

"gotopriorityexpression":<String_value>,

"policyname":<String_value>,

"name":<String_value>,

"threshold":<Double_value>,

"samplesize":<Double_value>,

"vserver":<String_value>

}]}
```

get (all)

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/nstimer_autoscalepolicy_binding

Query-parameters:

bulkbindings

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/nstimer_autoscalepolicy_binding?**bulkbindings=yes**

NITRO allows you to fetch bindings in bulk.

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "nstimer_autoscalepolicy_binding": [ {

"priority":<Double_value>,

"gotopriorityexpression":<String_value>,

"policyname":<String_value>,

"name":<String_value>,

"threshold":<Double_value>,

"samplesize":<Double_value>,

"vserver":<String_value>

}]}
```

count

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/nstimer_autoscalepolicy_binding/name_value<String>?**count=yes**

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{"nstimer_autoscalepolicy_binding": [ { "__count": "#no"} ] }
```

nstimer_binding

Binding object which returns the resources bound to nstimer.

Properties

(click to see Operations)

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
name	<String>	Read-write	Timer name. Minimum length = 1
nstimer_autoscalepolicy_binding	<nstimer_autoscalepolicy_binding[]>	Read-only	autoscalepolicy that can be bound to nstimer.

Operations

(click to see Properties)

GET| GET (ALL)

Some options that you can use for each operations:

- **Getting warnings in response:**NITRO allows you to get warnings in an operation by specifying the "warning" query parameter as "yes". For example, to get warnings while connecting to the Citrix ADC appliance, the URL is as follows:

  http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/login?warning=yes

  If any, the warnings are displayed in the response payload with the HTTP code "209 X-NITRO-WARNING".

- **Authenticated access for individual NITRO operations:**NITRO allows you to logon to the Citrix ADC appliance to perform individual operations. You can use this option instead of creating a NITRO session (using the login object) and then using that session to perform all operations,

  To do this, you must specify the username and password in the request header of the NITRO request as follows:

  X-NITRO-USER:<username>

  X-NITRO-PASS:<password>

  Note: In such cases, make sure that the request header DOES not include the following:

  Cookie:NITRO_AUTH_TOKEN=<tokenvalue>

`*Note:`*

Mandatory parameters are marked in redand placeholder content is marked in <green>.

get

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/nstimer_binding/name_value<String>

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "nstimer_binding": [ {

"name":<String_value>,

"nstimer_autoscalepolicy_binding":<nstimer_autoscalepolicy_binding[]_value
>

}]}
```

get (all)

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/nstimer_binding

Query-parameters:

bulkbindings

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/nstimer_binding?**bulkbindings=yes**

NITRO allows you to fetch bindings in bulk.

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "nstimer_binding": [ {

"name":<String_value>,

"nstimer_autoscalepolicy_binding":<nstimer_autoscalepolicy_binding[]_value
>

}]}
```

nstimezone

Configuration for 0 resource.

Properties

(click to see Operations)

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
value	<String>	Read-write	. Maximum length = 64
__count	<Double>	Read-only	count parameter

Operations

(click to see Properties)

GET (ALL)| GET| COUNT

Some options that you can use for each operations:

- **Getting warnings in response:**NITRO allows you to get warnings in an operation by specifying the "warning" query parameter as "yes". For example, to get warnings while connecting to the Citrix ADC appliance, the URL is as follows:

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/login?warning=yes

If any, the warnings are displayed in the response payload with the HTTP code "209 X-NITRO-WARNING".

- **Authenticated access for individual NITRO operations:**NITRO allows you to logon to the Citrix ADC appliance to perform individual operations. You can use this option instead of creating a NITRO session (using the login object) and then using that session to perform all operations,

  To do this, you must specify the username and password in the request header of the NITRO request as follows:

  X-NITRO-USER:<username>

  X-NITRO-PASS:<password>

  Note: In such cases, make sure that the request header DOES not include the following:

  Cookie:NITRO_AUTH_TOKEN=<tokenvalue>

`*Note:`*

Mandatory parameters are marked in redand placeholder content is marked in <green>.

get (all)

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/nstimezone

Query-parameters:

attrs

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/nstimezone?**attrs=property-name1,property-name2**

Use this query parameter to specify the resource details that you want to retrieve.

filter

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/nstimezone?**filter=property-name1:property-val1,property-name2:property-val2**

Use this query-parameter to get the filtered set of nstimezone resources configured on Citrix ADC. Filtering can be done on any of the properties of the resource.

view

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/nstimezone?**view=summary**

Note: By default, the retrieved results are displayed in detail view (?view=detail).

pagination

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/nstimezone?**pagesize=#no;pageno=#no**

Use this query-parameter to get the nstimezone resources in chunks.

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "nstimezone": [ {

"value":<String_value>,



}]}
```

get

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/nstimezone/value_value<String>

Query-parameters:

attrs

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/nstimezone/value_value<String>?**attrs=property-name1,property-name2**

Use this query parameter to specify the resource details that you want to retrieve.

view

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/nstimezone/value_value<String>?**view=summary**

Note: By default, the retrieved results are displayed in detail view (?view=detail).

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "nstimezone": [ {

"value":<String_value>,



}]}
```

count

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/nstimezone?**count=yes**

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "nstimezone": [ { "__count": "#no"} ] }
```

nstrafficdomain

Configuration for Traffic Domain resource.

Properties

(click to see Operations)

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
td	<Double>	Read-write	Integer value that uniquely identifies a traffic domain. Minimum value = 1 Maximum value = 4094
aliasname	<String>	Read-write	Name of traffic domain being added. Minimum length = 1

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
			Maximum length = 31
vmac	<String>	Read-write	Associate the traffic domain with a VMAC address instead of with VLANs. The Citrix ADC then sends the VMAC address of the traffic domain in all responses to ARP queries for network entities in that domain. As a result, the ADC can segregate subsequent incoming traffic for this traffic domain on the basis of the destination MAC address, because the destination MAC address is the VMAC address of the traffic domain. After creating entities on a traffic domain, you can easily manage and monitor them by performing traffic domain level operations. Default value: DISABLED Possible values = ENABLED, DISABLED

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
state	<String>	Read-only	The state of TrafficDmain. Default value: ENABLED Possible values = ENABLED, DISABLED
__count	<Double>	Read-only	count parameter

Operations

(click to see Properties)

ADD| DELETE| CLEAR| ENABLE| DISABLE| GET (ALL)| GET| COUNT

Some options that you can use for each operations:

- **Getting warnings in response:**NITRO allows you to get warnings in an operation by specifying the "warning" query parameter as "yes". For example, to get warnings while connecting to the Citrix ADC appliance, the URL is as follows:

  http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/login?warning=yes

  If any, the warnings are displayed in the response payload with the HTTP code "209 X-NITRO-WARNING".

- **Authenticated access for individual NITRO operations:**NITRO allows you to logon to the Citrix ADC appliance to perform individual operations. You can use this option instead of creating a NITRO session (using the login object) and then using that session to perform all operations,

  To do this, you must specify the username and password in the request header of the NITRO request as follows:

  X-NITRO-USER:<username>

  X-NITRO-PASS:<password>

  Note: In such cases, make sure that the request header DOES not include the following:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue>

`*Note:` *

Mandatory parameters are marked in redand placeholder content is marked in <green>.

add

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/nstrafficdomain

HTTP Method:POST

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Content-Type:application/json

Request Payload:

```
{"nstrafficdomain":{

<b>"td":<Double_value>,

</b>"aliasname":<String_value>,

"vmac":<String_value>

}}
```

Response:

HTTP Status Code on Success: 201 Created HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error

delete

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/nstrafficdomain/td_value<Double>

HTTP Method:DELETE

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue>

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error

clear

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/nstrafficdomain?**action=clear**

HTTP Method:POST

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Content-Type:application/json

Request Payload:

```
{"nstrafficdomain":{

<b>"td":<Double_value>

</b>}}
```

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error

enable

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/nstrafficdomain?**action=enable**

HTTP Method:POST

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Content-Type:application/json

Request Payload:

```
{"nstrafficdomain":{

<b>"td":<Double_value>

</b>}}
```

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error

disable

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/nstrafficdomain?**action=disable**

HTTP Method:POST

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Content-Type:application/json

Request Payload:

```
{"nstrafficdomain":{

<b>"td":<Double_value>

</b>}}
```

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error

get (all)

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/nstrafficdomain

Query-parameters:

attrs

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/nstrafficdomain?**attrs=property-name1,property-name2**

Use this query parameter to specify the resource details that you want to retrieve.

filter

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/nstrafficdomain?**filter=property-name1:property-val1,property-name2:property-val2**

Use this query-parameter to get the filtered set of nstrafficdomain resources configured on Citrix ADC. Filtering can be done on any of the properties of the resource.

view

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/nstrafficdomain?**view=summary**

Note: By default, the retrieved results are displayed in detail view (?view=detail).

pagination

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/nstrafficdomain?**pagesize=#no;pageno=#no**

Use this query-parameter to get the nstrafficdomain resources in chunks.

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "nstrafficdomain": [ {

"td":<Double_value>,

"aliasname":<String_value>,

"vmac":<String_value>,

"state":<String_value>

}]}
```

get

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/nstrafficdomain/td_value<Double>

Query-parameters:

attrs

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/nstrafficdomain/td_value<Double>?**attrs=property-name1,property-name2**

Use this query parameter to specify the resource details that you want to retrieve.

view

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/nstrafficdomain/td_value<Double>?**view=summary**

Note: By default, the retrieved results are displayed in detail view (?view=detail).

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "nstrafficdomain": [ {

"td":<Double_value>,

"aliasname":<String_value>,

"vmac":<String_value>,

"state":<String_value>

}]}
```

count

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/nstrafficdomain?**count=yes**

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "nstrafficdomain": [ { "__count": "#no"} ] }
```

nstrafficdomain_binding

Binding object which returns the resources bound to nstrafficdomain.

Properties

(click to see Operations)

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
td	<Double>	Read-write	Integer value that uniquely identifies a traffic domain. Minimum value = 1 Maximum value = 4094
nstrafficdomain_bridgegroup_binding	<nstrafficdomain_bridgegroup_binding[]>	Read-only	bridgegroup that can be bound to nstrafficdomain.
nstrafficdomain_vlan_binding	<nstrafficdomain_vlan_binding[]>	Read-only	vlan that can be bound to nstrafficdomain.
nstrafficdomain_vxlan_binding	<nstrafficdomain_vxlan_binding[]>	Read-only	vxlan that can be bound to nstrafficdomain.

Operations

(click to see Properties)

GET| GET (ALL)

Some options that you can use for each operations:

- **Getting warnings in response:**NITRO allows you to get warnings in an operation by specifying the "warning" query parameter as "yes". For example, to get warnings while connecting to the Citrix ADC appliance, the URL is as follows:

  http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/login?warning=yes

  If any, the warnings are displayed in the response payload with the HTTP code "209 X-NITRO-WARNING".

- **Authenticated access for individual NITRO operations:**NITRO allows you to logon to the Citrix ADC appliance to perform individual operations. You can use this option instead of creating a NITRO session (using the login object) and then using that session to perform all operations,

  To do this, you must specify the username and password in the request header of the NITRO request as follows:

  X-NITRO-USER:<username>

  X-NITRO-PASS:<password>

  Note: In such cases, make sure that the request header DOES not include the following:

  Cookie:NITRO_AUTH_TOKEN=<tokenvalue>

`*Note:`*

Mandatory parameters are marked in redand placeholder content is marked in <green>.

get

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/nstrafficdomain_binding/td_value<Double>

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "nstrafficdomain_binding": [ {

"td":<Double_value>,

"nstrafficdomain_vlan_binding":<nstrafficdomain_vlan_binding[]_value>,

"nstrafficdomain_vxlan_binding":<nstrafficdomain_vxlan_binding[]_value>,

"nstrafficdomain_bridgegroup_binding":<nstrafficdomain_bridgegroup_binding
[]_value>

}]}
```

get (all)

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/nstrafficdomain_binding

Query-parameters:

bulkbindings

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/nstrafficdomain_binding?**bulkbindings=yes**

NITRO allows you to fetch bindings in bulk.

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "nstrafficdomain_binding": [ {

"td":<Double_value>,

"nstrafficdomain_vlan_binding":<nstrafficdomain_vlan_binding[]_value>,

"nstrafficdomain_vxlan_binding":<nstrafficdomain_vxlan_binding[]_value>,

"nstrafficdomain_bridgegroup_binding":<nstrafficdomain_bridgegroup_binding
[]_value>

}]}
```

nstrafficdomain_bridgegroup_binding

Binding object showing the bridgegroup that can be bound to nstrafficdomain.

Properties

(click to see Operations)

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
bridgegroup	<Double>	Read-write	ID of the configured bridge to bind to this traffic domain. More than one bridge group can be bound to a traffic domain, but the same bridge group cannot be a part of multiple traffic domains. Minimum value = 1 Maximum value = 1000

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
td	<Double>	Read-write	Integer value that uniquely identifies a traffic domain. Minimum value = 1 Maximum value = 4094
__count	<Double>	Read-write	count parameter

Operations

(click to see Properties)

ADD:| DELETE:| GET| GET (ALL)| COUNT

Some options that you can use for each operations:

- **Getting warnings in response:**NITRO allows you to get warnings in an operation by specifying the "warning" query parameter as "yes". For example, to get warnings while connecting to the Citrix ADC appliance, the URL is as follows:

  http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/login?warning=yes

  If any, the warnings are displayed in the response payload with the HTTP code "209 X-NITRO-WARNING".
- **Authenticated access for individual NITRO operations:**NITRO allows you to logon to the Citrix ADC appliance to perform individual operations. You can use this option instead of creating a NITRO session (using the login object) and then using that session to perform all operations,

  To do this, you must specify the username and password in the request header of the NITRO request as follows:

  X-NITRO-USER:<username>

  X-NITRO-PASS:<password>

  Note: In such cases, make sure that the request header DOES not include the following:

  Cookie:NITRO_AUTH_TOKEN=<tokenvalue>

`*Note:` *

Mandatory parameters are marked in redand placeholder content is marked in <green>.

add

URL:http://<Citrix-ADC-IP-address(NSIP)/nitro/v1/config/nstrafficdomain_bridgegroup_binding

HTTP Method:PUT

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Content-Type:application/json

Request Payload:

```
{

"nstrafficdomain_bridgegroup_binding":{

<b>"td":<Double_value>,

</b>"bridgegroup":<Double_value>

}}
```

Response:

HTTP Status Code on Success: 201 Created HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error

delete

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/nstrafficdomain_bridgegroup_binding/td_value<Double>

Query-parameters:

args

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/nstrafficdomain_bridgegroup_binding/td_value<Double>?**args=bridgegroup:<Double_value>**

HTTP Method:DELETE

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue>

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error

get

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/nstrafficdomain_bridgegroup_binding/td_value<Double>

Query-parameters:

filter

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/nstrafficdomain_bridgegroup_binding/td_value<Double>?**filter=property-name1:property-value1,property-name2:property-value2**

Use this query-parameter to get the filtered set of nstrafficdomain_bridgegroup_binding resources configured on Citrix ADC. Filtering can be done on any of the properties of the resource.

pagination

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/nstrafficdomain_bridgegroup_binding/td_value<Double>?**pagesize=#no;pageno=#no**

Use this query-parameter to get the nstrafficdomain_bridgegroup_binding resources in chunks.

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "nstrafficdomain_bridgegroup_binding": [ {

"bridgegroup":<Double_value>,

"td":<Double_value>

}]}
```

get (all)

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/nstrafficdomain_bridgegroup_binding

Query-parameters:

bulkbindings

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/nstrafficdomain_bridgegroup_binding?**bulkbindings=yes**

NITRO allows you to fetch bindings in bulk.

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "nstrafficdomain_bridgegroup_binding": [ {

"bridgegroup":<Double_value>,

"td":<Double_value>

}]}
```

count

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/nstrafficdomain_bridgegroup_binding/td_value<Double>?**count=yes**

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{"nstrafficdomain_bridgegroup_binding": [ { "__count": "#no"} ] }
```

nstrafficdomain_vlan_binding

Binding object showing the vlan that can be bound to nstrafficdomain.

Properties

(click to see Operations)

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
td	<Double>	Read-write	Integer value that uniquely identifies a traffic domain. Minimum value = 1 Maximum value = 4094
vlan	<Double>	Read-write	ID of the VLAN to bind to this traffic domain. More than one VLAN can be bound to a traffic domain, but the same VLAN cannot be a part of multiple traffic domains. Minimum value = 1 Maximum value = 4094
__count	<Double>	Read-write	count parameter

Operations

(click to see Properties)

ADD:| DELETE:| GET| GET (ALL)| COUNT

Some options that you can use for each operations:

- **Getting warnings in response:**NITRO allows you to get warnings in an operation by specifying the "warning" query parameter as "yes". For example, to get warnings while connecting to the Citrix ADC appliance, the URL is as follows:

  http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/login?warning=yes

  If any, the warnings are displayed in the response payload with the HTTP code "209 X-NITRO-WARNING".

- **Authenticated access for individual NITRO operations:**NITRO allows you to logon to the Citrix ADC appliance to perform individual operations. You can use this option instead of creating a NITRO session (using the login object) and then using that session to perform all operations,

  To do this, you must specify the username and password in the request header of the NITRO request as follows:

  X-NITRO-USER:<username>

  X-NITRO-PASS:<password>

  Note: In such cases, make sure that the request header DOES not include the following:

  Cookie:NITRO_AUTH_TOKEN=<tokenvalue>

`*Note:`*

Mandatory parameters are marked in redand placeholder content is marked in <green>.

add

URL:http://<Citrix-ADC-IP-address(NSIP)/nitro/v1/config/nstrafficdomain_vlan_binding

HTTP Method:PUT

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Content-Type:application/json

Request Payload:

```
{
```

```
"nstrafficdomain_vlan_binding":{

<b>"td":<Double_value>,

</b>"vlan":<Double_value>

}}
```

Response:

HTTP Status Code on Success: 201 Created HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error

delete

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/nstrafficdomain_vlan_binding/td_value<Double>

Query-parameters:

args

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/nstrafficdomain_vlan_binding/td_value<Double>?**args=vlan:<Double_value>**

HTTP Method:DELETE

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue>

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error

get

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/nstrafficdomain_vlan_binding/td_value<Double>

Query-parameters:

filter

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/nstrafficdomain_vlan_binding/td_value<Double>?**filter=property-name1:property-value1,property-name2:property-value2**

Use this query-parameter to get the filtered set of nstrafficdomain_vlan_binding resources configured on Citrix ADC. Filtering can be done on any of the properties of the resource.

pagination

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/nstrafficdomain_vlan_binding/td_value<Double>?**pagesize=#no;pageno=#no**

Use this query-parameter to get the nstrafficdomain_vlan_binding resources in chunks.

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "nstrafficdomain_vlan_binding": [ {

"td":<Double_value>,

"vlan":<Double_value>

}]}
```

get (all)

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/nstrafficdomain_vlan_binding

Query-parameters:

bulkbindings

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/nstrafficdomain_vlan_binding?**bulkbindings=yes**

NITRO allows you to fetch bindings in bulk.

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "nstrafficdomain_vlan_binding": [ {

"td":<Double_value>,

"vlan":<Double_value>

}]}
```

count

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/nstrafficdomain_vlan_binding/td_value<Double>?**count=yes**

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{"nstrafficdomain_vlan_binding": [ { "__count": "#no"} ] }
```

nstrafficdomain_vxlan_binding

Binding object showing the vxlan that can be bound to nstrafficdomain.

Properties

(click to see Operations)

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
td	<Double>	Read-write	Integer value that uniquely identifies a traffic domain. Minimum value = 1 Maximum value = 4094
vxlan	<Double>	Read-write	ID of the VXLAN to bind to this traffic domain. More than one VXLAN can be bound to a traffic

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
			domain, but the same VXLAN cannot be a part of multiple traffic domains. Minimum value = 1 Maximum value = 16777215
__count	<Double>	Read-write	count parameter

Operations

(click to see Properties)

ADD:| DELETE:| GET| GET (ALL)| COUNT

Some options that you can use for each operations:

- **Getting warnings in response:**NITRO allows you to get warnings in an operation by specifying the "warning" query parameter as "yes". For example, to get warnings while connecting to the Citrix ADC appliance, the URL is as follows:

  http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/login?warning=yes

  If any, the warnings are displayed in the response payload with the HTTP code "209 X-NITRO-WARNING".

- **Authenticated access for individual NITRO operations:**NITRO allows you to logon to the Citrix ADC appliance to perform individual operations. You can use this option instead of creating a NITRO session (using the login object) and then using that session to perform all operations,

  To do this, you must specify the username and password in the request header of the NITRO request as follows:

  X-NITRO-USER:<username>

  X-NITRO-PASS:<password>

  Note: In such cases, make sure that the request header DOES not include the following:

  Cookie:NITRO_AUTH_TOKEN=<tokenvalue>

`*Note:` *

Mandatory parameters are marked in redand placeholder content is marked in <green>.

add

URL:http://<Citrix-ADC-IP-address(NSIP)/nitro/v1/config/nstrafficdomain_vxlan_binding

HTTP Method:PUT

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Content-Type:application/json

Request Payload:

```
{

"nstrafficdomain_vxlan_binding":{

<b>"td":<Double_value>,

</b>"vxlan":<Double_value>

}}
```

Response:

HTTP Status Code on Success: 201 Created HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error

delete

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/nstrafficdomain_vxlan_binding/td_value<Double>

Query-parameters:

args

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/nstrafficdomain_vxlan_binding/td_value<Double>?**args=vxlan:<Double_value>**

HTTP Method:DELETE

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue>

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error

get

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/nstrafficdomain_vxlan_binding/td_value<Double>

Query-parameters:

filter

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/nstrafficdomain_vxlan_binding/td_value<Double>?**filter=property-name1:property-value1,property-name2:property-value2**

Use this query-parameter to get the filtered set of nstrafficdomain_vxlan_binding resources configured on Citrix ADC. Filtering can be done on any of the properties of the resource.

pagination

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/nstrafficdomain_vxlan_binding/td_value<Double>?**pagesize=#no;pageno=#no**

Use this query-parameter to get the nstrafficdomain_vxlan_binding resources in chunks.

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "nstrafficdomain_vxlan_binding": [ {

"td":<Double_value>,

"vxlan":<Double_value>

}]}
```

get (all)

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/nstrafficdomain_vxlan_binding

Query-parameters:

bulkbindings

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/nstrafficdomain_vxlan_binding?**bulkbindings=yes**

NITRO allows you to fetch bindings in bulk.

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "nstrafficdomain_vxlan_binding": [ {

"td":<Double_value>,

"vxlan":<Double_value>

}]}
```

count

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/nstrafficdomain_vxlan_binding/td_value<Double>?**count=yes**

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{"nstrafficdomain_vxlan_binding": [ { "__count": "#no"} ] }
```

nsvariable

Configuration for variable resource.

Properties

(click to see Operations)

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
name	<String>	Read-write	Variable name. This follows the same syntax rules as other expression entity names: It must begin with an alpha character (A-Z or a-z) or an underscore (_). The rest of the characters must be alpha, numeric (0-9) or underscores. It cannot be re or xp (reserved for regular and XPath expressions). It cannot be an expression reserved word (e.g. SYS or HTTP). It cannot be used for an existing expression object (HTTP callout, patset, dataset,

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
			stringmap, or named expression). Minimum length = 1
type	<String>	Read-write	Specification of the variable type; one of the following: ulong - singleton variable with an unsigned 64-bit value. text(value-max-size) - singleton variable with a text string value. map(text(key-max-size),ulong,max-entries) - map of text string keys to unsigned 64-bit values. map(text(key-max-size),text(value-max-size),max-entries) - map of text string keys to text string values. where value-max-size is a positive integer that is the maximum

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
			number of bytes in a text string value. key-max-size is a positive integer that is the maximum number of bytes in a text string key. max-entries is a positive integer that is the maximum number of entries in a map variable. For a global singleton text variable, value-max-size <= 64000. For a global map with ulong values, key-max-size <= 64000. For a global map with text values, key-max-size + value-max-size <= 64000. max-entries is a positive integer that is the maximum number of entries in a map variable. This has a theoretical maximum of

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
			2\^64-1, but in actual use will be much smaller, considering the memory available for use by the map. Example: map(text(10),text(20),100) specifies a map of text string keys (max size 10 bytes) to text string values (max size 20 bytes), with 100 max entries. Minimum length = 1
scope	<String>	Read-write	Scope of the variable: global - (default) one set of values visible across all Packet Engines on a standalone Citrix ADC, an HA pair, or all nodes of a cluster transaction - one value for each request-response transaction (singleton variables only; no expiration).

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
			Default value: global Possible values = global, transaction
iffull	<String>	Read-write	Action to perform if an assignment to a map exceeds its configured max-entries: lru - (default) reuse the least recently used entry in the map. undef - force the assignment to return an undefined (Undef) result to the policy executing the assignment. Default value: lru Possible values = undef, lru
ifvaluetoobig	<String>	Read-write	Action to perform if an value is assigned to a text variable that exceeds its configured max-size, or if a key is used that exceeds its configured max-size:

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
			truncate - (default) truncate the text string to the first max-size bytes and proceed. undef - force the assignment or expression evaluation to return an undefined (Undef) result to the policy executing the assignment or expression. Default value: truncate Possible values = undef, truncate
ifnovalue	<String>	Read-write	Action to perform if on a variable reference in an expression if the variable is single-valued and uninitialized or if the variable is a map and there is no value for the specified key: init - (default) initialize the single-value variable, or create a map entry

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
			for the key and the initial value, using the -init value or its default. undef - force the expression evaluation to return an undefined (Undef) result to the policy executing the expression. Default value: init Possible values = undef, init
init	<String>	Read-write	Initialization value for this variable, to which a singleton variable or map entry will be set if it is referenced before an assignment action has assigned it a value. If the singleton variable or map entry already has been assigned a value, setting this parameter will have no effect on that variable value. Default: 0 for ulong, NULL for text.
expires	<Double>	Read-write	Value expiration in seconds. If the value is not referenced

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
			within the expiration period it will be deleted. 0 (the default) means no expiration. Minimum value = 0 Maximum value = 31622400
comment	<String>	Read-write	Comments associated with this variable.
referencecount	<Double>	Read-only	The number of references to the variable in expressions and assignments.
__count	<Double>	Read-only	count parameter

Operations

(click to see Properties)

ADD| UPDATE| UNSET| DELETE| GET (ALL)| GET| COUNT

Some options that you can use for each operations:

- **Getting warnings in response:**NITRO allows you to get warnings in an operation by specifying the "warning" query parameter as "yes". For example, to get warnings while connecting to the Citrix ADC appliance, the URL is as follows:

  http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/login?warning=yes

  If any, the warnings are displayed in the response payload with the HTTP code "209 X-NITRO-WARNING".
- **Authenticated access for individual NITRO operations:**NITRO allows you to logon to the Citrix ADC appliance to perform individual operations. You can use this option instead of

creating a NITRO session (using the login object) and then using that session to perform all operations,

To do this, you must specify the username and password in the request header of the NITRO request as follows:

X-NITRO-USER:<username>

X-NITRO-PASS:<password>

Note: In such cases, make sure that the request header DOES not include the following:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue>

`*Note:`*

Mandatory parameters are marked in redand placeholder content is marked in <green>.

add

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/nsvariable

HTTP Method:POST

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Content-Type:application/json

Request Payload:

```
{"nsvariable":{

<b>"name":<String_value>,

</b><b>"type":<String_value>,

</b>"scope":<String_value>,

"iffull":<String_value>,

"ifvaluetoobig":<String_value>,

"ifnovalue":<String_value>,
```

```
"init":<String_value>,

"expires":<Double_value>,

"comment":<String_value>

}}
```

Response:

HTTP Status Code on Success: 201 Created HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error

update

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/nsvariable

HTTP Method:PUT

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Content-Type:application/json

Request Payload:

```
{"nsvariable":{

<b>"name":<String_value>,

</b>"type":<String_value>,

"iffull":<String_value>,

"ifvaluetoobig":<String_value>,

"ifnovalue":<String_value>,

"init":<String_value>,
```

```
"expires":<Double_value>,

"comment":<String_value>

}}
```

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error

unset

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/nsvariable?**action=unset**

HTTP Method:POST

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Content-Type:application/json

Request Payload:

```
{"nsvariable":{

<b>"name":<String_value>,

</b>"iffull":true,

"ifvaluetoobig":true,

"ifnovalue":true,

"init":true,

"expires":true,

"comment":true
```

```
}}
```

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error

delete

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/nsvariable/name_value<String>

HTTP Method:DELETE

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue>

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error

get (all)

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/nsvariable

Query-parameters:

attrs

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/nsvariable?**attrs=property-name1,property-name2**

Use this query parameter to specify the resource details that you want to retrieve.

filter

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/nsvariable?**filter=property-name1:property-val1,property-name2:property-val2**

Use this query-parameter to get the filtered set of nsvariable resources configured on Citrix ADC. Filtering can be done on any of the properties of the resource.

view

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/nsvariable?**view=summary**

Note: By default, the retrieved results are displayed in detail view (?view=detail).

pagination

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/nsvariable?**pagesize=#no;pageno=#no**

Use this query-parameter to get the nsvariable resources in chunks.

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "nsvariable": [ {

"name":<String_value>,

"type":<String_value>,

"scope":<String_value>,

"iffull":<String_value>,

"ifvaluetoobig":<String_value>,

"ifnovalue":<String_value>,

"expires":<Double_value>,
```

```
"init":<String_value>,

"comment":<String_value>,

"referencecount":<Double_value>

}]}
```

get

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/nsvariable/name_value<String>

Query-parameters:

attrs

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/nsvariable/name_value<String>?**attrs=property-name1,property-name2**

Use this query parameter to specify the resource details that you want to retrieve.

view

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/nsvariable/name_value<String>?**view=summary**

Note: By default, the retrieved results are displayed in detail view (?view=detail).

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "nsvariable": [ {

"name":<String_value>,

"type":<String_value>,

"scope":<String_value>,

"iffull":<String_value>,

"ifvaluetoobig":<String_value>,

"ifnovalue":<String_value>,

"expires":<Double_value>,

"init":<String_value>,

"comment":<String_value>,

"referencecount":<Double_value>

}]}
```

count

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/nsvariable?**count=yes**

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "nsvariable": [ { "__count": "#no"} ] }
```

nsversion

Configuration for version resource.

Properties

(click to see Operations)

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
installedversion	<Boolean>	Read-write	Installed version.
version	<String>	Read-only	Version.
mode	<Double>	Read-only	Kernel mode (KMPE/VMPE).

Operations

(click to see Properties)

GET (ALL)

Some options that you can use for each operations:

- **Getting warnings in response:**NITRO allows you to get warnings in an operation by specifying the "warning" query parameter as "yes". For example, to get warnings while connecting to the Citrix ADC appliance, the URL is as follows:

  http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/login?warning=yes

  If any, the warnings are displayed in the response payload with the HTTP code "209 X-NITRO-WARNING".
- **Authenticated access for individual NITRO operations:**NITRO allows you to logon to the Citrix ADC appliance to perform individual operations. You can use this option instead of

creating a NITRO session (using the login object) and then using that session to perform all operations,

To do this, you must specify the username and password in the request header of the NITRO request as follows:

X-NITRO-USER:<username>

X-NITRO-PASS:<password>

Note: In such cases, make sure that the request header DOES not include the following:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue>

`*Note:`*

Mandatory parameters are marked in redand placeholder content is marked in <green>.

get (all)

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/nsversion

Query-parameters:

args

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/nsversion?
args=installedversion:<Boolean_value>

Use this query-parameter to get nsversion resources based on additional properties.

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "nsversion": [ {

installedversion:<Boolean_value>"version":<String_value>,

"mode":<Double_value>

}]}
```

nsvpxparam

Configuration for "VPX" resource.

Properties

(click to see Operations)

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
cpuyield	<String>	Read-write	This setting applicable in virtual appliances, is to affect the cpu yield(relinquishing the cpu resources) in any hypervised environment. * There are 3 options for the behavior: 1. YES - Allow the Virtual Appliance to yield its vCPUs periodically, if there is no data traffic.

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
			2. NO - Virtual Appliance will not yield the vCPU. 3. DEFAULT - Restores the default behaviour, according to the license. * Its behavior in different scenarios: 1. As this setting is node specific only, it will not be propagated to other nodes, when executed on Cluster(CLIP) and HA(Primary). 2. In cluster setup, use '-ownerNode' to specify ID of the cluster node. 3. This setting is a system wide implementation and not granular to vCPUs. 4. No effect on the management PE. Possible values = DEFAULT, YES, NO

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
ownernode	<Double>	Read-write	ID of the cluster node for which you are setting the cpuyield. It can be configured only through the cluster IP address. Default value: 255 Minimum value = 0 Maximum value = 31
__count	<Double>	Read-only	count parameter

Operations

(click to see Properties)

UPDATE| GET (ALL)| COUNT

Some options that you can use for each operations:

- **Getting warnings in response:**NITRO allows you to get warnings in an operation by specifying the "warning" query parameter as "yes". For example, to get warnings while connecting to the Citrix ADC appliance, the URL is as follows:

  http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/login?warning=yes

  If any, the warnings are displayed in the response payload with the HTTP code "209 X-NITRO-WARNING".

- **Authenticated access for individual NITRO operations:**NITRO allows you to logon to the Citrix ADC appliance to perform individual operations. You can use this option instead of creating a NITRO session (using the login object) and then using that session to perform all operations,

  To do this, you must specify the username and password in the request header of the NITRO request as follows:

  X-NITRO-USER:<username>

X-NITRO-PASS:<password>

Note: In such cases, make sure that the request header DOES not include the following:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue>

`*Note:`*

Mandatory parameters are marked in redand placeholder content is marked in <green>.

update

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/nsvpxparam

HTTP Method:PUT

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Content-Type:application/json

Request Payload:

```
{"nsvpxparam":{

<b>"cpuyield":<String_value>,

</b>"ownernode":<Double_value>

}}
```

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error

get (all)

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/nsvpxparam

Query-parameters:

attrs

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/nsvpxparam?**attrs=property-name1,property-name2**

Use this query parameter to specify the resource details that you want to retrieve.

filter

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/nsvpxparam?**filter=property-name1:property-val1,property-name2:property-val2**

Use this query-parameter to get the filtered set of nsvpxparam resources configured on Citrix ADC. Filtering can be done on any of the properties of the resource.

view

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/nsvpxparam?**view=summary**

Note: By default, the retrieved results are displayed in detail view (?view=detail).

pagination

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/nsvpxparam?**pagesize=#no;pageno=#no**

Use this query-parameter to get the nsvpxparam resources in chunks.

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "nsvpxparam": [ {
```

```
"cpuyield":<String_value>,

"ownernode":<Double_value>

}]}
```

count

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/nsvpxparam?**count=yes**

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "nsvpxparam": [ { "__count": "#no"} ] }
```

nsweblogparam

Configuration for Web log parameters resource.

Properties

(click to see Operations)

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
buffersizemb	<Double>	Read-write	Buffer size, in MB, allocated for log transaction data on the system. The maximum value is limited to the memory available on the system. Default value: 16 Minimum value = 1 Maximum value = 4294967294LU
customreqhdrs	<String[]>	Read-write	Name(s) of HTTP request headers whose values should be exported by the Web Logging feature. Minimum length = 1
customrsphdrs	<String[]>	Read-write	Name(s) of HTTP response headers whose values should be exported by the Web Logging feature. Minimum length = 1

Operations

(click to see Properties)

UPDATE| UNSET| GET (ALL)

Some options that you can use for each operations:

- **Getting warnings in response:**NITRO allows you to get warnings in an operation by specifying the "warning" query parameter as "yes". For example, to get warnings while connecting to the Citrix ADC appliance, the URL is as follows:

  http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/login?warning=yes

  If any, the warnings are displayed in the response payload with the HTTP code "209 X-NITRO-WARNING".

- **Authenticated access for individual NITRO operations:**NITRO allows you to logon to the Citrix ADC appliance to perform individual operations. You can use this option instead of creating a NITRO session (using the login object) and then using that session to perform all operations,

  To do this, you must specify the username and password in the request header of the NITRO request as follows:

  X-NITRO-USER:<username>

  X-NITRO-PASS:<password>

  Note: In such cases, make sure that the request header DOES not include the following:

  Cookie:NITRO_AUTH_TOKEN=<tokenvalue>

`*Note:`*

Mandatory parameters are marked in redand placeholder content is marked in <green>.

update

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/nsweblogparam

HTTP Method:PUT

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Content-Type:application/json

Request Payload:

```
{"nsweblogparam":{
```

```
"buffersizemb":<Double_value>,

"customreqhdrs":<String[]_value>,

"customrsphdrs":<String[]_value>

}}
```

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error

unset

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/nsweblogparam?**action=unset**

HTTP Method:POST

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Content-Type:application/json

Request Payload:

```
{"nsweblogparam":{

"buffersizemb":true,

"customreqhdrs":true,

"customrsphdrs":true

}}
```

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error

get (all)

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/nsweblogparam

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "nsweblogparam": [ {

"buffersizemb":<Double_value>,

"customreqhdrs":<String[]_value>,

"customrsphdrs":<String[]_value>

}]}
```

nsxmlnamespace

Configuration for XML namespace resource.

Properties

(click to see Operations)

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
prefix	<String>	Read-write	XML prefix. Minimum length = 1
Namespace	<String>	Read-write	Expanded namespace for which the XML prefix is provided. Minimum length = 1
description	<String>	Read-write	Description for the prefix. Minimum length = 1
__count	<Double>	Read-only	count parameter

Operations

(click to see Properties)

ADD| DELETE| UPDATE| UNSET| GET (ALL)| GET| COUNT

Some options that you can use for each operations:

- **Getting warnings in response:**NITRO allows you to get warnings in an operation by specifying the "warning" query parameter as "yes". For example, to get warnings while connecting to the Citrix ADC appliance, the URL is as follows:

  http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/login?warning=yes

  If any, the warnings are displayed in the response payload with the HTTP code "209 X-NITRO-WARNING".

- **Authenticated access for individual NITRO operations:**NITRO allows you to logon to the Citrix ADC appliance to perform individual operations. You can use this option instead of creating a NITRO session (using the login object) and then using that session to perform all operations,

  To do this, you must specify the username and password in the request header of the NITRO request as follows:

  X-NITRO-USER:<username>

X-NITRO-PASS:<password>

Note: In such cases, make sure that the request header DOES not include the following:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue>

`*Note:`*

Mandatory parameters are marked in redand placeholder content is marked in <green>.

add

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/nsxmlnamespace

HTTP Method:POST

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Content-Type:application/json

Request Payload:

```
{"nsxmlnamespace":{

<b>"prefix":<String_value>,

</b><b>"Namespace":<String_value>,

</b>"description":<String_value>

}}
```

Response:

HTTP Status Code on Success: 201 Created HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error

delete

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/nsxmlnamespace/prefix_value<String>

HTTP Method:DELETE

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue>

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error

update

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/nsxmlnamespace

HTTP Method:PUT

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Content-Type:application/json

Request Payload:

```
{"nsxmlnamespace":{

<b>"prefix":<String_value>,

</b>"Namespace":<String_value>,

"description":<String_value>

}}
```

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error

unset

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/nsxmlnamespace?**action=unset**

HTTP Method:POST

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Content-Type:application/json

Request Payload:

```
{"nsxmlnamespace":{

<b>"prefix":<String_value>,

</b>"Namespace":true,

"description":true

}}
```

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error

get (all)

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/nsxmlnamespace

Query-parameters:

attrs

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/nsxmlnamespace?**attrs=property-name1,property-name2**

Use this query parameter to specify the resource details that you want to retrieve.

filter

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/nsxmlnamespace?**filter=property-name1:property-val1,property-name2:property-val2**

Use this query-parameter to get the filtered set of nsxmlnamespace resources configured on Citrix ADC. Filtering can be done on any of the properties of the resource.

view

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/nsxmlnamespace?**view=summary**

Note: By default, the retrieved results are displayed in detail view (?view=detail).

pagination

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/nsxmlnamespace?**pagesize=#no;pageno=#no**

Use this query-parameter to get the nsxmlnamespace resources in chunks.

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "nsxmlnamespace": [ {

"prefix":<String_value>,

"Namespace":<String_value>,

"description":<String_value>

}]}
```

get

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/nsxmlnamespace/prefix_value<String>

Query-parameters:

attrs

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/nsxmlnamespace/prefix_value<String>?**attrs=property-name1,property-name2**

Use this query parameter to specify the resource details that you want to retrieve.

view

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/nsxmlnamespace/prefix_value<String>?**view=summary**

Note: By default, the retrieved results are displayed in detail view (?view=detail).

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "nsxmlnamespace": [ {

"prefix":<String_value>,

"Namespace":<String_value>,
```

```
"description":<String_value>

}]}
```

count

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/nsxmlnamespace?**count=yes**

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "nsxmlnamespace": [ { "__count": "#no"} ] }
```

reboot

Configuration for 0 resource.

Properties

(click to see Operations)

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
warm	<Boolean>	Read-write	Restarts the Citrix ADC software without rebooting the

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
			underlying operating system. The session terminates and you must log on to the appliance after it has restarted. Note: This argument is required only for nCore appliances. Classic appliances ignore this argument.

Operations

(click to see Properties)

REBOOT

Some options that you can use for each operations:

- **Getting warnings in response:**NITRO allows you to get warnings in an operation by specifying the "warning" query parameter as "yes". For example, to get warnings while connecting to the Citrix ADC appliance, the URL is as follows:

  http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/login?warning=yes

  If any, the warnings are displayed in the response payload with the HTTP code "209 X-NITRO-WARNING".

- **Authenticated access for individual NITRO operations:**NITRO allows you to logon to the Citrix ADC appliance to perform individual operations. You can use this option instead of creating a NITRO session (using the login object) and then using that session to perform all operations,

  To do this, you must specify the username and password in the request header of the NITRO request as follows:

  X-NITRO-USER:<username>

X-NITRO-PASS:<password>

Note: In such cases, make sure that the request header DOES not include the following:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue>

`*Note:`*

Mandatory parameters are marked in redand placeholder content is marked in <green>.

Reboot

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/reboot

HTTP Method:POST

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Content-Type:application/json

Request Payload:

```
{"reboot":{

"warm":<Boolean_value>

}}
```

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error

shutdown

Configuration for 0 resource.

Operations

(click to see Properties)

SHUTDOWN

Some options that you can use for each operations:

- **Getting warnings in response:**NITRO allows you to get warnings in an operation by specifying the "warning" query parameter as "yes". For example, to get warnings while connecting to the Citrix ADC appliance, the URL is as follows:

  http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/login?warning=yes

  If any, the warnings are displayed in the response payload with the HTTP code "209 X-NITRO-WARNING".

- **Authenticated access for individual NITRO operations:**NITRO allows you to logon to the Citrix ADC appliance to perform individual operations. You can use this option instead of creating a NITRO session (using the login object) and then using that session to perform all operations,

  To do this, you must specify the username and password in the request header of the NITRO request as follows:

  X-NITRO-USER:<username>

  X-NITRO-PASS:<password>

  Note: In such cases, make sure that the request header DOES not include the following:

  Cookie:NITRO_AUTH_TOKEN=<tokenvalue>

`*Note:`*

Mandatory parameters are marked in redand placeholder content is marked in <green>.

Shutdown

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/shutdown

HTTP Method:POST

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Content-Type:application/json

Request Payload:

```
{"shutdown":{

}}
```

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error

NTP

NTP

NTP configuration.

NAME	DESCRIPTION
ntpparam	Configuration for NTP parameter
ntpserver	Configuration for NTP server
ntpstatus	Configuration for ntp status
ntpsync	Configuration for NTP sync

ntpparam

Configuration for NTP parameter resource.

Properties

(click to see Operations)

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
authentication	<String>	Read-write	Apply NTP authentication, which enables the NTP

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
			client (Citrix ADC) to verify that the server is in fact known and trusted. Default value: YES Possible values = YES, NO
trustedkey	<Double[]>	Read-write	Key identifiers that are trusted for server authentication with symmetric key cryptography in the keys file. Minimum value = 1 Maximum value = 65534
autokeylogsec	<Double>	Read-write	Autokey protocol requires the keys to be refreshed periodically. This parameter specifies the interval between regenerations of new session keys. In seconds, expressed as a power of 2. Default value: 12 Minimum value = 0 Maximum value = 32
revokelogsec	<Double>	Read-write	Interval between re-randomizations of the autokey seeds to prevent brute-force

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
			attacks on the autokey algorithms. Default value: 16 Minimum value = 0 Maximum value = 32

Operations

(click to see Properties)

UPDATE| UNSET| GET (ALL)

Some options that you can use for each operations:

- **Getting warnings in response:**NITRO allows you to get warnings in an operation by specifying the "warning" query parameter as "yes". For example, to get warnings while connecting to the Citrix ADC appliance, the URL is as follows:

  http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/login?warning=yes

  If any, the warnings are displayed in the response payload with the HTTP code "209 X-NITRO-WARNING".

- **Authenticated access for individual NITRO operations:**NITRO allows you to logon to the Citrix ADC appliance to perform individual operations. You can use this option instead of creating a NITRO session (using the login object) and then using that session to perform all operations,

  To do this, you must specify the username and password in the request header of the NITRO request as follows:

  X-NITRO-USER:<username>

  X-NITRO-PASS:<password>

  Note: In such cases, make sure that the request header DOES not include the following:

  Cookie:NITRO_AUTH_TOKEN=<tokenvalue>

`*Note:`*

Mandatory parameters are marked in redand placeholder content is marked in <green>.

update

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/ntpparam

HTTP Method:PUT

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Content-Type:application/json

Request Payload:

```
{"ntpparam":{

"authentication":<String_value>,

"trustedkey":<Double[]_value>,

"autokeylogsec":<Double_value>,

"revokelogsec":<Double_value>

}}
```

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error

unset

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/ntpparam?**action=unset**

HTTP Method:POST

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Content-Type:application/json

Request Payload:

```
{"ntpparam":{

"authentication":true,

"trustedkey":true,

"autokeylogsec":true,

"revokelogsec":true

}}
```

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error

get (all)

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/ntpparam

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "ntpparam": [ {
"authentication":<String_value>,
"trustedkey":<Double[]_value>,
"autokeylogsec":<Double_value>,
"revokelogsec":<Double_value>
}]}
```

ntpserver

Configuration for NTP server resource.

Properties

(click to see Operations)

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
serverip	<String>	Read-write	IP address of the NTP server. Minimum length = 1
servername	<String>	Read-write	Fully qualified domain name of the NTP server.
minpoll	<Double>	Read-write	Minimum time after which the NTP server must poll the NTP messages. In seconds, expressed as a power of 2. Minimum value = 4

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
			Maximum value = 17
maxpoll	<Double>	Read-write	Maximum time after which the NTP server must poll the NTP messages. In seconds, expressed as a power of 2. Minimum value = 4 Maximum value = 17
autokey	<Boolean>	Read-write	Use the Autokey protocol for key management for this server, with the cryptographic values (for example, symmetric key, host and public certificate files, and sign key) generated by the ntp-keygen utility. To require authentication for communication with the server, you must set either the value of this parameter or the key parameter.
key	<Double>	Read-write	Key to use for encrypting authentication fields. All packets sent to and received from the server must include authentication fields

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
			encrypted by using this key. To require authentication for communication with the server, you must set either the value of this parameter or the autokey parameter. Minimum value = 1 Maximum value = 65534
preferredntpserver	<String>	Read-write	Preferred NTP server. The Citrix ADC chooses this NTP server for time synchronization among a set of correctly operating hosts. Default value: NO Possible values = YES, NO
__count	<Double>	Read-only	count parameter

Operations

(click to see Properties)

ADD| DELETE| UPDATE| UNSET| GET (ALL)| COUNT

Some options that you can use for each operations:

- **Getting warnings in response:**NITRO allows you to get warnings in an operation by specifying the "warning" query parameter as "yes". For example, to get warnings while connecting to the Citrix ADC appliance, the URL is as follows:

  http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/login?warning=yes

If any, the warnings are displayed in the response payload with the HTTP code "209 X-NITRO-WARNING".

- **Authenticated access for individual NITRO operations:**NITRO allows you to logon to the Citrix ADC appliance to perform individual operations. You can use this option instead of creating a NITRO session (using the login object) and then using that session to perform all operations,

  To do this, you must specify the username and password in the request header of the NITRO request as follows:

  X-NITRO-USER:<username>

  X-NITRO-PASS:<password>

  Note: In such cases, make sure that the request header DOES not include the following:

  Cookie:NITRO_AUTH_TOKEN=<tokenvalue>

`*Note:`*

Mandatory parameters are marked in redand placeholder content is marked in <green>.

add

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/ntpserver

HTTP Method:POST

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Content-Type:application/json

Request Payload:

```
{"ntpserver":{

"serverip":<String_value>,

"servername":<String_value>,

"minpoll":<Double_value>,
```

```
"maxpoll":<Double_value>,

"autokey":<Boolean_value>,

"key":<Double_value>

}}
```

Response:

HTTP Status Code on Success: 201 Created HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error

delete

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/ntpserver/serverip_value<String>

HTTP Method:DELETE

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue>

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error

update

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/ntpserver

HTTP Method:PUT

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Content-Type:application/json

Request Payload:

```
{"ntpserver":{

"serverip":<String_value>,

"servername":<String_value>,

"minpoll":<Double_value>,

"maxpoll":<Double_value>,

"preferredntpserver":<String_value>,

"autokey":<Boolean_value>,

"key":<Double_value>

}}
```

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error

unset

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/ntpserver?**action=unset**

HTTP Method:POST

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Content-Type:application/json

Request Payload:

```
{"ntpserver":{
```

```
"serverip":<String_value>,

"servername":<String_value>,

"autokey":true,

"minpoll":true,

"maxpoll":true,

"preferredntpserver":true,

"key":true

}}
```

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error

get (all)

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/ntpserver

Query-parameters:

attrs

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/ntpserver?**attrs=property-name1,property-name2**

Use this query parameter to specify the resource details that you want to retrieve.

filter

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/ntpserver?**filter=property-name1:property-val1,property-name2:property-val2**

Use this query-parameter to get the filtered set of ntpserver resources configured on Citrix ADC. Filtering can be done on any of the properties of the resource.

view

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/ntpserver?**view=summary**

Note: By default, the retrieved results are displayed in detail view (?view=detail).

pagination

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/ntpserver?**pagesize=#no;pageno=#no**

Use this query-parameter to get the ntpserver resources in chunks.

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "ntpserver": [ {

"serverip":<String_value>,

"servername":<String_value>,

"minpoll":<Double_value>,

"maxpoll":<Double_value>,

"preferredntpserver":<String_value>,

"autokey":<Boolean_value>,

"key":<Double_value>
```

```
}]}
```

count

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/ntpserver?**count=yes**

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "ntpserver": [ { "__count": "#no"} ] }
```

ntpstatus

Configuration for ntp status resource.

Properties

(click to see Operations)

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
response	<String>	Read-only	.

Operations

(click to see Properties)

GET (ALL)

Some options that you can use for each operations:

- **Getting warnings in response:**NITRO allows you to get warnings in an operation by specifying the "warning" query parameter as "yes". For example, to get warnings while connecting to the Citrix ADC appliance, the URL is as follows:

  http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/login?warning=yes

  If any, the warnings are displayed in the response payload with the HTTP code "209 X-NITRO-WARNING".

- **Authenticated access for individual NITRO operations:**NITRO allows you to logon to the Citrix ADC appliance to perform individual operations. You can use this option instead of creating a NITRO session (using the login object) and then using that session to perform all operations,

  To do this, you must specify the username and password in the request header of the NITRO request as follows:

  X-NITRO-USER:<username>

  X-NITRO-PASS:<password>

  Note: In such cases, make sure that the request header DOES not include the following:

  Cookie:NITRO_AUTH_TOKEN=<tokenvalue>

`*Note:`*

Mandatory parameters are marked in redand placeholder content is marked in <green>.

get (all)

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/ntpstatus

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "ntpstatus": [ {

"response":<String_value>

}]}
```

ntpsync

Configuration for NTP sync resource.

Properties

(click to see Operations)

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
state	<String>	Read-only	Show NTP status. Possible values = ENABLED, DISABLED

Operations

(click to see Properties)

ENABLE| DISABLE| GET (ALL)

Some options that you can use for each operations:

- **Getting warnings in response:**NITRO allows you to get warnings in an operation by specifying the "warning" query parameter as "yes". For example, to get warnings while connecting to the Citrix ADC appliance, the URL is as follows:

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/login?warning=yes

If any, the warnings are displayed in the response payload with the HTTP code "209 X-NITRO-WARNING".

- **Authenticated access for individual NITRO operations:**NITRO allows you to logon to the Citrix ADC appliance to perform individual operations. You can use this option instead of creating a NITRO session (using the login object) and then using that session to perform all operations,

  To do this, you must specify the username and password in the request header of the NITRO request as follows:

  X-NITRO-USER:<username>

  X-NITRO-PASS:<password>

  Note: In such cases, make sure that the request header DOES not include the following:

  Cookie:NITRO_AUTH_TOKEN=<tokenvalue>

`*Note:`*

Mandatory parameters are marked in redand placeholder content is marked in <green>.

enable

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/ntpsync?**action=enable**

HTTP Method:POST

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Content-Type:application/json

Request Payload:

```
{"ntpsync":{

}}
```

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error

disable

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/ntpsync?**action=disable**

HTTP Method:POST

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Content-Type:application/json

Request Payload:

```
{"ntpsync":{

}}
```

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error

get (all)

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/ntpsync

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "ntpsync": [ {

"state":<String_value>

}]}
```

PCP

Pcp

Port Control Protocol Configuration.

NAME	DESCRIPTION
pcpmap	Configuration for server
pcpprofile	Configuration for PCP Profile
pcpserver	Configuration for server

pcpmap

Configuration for server resource.

Properties

(click to see Operations)

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
nattype	<String>	Read-write	Type of sessions to be displayed. Default value: NAT44

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
			Possible values = NAT44, DS-Lite, NAT64
pcpsrcip	<String>	Read-only	Internal IP address.
subscrip	<String>	Read-only	IPV4 src address of Client in DS-Lite.
pcpsrcport	<Integer>	Read-only	Internal Port. Range 1 - 65535 * in CLI is represented as 65535 in NITRO API
pcpdstip	<String>	Read-only	Remote PEER IP address.
pcpdstport	<Integer>	Read-only	Remote PEER Port. Range 1 - 65535 * in CLI is represented as 65535 in NITRO API
pcpnatip	<String>	Read-only	External IP address.
pcpnatport	<Integer>	Read-only	External Port Number. Range 1 - 65535 * in CLI is represented as 65535 in NITRO API

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
pcpprotocol	<String>	Read-only	Protocol of the PCP mapping. Possible values = HopOpt, icmp, igmp, ggp, ipv4, st, tcp, udp
pcpaddr	<Double>	Read-only	Address of PCP Map entity.
pcpnounce	<Double>	Read-only	PCP Mapping Nounce.
pcprefcnt	<Double>	Read-only	Refcount, which indicates how many session is active.
pcplifetime	<Double>	Read-only	Integer value that shows the requested lifetime.
__count	<Double>	Read-only	count parameter

Operations

(click to see Properties)

GET (ALL)| COUNT

Some options that you can use for each operations:

- **Getting warnings in response:**NITRO allows you to get warnings in an operation by specifying the "warning" query parameter as "yes". For example, to get warnings while connecting to the Citrix ADC appliance, the URL is as follows:

  http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/login?warning=yes

  If any, the warnings are displayed in the response payload with the HTTP code "209 X-NITRO-WARNING".

- **Authenticated access for individual NITRO operations:**NITRO allows you to logon to the Citrix ADC appliance to perform individual operations. You can use this option instead of creating a NITRO session (using the login object) and then using that session to perform all operations,

  To do this, you must specify the username and password in the request header of the NITRO request as follows:

  X-NITRO-USER:<username>

  X-NITRO-PASS:<password>

  Note: In such cases, make sure that the request header DOES not include the following:

  Cookie:NITRO_AUTH_TOKEN=<tokenvalue>

`*Note:`*

Mandatory parameters are marked in redand placeholder content is marked in <green>.

get (all)

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/pcpmap

Query-parameters:

args

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/pcpmap?**args=nattype:<String_value>**

Use this query-parameter to get pcpmap resources based on additional properties.

attrs

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/pcpmap?**attrs=property-name1,property-name2**

Use this query parameter to specify the resource details that you want to retrieve.

filter

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/pcpmap?**filter=property-name1:property-val1,property-name2:property-val2**

Use this query-parameter to get the filtered set of pcpmap resources configured on Citrix ADC. Filtering can be done on any of the properties of the resource.

view

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/pcpmap?**view=summary**

Note: By default, the retrieved results are displayed in detail view (?view=detail).

pagination

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/pcpmap?**pagesize=#no;pageno=#no**

Use this query-parameter to get the pcpmap resources in chunks.

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "pcpmap": [ {

nattype:<String_value>"pcpsrcip":<String_value>,

"subscrip":<String_value>,

"pcpsrcport":<Integer_value>,

"pcpdstip":<String_value>,

"pcpdstport":<Integer_value>,

"pcpnatip":<String_value>,

"pcpnatport":<Integer_value>,
```

```
"pcpprotocol":<String_value>,

"pcpaddr":<Double_value>,

"pcpnounce":<Double_value>,

"pcprefcnt":<Double_value>,

"pcplifetime":<Double_value>

}]}
```

count

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/pcpmap?**count=yes**

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "pcpmap": [ { "__count": "#no"} ] }
```

pcpprofile

Configuration for PCP Profile resource.

Properties

(click to see Operations)

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
name	<String>	Read-write	Name for the PCP Profile. Must begin with an ASCII alphanumeric or underscore (_) character, and must contain only ASCII alphanumeric, underscore CLI Users: If the name includes one or more spaces, enclose the name in double or single quotation marks (for example, "my pcpProfile" or my pcpProfile).
mapping	<String>	Read-write	This argument is for enabling/disabling the MAP opcode of current PCP Profile. Default value: ENABLED Possible values = ENABLED, DISABLED
peer	<String>	Read-write	This argument is for enabling/disabling the PEER opcode of current PCP Profile. Default value: ENABLED

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
			Possible values = ENABLED, DISABLED
minmaplife	<Double>	Read-write	Integer value that identify the minimum mapping lifetime (in seconds) for a pcp profile. default(120s). Minimum value = 1 Maximum value = 2147483647
maxmaplife	<Double>	Read-write	Integer value that identify the maximum mapping lifetime (in seconds) for a pcp profile. default(86400s = 24Hours). Minimum value = 1 Maximum value = 2147483647
announcemulticount	<Double>	Read-write	Integer value that identify the number announce message to be send. Default value: 10 Minimum value = 0 Maximum value = 65535
thirdparty	<String>	Read-write	This argument is for enabling/disabling the THIRD PARTY

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
			opcode of current PCP Profile. Default value: DISABLED Possible values = ENABLED, DISABLED
__count	<Double>	Read-only	count parameter

Operations

(click to see Properties)

ADD| DELETE| UPDATE| UNSET| GET (ALL)| GET| COUNT

Some options that you can use for each operations:

- **Getting warnings in response:**NITRO allows you to get warnings in an operation by specifying the "warning" query parameter as "yes". For example, to get warnings while connecting to the Citrix ADC appliance, the URL is as follows:

  http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/login?warning=yes

  If any, the warnings are displayed in the response payload with the HTTP code "209 X-NITRO-WARNING".

- **Authenticated access for individual NITRO operations:**NITRO allows you to logon to the Citrix ADC appliance to perform individual operations. You can use this option instead of creating a NITRO session (using the login object) and then using that session to perform all operations,

  To do this, you must specify the username and password in the request header of the NITRO request as follows:

  X-NITRO-USER:<username>

  X-NITRO-PASS:<password>

  Note: In such cases, make sure that the request header DOES not include the following:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue>

`*Note:` *

Mandatory parameters are marked in redand placeholder content is marked in <green>.

add

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/pcpprofile

HTTP Method:POST

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Content-Type:application/json

Request Payload:

```
{"pcpprofile":{

<b>"name":<String_value>,

</b>"mapping":<String_value>,

"peer":<String_value>,

"minmaplife":<Double_value>,

"maxmaplife":<Double_value>,

"announcemulticount":<Double_value>,

"thirdparty":<String_value>

}}
```

Response:

HTTP Status Code on Success: 201 Created HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error

delete

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/pcpprofile/name_value<String>

HTTP Method:DELETE

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue>

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error

update

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/pcpprofile

HTTP Method:PUT

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Content-Type:application/json

Request Payload:

```
{"pcpprofile":{

<b>"name":<String_value>,

</b>"mapping":<String_value>,

"peer":<String_value>,

"minmaplife":<Double_value>,

"maxmaplife":<Double_value>,

"announcemulticount":<Double_value>,

"thirdparty":<String_value>
```

```
}}
```

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error

unset

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/pcpprofile?**action=unset**

HTTP Method:POST

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Content-Type:application/json

Request Payload:

```
{"pcpprofile":{

<b>"name":<String_value>,

</b>"mapping":true,

"peer":true,

"minmaplife":true,

"maxmaplife":true,

"announcemulticount":true,

"thirdparty":true

}}
```

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error

get (all)

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/pcpprofile

Query-parameters:

attrs

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/pcpprofile?**attrs=property-name1,property-name2**

Use this query parameter to specify the resource details that you want to retrieve.

filter

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/pcpprofile?**filter=property-name1:property-val1,property-name2:property-val2**

Use this query-parameter to get the filtered set of pcpprofile resources configured on Citrix ADC. Filtering can be done on any of the properties of the resource.

view

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/pcpprofile?**view=summary**

Note: By default, the retrieved results are displayed in detail view (?view=detail).

pagination

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/pcpprofile?**pagesize=#no;pageno=#no**

Use this query-parameter to get the pcpprofile resources in chunks.

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "pcpprofile": [ {

"name":<String_value>,

"mapping":<String_value>,

"peer":<String_value>,

"minmaplife":<Double_value>,

"maxmaplife":<Double_value>,

"announcemulticount":<Double_value>,

"thirdparty":<String_value>

}]}
```

get

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/pcpprofile/name_value<String>

Query-parameters:

attrs

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/pcpprofile/name_value<String>?**attrs=property-name1,property-name2**

Use this query parameter to specify the resource details that you want to retrieve.

view

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/pcpprofile/name_value<String>?**view=summary**

Note: By default, the retrieved results are displayed in detail view (?view=detail).

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "pcpprofile": [ {

"name":<String_value>,

"mapping":<String_value>,

"peer":<String_value>,

"minmaplife":<Double_value>,

"maxmaplife":<Double_value>,

"announcemulticount":<Double_value>,

"thirdparty":<String_value>

}]}
```

count

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/pcpprofile?**count=yes**

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "pcpprofile": [ { "__count": "#no"} ] }
```

pcpserver

Configuration for server resource.

Properties

(click to see Operations)

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
name	<String>	Read-write	Name for the PCP server. Must begin with an ASCII alphanumeric or underscore (_) character, and must contain only ASCII alphanumeric, underscore CLI

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
			Users: If the name includes one or more spaces, enclose the name in double or single quotation marks (for example, "my pcpServer" or my pcpServer).
ipaddress	<String>	Read-write	The IP address of the PCP server.
port	<Integer>	Read-write	Port number for the PCP server. Default value: 5351 Range 1 - 65535 * in CLI is represented as 65535 in NITRO API
pcpprofile	<String>	Read-write	pcp profile name.
__count	<Double>	Read-only	count parameter

Operations

(click to see Properties)

ADD| DELETE| UPDATE| UNSET| GET (ALL)| GET| COUNT

Some options that you can use for each operations:

- **Getting warnings in response:**NITRO allows you to get warnings in an operation by specifying the "warning" query parameter as "yes". For example, to get warnings while connecting to the Citrix ADC appliance, the URL is as follows:

  http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/login?warning=yes

If any, the warnings are displayed in the response payload with the HTTP code "209 X-NITRO-WARNING".

- **Authenticated access for individual NITRO operations:**NITRO allows you to logon to the Citrix ADC appliance to perform individual operations. You can use this option instead of creating a NITRO session (using the login object) and then using that session to perform all operations,

  To do this, you must specify the username and password in the request header of the NITRO request as follows:

  X-NITRO-USER:<username>

  X-NITRO-PASS:<password>

  Note: In such cases, make sure that the request header DOES not include the following:

  Cookie:NITRO_AUTH_TOKEN=<tokenvalue>

`*Note:`*

Mandatory parameters are marked in redand placeholder content is marked in <green>.

add

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/pcpserver

HTTP Method:POST

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Content-Type:application/json

Request Payload:

```
{"pcpserver":{

<b>"name":<String_value>,

</b><b>"ipaddress":<String_value>,

</b>"port":<Integer_value>,
```

```
"pcpprofile":<String_value>

}}
```

Response:

HTTP Status Code on Success: 201 Created HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error

delete

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/pcpserver/name_value<String>

HTTP Method:DELETE

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue>

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error

update

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/pcpserver

HTTP Method:PUT

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Content-Type:application/json

Request Payload:

```
{"pcpserver":{

<b>"name":<String_value>,
```

```
</b>"port":<Integer_value>,

"pcpprofile":<String_value>

}}
```

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error

unset

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/pcpserver?**action=unset**

HTTP Method:POST

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Content-Type:application/json

Request Payload:

```
{"pcpserver":{

<b>"name":<String_value>,

</b>"port":true,

"pcpprofile":true

}}
```

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error

get (all)

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/pcpserver

Query-parameters:

attrs

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/pcpserver?**attrs=property-name1,property-name2**

Use this query parameter to specify the resource details that you want to retrieve.

filter

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/pcpserver?**filter=property-name1:property-val1,property-name2:property-val2**

Use this query-parameter to get the filtered set of pcpserver resources configured on Citrix ADC. Filtering can be done on any of the properties of the resource.

view

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/pcpserver?**view=summary**

Note: By default, the retrieved results are displayed in detail view (?view=detail).

pagination

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/pcpserver?**pagesize=#no;pageno=#no**

Use this query-parameter to get the pcpserver resources in chunks.

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "pcpserver": [ {

"name":<String_value>,

"ipaddress":<String_value>,

"port":<Integer_value>,

"pcpprofile":<String_value>

}]}
```

get

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/pcpserver/name_value<String>

Query-parameters:

attrs

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/pcpserver/name_value<String>?**attrs=property-name1,property-name2**

Use this query parameter to specify the resource details that you want to retrieve.

view

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/pcpserver/name_value<String>?**view=summary**

Note: By default, the retrieved results are displayed in detail view (?view=detail).

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "pcpserver": [ {

"name":<String_value>,

"ipaddress":<String_value>,

"port":<Integer_value>,

"pcpprofile":<String_value>

}]}
```

count

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/pcpserver?**count=yes**

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "pcpserver": [ { "__count": "#no"} ] }
```

Policy

Policy

Policy configuration.

NAME	DESCRIPTION
policydataset	Configuration for TYPE set
policydataset_binding	Binding object showing the resources that can be bound to policydataset
policydataset_value_binding	Binding object showing the value that can be bound to policydataset
policyevaluation	Configuration for expr evaluation
policyexpression	Configuration for expression
policyhttpcallout	Configuration for HTTP callout
policymap	Configuration for map policy
policyparam	Configuration for policy parameter
policypatset	Configuration for PAT set
policypatset_binding	Binding object showing the resources that can be bound to policypatset
policypatset_pattern_binding	Binding object showing the pattern that can be bound to policypatset
policystringmap	Configuration for string map

NAME	DESCRIPTION
policystringmap_binding	Binding object showing the resources that can be bound to policystringmap
policystringmap_pattern_binding	Binding object showing the pattern that can be bound to policystringmap
policyurlset	Configuration for URL set

policydataset

Configuration for TYPE set resource.

Properties

(click to see Operations)

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
name	<String>	Read-write	Name of the dataset. Must not exceed 127 characters. Minimum length = 1
type	<String>	Read-write	Type of value to bind to the dataset. Possible values = ipv4, number, ipv6, ulong, double, mac
indextype	<String>	Read-write	Index type. Possible values = Auto-generated, User-defined
comment	<String>	Read-write	Any comments to preserve information about this dataset.
description	<String>	Read-only	Description of the set.

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
__count	<Double>	Read-only	count parameter

Operations

(click to see Properties)

ADD| DELETE| GET (ALL)| GET| COUNT

Some options that you can use for each operations:

- **Getting warnings in response:**NITRO allows you to get warnings in an operation by specifying the "warning" query parameter as "yes". For example, to get warnings while connecting to the Citrix ADC appliance, the URL is as follows:

  http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/login?warning=yes

  If any, the warnings are displayed in the response payload with the HTTP code "209 X-NITRO-WARNING".
- **Authenticated access for individual NITRO operations:**NITRO allows you to logon to the Citrix ADC appliance to perform individual operations. You can use this option instead of creating a NITRO session (using the login object) and then using that session to perform all operations,

  To do this, you must specify the username and password in the request header of the NITRO request as follows:

  X-NITRO-USER:<username>

  X-NITRO-PASS:<password>

  Note: In such cases, make sure that the request header DOES not include the following:

  Cookie:NITRO_AUTH_TOKEN=<tokenvalue>

`*Note:`*

Mandatory parameters are marked in redand placeholder content is marked in <green>.

add

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/policydataset

HTTP Method:POST

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Content-Type:application/json

Request Payload:

```
{"policydataset":{

<b>"name":<String_value>,

</b><b>"type":<String_value>,

</b>"indextype":<String_value>,

"comment":<String_value>

}}
```

Response:

HTTP Status Code on Success: 201 Created HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error

delete

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/policydataset/name_value<String>

HTTP Method:DELETE

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue>

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error

get (all)

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/policydataset

Query-parameters:

attrs

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/policydataset?**attrs=property-name1,property-name2**

Use this query parameter to specify the resource details that you want to retrieve.

filter

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/policydataset?**filter=property-name1:property-val1,property-name2:property-val2**

Use this query-parameter to get the filtered set of policydataset resources configured on Citrix ADC. Filtering can be done on any of the properties of the resource.

view

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/policydataset?**view=summary**

Note: By default, the retrieved results are displayed in detail view (?view=detail).

pagination

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/policydataset?**pagesize=#no;pageno=#no**

Use this query-parameter to get the policydataset resources in chunks.

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "policydataset": [ {

"name":<String_value>,

"description":<String_value>,

"type":<String_value>,

"indextype":<String_value>,

"comment":<String_value>

}]}
```

get

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/policydataset/name_value<String>

Query-parameters:

attrs

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/policydataset/name_value<String>?**attrs=property-name1,property-name2**

Use this query parameter to specify the resource details that you want to retrieve.

view

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/policydataset/name_value<String>?**view=summary**

Note: By default, the retrieved results are displayed in detail view (?view=detail).

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "policydataset": [ {

"name":<String_value>,

"description":<String_value>,

"type":<String_value>,

"indextype":<String_value>,

"comment":<String_value>

}]}
```

count

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/policydataset?**count=yes**

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "policydataset": [ { "__count": "#no"} ] }
```

policydataset_binding

Binding object which returns the resources bound to policydataset.

Properties

(click to see Operations)

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
name	<String>	Read-write	Name of the dataset. Must not exceed 127 characters. Minimum length = 1
policydataset_value_binding	<policydataset_value_binding[]>	Read-only	value that can be bound to policydataset.

Operations

(click to see Properties)

GET| GET (ALL)

Some options that you can use for each operations:

- **Getting warnings in response:**NITRO allows you to get warnings in an operation by specifying the "warning" query parameter as "yes". For example, to get warnings while connecting to the Citrix ADC appliance, the URL is as follows:

  http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/login?warning=yes

  If any, the warnings are displayed in the response payload with the HTTP code "209 X-NITRO-WARNING".

- **Authenticated access for individual NITRO operations:**NITRO allows you to logon to the Citrix ADC appliance to perform individual operations. You can use this option instead of creating a NITRO session (using the login object) and then using that session to perform all operations,

  To do this, you must specify the username and password in the request header of the NITRO request as follows:

  X-NITRO-USER:<username>

  X-NITRO-PASS:<password>

  Note: In such cases, make sure that the request header DOES not include the following:

  Cookie:NITRO_AUTH_TOKEN=<tokenvalue>

`*Note:`*

Mandatory parameters are marked in redand placeholder content is marked in <green>.

get

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/policydataset_binding/name_value<String>

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "policydataset_binding": [ {
```

```
"name":<String_value>,

"policydataset_value_binding":<policydataset_value_binding[]_value>

}]}
```

get (all)

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/policydataset_binding

Query-parameters:

bulkbindings

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/policydataset_binding?**bulkbindings=yes**

NITRO allows you to fetch bindings in bulk.

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "policydataset_binding": [ {

"name":<String_value>,

"policydataset_value_binding":<policydataset_value_binding[]_value>

}]}
```

policydataset_value_binding

Binding object showing the value that can be bound to policydataset.

Properties

(click to see Operations)

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
value	<String>	Read-write	Value of the specified type that is associated with the dataset.
name	<String>	Read-write	Name of the dataset to which to bind the value. Minimum length = 1
index	<Double>	Read-write	The index of the value (ipv4, ipv6, number) associated with the set.
__count	<Double>	Read-write	count parameter

Operations

(click to see Properties)

ADD:| DELETE:| GET| GET (ALL)| COUNT

Some options that you can use for each operations:

- **Getting warnings in response:**NITRO allows you to get warnings in an operation by specifying the "warning" query parameter as "yes". For example, to get warnings while connecting to the Citrix ADC appliance, the URL is as follows:

  http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/login?warning=yes

If any, the warnings are displayed in the response payload with the HTTP code "209 X-NITRO-WARNING".

- **Authenticated access for individual NITRO operations:**NITRO allows you to logon to the Citrix ADC appliance to perform individual operations. You can use this option instead of creating a NITRO session (using the login object) and then using that session to perform all operations,

  To do this, you must specify the username and password in the request header of the NITRO request as follows:

  X-NITRO-USER:<username>

  X-NITRO-PASS:<password>

  Note: In such cases, make sure that the request header DOES not include the following:

  Cookie:NITRO_AUTH_TOKEN=<tokenvalue>

`*Note:`*

Mandatory parameters are marked in redand placeholder content is marked in <green>.

add

URL:http://<Citrix-ADC-IP-address(NSIP)/nitro/v1/config/policydataset_value_binding

HTTP Method:PUT

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Content-Type:application/json

Request Payload:

```
{

"policydataset_value_binding":{

<b>"name":<String_value>,

</b><b>"value":<String_value>,
```

```
</b>"index":<Double_value>

}}
```

Response:

HTTP Status Code on Success: 201 Created HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error

delete

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/policydataset_value_binding/name_value<String>

Query-parameters:

args

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/policydataset_value_binding/name_value<String>?**args=value:<String_value>**

HTTP Method:DELETE

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue>

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error

get

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/policydataset_value_binding/name_value<String>

Query-parameters:

filter

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/policydataset_value_binding/name_value<String>?**filter=property-name1:property-value1,property-name2:property-value2**

Use this query-parameter to get the filtered set of policydataset_value_binding resources configured on Citrix ADC. Filtering can be done on any of the properties of the resource.

pagination

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/policydataset_value_binding/name_value<String>?**pagesize=#no;pageno=#no**

Use this query-parameter to get the policydataset_value_binding resources in chunks.

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "policydataset_value_binding": [ {

"value":<String_value>,

"name":<String_value>,

"index":<Double_value>

}]}
```

get (all)

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/policydataset_value_binding

Query-parameters:

bulkbindings

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/policydataset_value_binding?**bulkbindings=yes**

NITRO allows you to fetch bindings in bulk.

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "policydataset_value_binding": [ {

"value":<String_value>,

"name":<String_value>,

"index":<Double_value>

}]}
```

count

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/policydataset_value_binding/name_value<String>?**count=yes**

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{"policydataset_value_binding": [ { "__count": "#no"} ] }
```

policyevaluation

Configuration for expr evaluation resource.

Properties

(click to see Operations)

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
expression	<String>	Read-write	Expression string. For example: http.req.body(100).contains("this").
action	<String>	Read-write	Rewrite action name. Supported rewrite action types are:

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
			-delete -delete_all -delete_http_header -insert_after -insert_after_all -insert_before -insert_before_all -insert_http_header -replace -replace_all . Minimum length = 1
type	<String>	Read-write	Indicates request or response input packet. Possible values = HTTP_REQ, HTTP_RES, TEXT
input	<String>	Read-write	Text representation of input packet.
pitmodifiedinputdata	<String>	Read-only	Text representation of packet after evaluating expression or rewrite action.
pitboolresult	<Boolean>	Read-only	Result of the expression in bool format.

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
pitnumresult	<Double>	Read-only	Result of the expression in num format.
pitdoubleresult	<Double>	Read-only	Result of the expression in double format.
pitulongresult	<Double>	Read-only	Result of the expression in unsigned long format.
pitrefresult	<String>	Read-only	Result of the expression in string format.
pitoffsetresult	<Double>	Read-only	Offset of the resultant sting.
pitoffsetresultlen	<Double>	Read-only	Offset length of the resultant sting.
istruncatedrefresult	<Boolean>	Read-only	Identify whether ref result is truncated result.
pitboolevaltime	<Double>	Read-only	Average evaluation time of bool type expression in nanoseconds.
pitnumevaltime	<Double>	Read-only	Average evaluation time of num type expression in nanoseconds.
pitdoubleevaltime	<Double>	Read-only	Average evaluation time of double type expression in nanoseconds.

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
pitulongevaltime	<Double>	Read-only	Average evaluation time of unsigned long type expression in nanoseconds.
pitrefevaltime	<Double>	Read-only	Average evaluation time of string type expression in nanoseconds.
pitoffsetevaltime	<Double>	Read-only	Average evaluation time in finding offset of the resultant string in the input. Time is in nanoseconds.
pitactionevaltime	<Double>	Read-only	Average evaluation time of rewrite action in nanoseconds.
pitoperationperformerarray	<String[]>	Read-only	Details of the operation Citrix ADC performed at various offsets during applying of rewrite action on input data. Operation can be insertion, modfication or deletion. Possible values = INSERT, MODIFY, DELETE
pitoldoffsetarray	<Double[]>	Read-only	Details of the offsets in the input data at which Citrix ADC either inserted or modified or deleted data during applying of rewrite action.

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
pitnewoffsetarray	<Double[]>	Read-only	Details of the offsets in the output data at which Citrix ADC either inserted or modified or deleted data during applying of rewrite action.
pitoffsetlengtharray	<Integer[]>	Read-only	Details of the lengths of the data which Citrix ADC either inserted or modified or deleted during applying of rewrite action.
pitboolerrorresult	<String>	Read-only	Result of the bool type expression if any error occurs during evaluation. Result will be in string format.
pitnumerrorresult	<String>	Read-only	Result of the num type expression if any error occurs during evaluation. Result will be in string format.
pitdoubleerrorresult	<String>	Read-only	Result of the double type expression if any error occurs during evaluation. Result will be in string format.
pitulongerrorresult	<String>	Read-only	Result of the unsigned long type expression if any error occurs during

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
			evaluation. Result will be in string format.
pitreferrorresult	<String>	Read-only	Result of the ref type expression if any error occurs during evaluation. Result will be in string format.
pitoffseterrorresult	<String>	Read-only	Result of the expression if any error occurs in calculating offset. Result will be in string format.
pitactionerrorresult	<String>	Read-only	Result of the action if any error occurs in evaluation. Result will be in string format.
__count	<Double>	Read-only	count parameter

Operations

(click to see Properties)

GET (ALL)| COUNT

Some options that you can use for each operations:

- **Getting warnings in response:**NITRO allows you to get warnings in an operation by specifying the "warning" query parameter as "yes". For example, to get warnings while connecting to the Citrix ADC appliance, the URL is as follows:

  http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/login?warning=yes

  If any, the warnings are displayed in the response payload with the HTTP code "209 X-NITRO-WARNING".

- **Authenticated access for individual NITRO operations:**NITRO allows you to logon to the Citrix ADC appliance to perform individual operations. You can use this option instead of creating a NITRO session (using the login object) and then using that session to perform all operations,

  To do this, you must specify the username and password in the request header of the NITRO request as follows:

  X-NITRO-USER:<username>

  X-NITRO-PASS:<password>

  Note: In such cases, make sure that the request header DOES not include the following:

  Cookie:NITRO_AUTH_TOKEN=<tokenvalue>

`*Note:`*

Mandatory parameters are marked in redand placeholder content is marked in <green>.

get (all)

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/policyevaluation

Query-parameters:

args

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/policyevaluation?**args=expression:<String_value>,action:<String_value>,type:<String_value>,input:<String_value>,**

Use this query-parameter to get policyevaluation resources based on additional properties.

attrs

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/policyevaluation?**attrs=property-name1,property-name2**

Use this query parameter to specify the resource details that you want to retrieve.

filter

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/policyevaluation?**filter=property-name1:property-val1,property-name2:property-val2**

Use this query-parameter to get the filtered set of policyevaluation resources configured on Citrix ADC. Filtering can be done on any of the properties of the resource.

view

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/policyevaluation?**view=summary**

Note: By default, the retrieved results are displayed in detail view (?view=detail).

pagination

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/policyevaluation?**pagesize=#no;pageno=#no**

Use this query-parameter to get the policyevaluation resources in chunks.

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "policyevaluation": [ {

expression:<String_value>,action:<String_value>,<b>type:<String_value>,</
b><b>input:<String_value>,</b>"pitmodifiedinputdata":<String_value>,

"pitboolresult":<Boolean_value>,

"pitnumresult":<Double_value>,

"pitdoubleresult":<Double_value>,
```

```
"pitulongresult":<Double_value>,

"pitrefresult":<String_value>,

"pitoffsetresult":<Double_value>,

"pitoffsetresultlen":<Double_value>,

"istruncatedrefresult":<Boolean_value>,

"pitboolevaltime":<Double_value>,

"pitnumevaltime":<Double_value>,

"pitdoubleevaltime":<Double_value>,

"pitulongevaltime":<Double_value>,

"pitrefevaltime":<Double_value>,

"pitoffsetevaltime":<Double_value>,

"pitactionevaltime":<Double_value>,

"pitoperationperformerarray":<String[]_value>,

"pitoldoffsetarray":<Double[]_value>,

"pitnewoffsetarray":<Double[]_value>,

"pitoffsetlengtharray":<Integer[]_value>,

"pitboolerrorresult":<String_value>,

"pitnumerrorresult":<String_value>,

"pitdoubleerrorresult":<String_value>,

"pitulongerrorresult":<String_value>,

"pitreferrorresult":<String_value>,

"pitoffseterrorresult":<String_value>,
```

```
"pitactionerrorresult":<String_value>

}]}
```

count

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/policyevaluation?**args=expression:<String_value>,action:<String_value>,type:<String_value>,input:<String_value>,;cour**

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "policyevaluation": [ { "__count": "#no"} ] }
```

policyexpression

Configuration for expression resource.

Properties

(click to see Operations)

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
name	<String>	Read-write	Unique name for the expression. Not case

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
			sensitive. Must begin with an ASCII letter or underscore (_) character, and must consist only of ASCII alphanumeric or underscore characters. Must not begin with 're' or 'xp' or be a word reserved for use as an expression qualifier prefix (such as HTTP) or enumeration value (such as ASCII). Must not be the name of an existing named expression, pattern set, dataset, stringmap, or HTTP callout. Minimum length = 1
value	<String>	Read-write	Expression string. For example: http.req.body(100).contains("this").
description	<String>	Read-write	Description for the expression.
comment	<String>	Read-write	Any comments associated with the expression. Displayed upon viewing the policy expression.
clientsecuritymessage	<String>	Read-write	Message to display if the expression fails.

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
			Allowed for classic end-point check expressions only. Minimum length = 1
type	<String>	Read-write	Type of expression. Can be a classic or default syntax (advanced) expression. Possible values = CLASSIC, ADVANCED
hits	<Double>	Read-only	The total number of hits.
pihits	<Double>	Read-only	The total number of hits.
type1	<String>	Read-only	The type of expression. This is for output only. Possible values = CLASSIC, ADVANCED
isdefault	<Boolean>	Read-only	A value of true is returned if it is a default policy expression.
__count	<Double>	Read-only	count parameter

Operations

(click to see Properties)

ADD| DELETE| UPDATE| UNSET| GET (ALL)| GET| COUNT

Some options that you can use for each operations:

- **Getting warnings in response:**NITRO allows you to get warnings in an operation by specifying the "warning" query parameter as "yes". For example, to get warnings while connecting to the Citrix ADC appliance, the URL is as follows:

  http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/login?warning=yes

  If any, the warnings are displayed in the response payload with the HTTP code "209 X-NITRO-WARNING".

- **Authenticated access for individual NITRO operations:**NITRO allows you to logon to the Citrix ADC appliance to perform individual operations. You can use this option instead of creating a NITRO session (using the login object) and then using that session to perform all operations,

  To do this, you must specify the username and password in the request header of the NITRO request as follows:

  X-NITRO-USER:<username>

  X-NITRO-PASS:<password>

  Note: In such cases, make sure that the request header DOES not include the following:

  Cookie:NITRO_AUTH_TOKEN=<tokenvalue>

`*Note:`*

Mandatory parameters are marked in redand placeholder content is marked in <green>.

add

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/policyexpression

HTTP Method:POST

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Content-Type:application/json

Request Payload:

```
{"policyexpression":{
```

```
<b>"name":<String_value>,

</b><b>"value":<String_value>,

</b>"description":<String_value>,

"comment":<String_value>,

"clientsecuritymessage":<String_value>

}}
```

Response:

HTTP Status Code on Success: 201 Created HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error

delete

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/policyexpression/name_value<String>

HTTP Method:DELETE

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue>

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error

update

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/policyexpression

HTTP Method:PUT

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Content-Type:application/json

Request Payload:

```
{"policyexpression":{

<b>"name":<String_value>,

</b>"value":<String_value>,

"description":<String_value>,

"comment":<String_value>,

"clientsecuritymessage":<String_value>

}}
```

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error

unset

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/policyexpression?**action=unset**

HTTP Method:POST

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Content-Type:application/json

Request Payload:

```
{"policyexpression":{

<b>"name":<String_value>,

</b>"description":true,
```

```
"comment":true,

"clientsecuritymessage":true

}}
```

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error

get (all)

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/policyexpression

Query-parameters:

args

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/policyexpression?**args=name:<String_value>,type:<String_value>**

Use this query-parameter to get policyexpression resources based on additional properties.

attrs

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/policyexpression?**attrs=property-name1,property-name2**

Use this query parameter to specify the resource details that you want to retrieve.

filter

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/policyexpression?**filter=property-name1:property-val1,property-name2:property-val2**

Use this query-parameter to get the filtered set of policyexpression resources configured on Citrix ADC. Filtering can be done on any of the properties of the resource.

view

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/policyexpression?**view=summary**

Note: By default, the retrieved results are displayed in detail view (?view=detail).

pagination

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/policyexpression?**pagesize=#no;pageno=#no**

Use this query-parameter to get the policyexpression resources in chunks.

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "policyexpression": [ {

name:<String_value>,type:<String_value>"value":<String_value>,

"hits":<Double_value>,

"pihits":<Double_value>,

"type1":<String_value>,

"clientsecuritymessage":<String_value>,

"description":<String_value>,

"comment":<String_value>,

"isdefault":<Boolean_value>
```

```
}]}
```

get

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/policyexpression/name_value<String>

Query-parameters:

attrs

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/policyexpression/name_value<String>?**attrs=property-name1,property-name2**

Use this query parameter to specify the resource details that you want to retrieve.

view

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/policyexpression/name_value<String>?**view=summary**

Note: By default, the retrieved results are displayed in detail view (?view=detail).

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "policyexpression": [ {

name:<String_value>,type:<String_value>"value":<String_value>,
```

```
"hits":<Double_value>,

"pihits":<Double_value>,

"type1":<String_value>,

"clientsecuritymessage":<String_value>,

"description":<String_value>,

"comment":<String_value>,

"isdefault":<Boolean_value>

}]}
```

count

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/policyexpression?**count=yes**

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "policyexpression": [ { "__count": "#no"} ] }
```

policyhttpcallout

Configuration for HTTP callout resource.

Properties

(click to see Operations)

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
name	<String>	Read-write	Name for the HTTP callout. Not case sensitive. Must begin with an ASCII letter or underscore (_) character, and must consist only of ASCII alphanumeric or underscore characters. Must not begin with 're' or 'xp' or be a word reserved for use as an expression qualifier prefix (such as HTTP) or enumeration value (such as ASCII). Must not be the name of an existing named expression, pattern set, dataset, stringmap, or HTTP callout. Minimum length = 1
ipaddress	<String>	Read-write	IP Address of the server (callout agent) to which the callout is sent. Can be an IPv4 or IPv6 address.

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
			Mutually exclusive with the Virtual Server parameter. Therefore, you cannot set the <IP Address, Port> and the Virtual Server in the same HTTP callout.
port	<Integer>	Read-write	Server port to which the HTTP callout agent is mapped. Mutually exclusive with the Virtual Server parameter. Therefore, you cannot set the <IP Address, Port> and the Virtual Server in the same HTTP callout. Minimum value = 1
vserver	<String>	Read-write	Name of the load balancing, content switching, or cache redirection virtual server (the callout agent) to which the HTTP callout is sent. The service type of the virtual server must be HTTP. Mutually exclusive with the IP address and port parameters. Therefore, you cannot set the <IP Address, Port> and

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
			the Virtual Server in the same HTTP callout. Minimum length = 1
returntype	<String>	Read-write	Type of data that the target callout agent returns in response to the callout. Available settings function as follows: * TEXT - Treat the returned value as a text string. * NUM - Treat the returned value as a number. * BOOL - Treat the returned value as a Boolean value. Note: You cannot change the return type after it is set. Possible values = BOOL, NUM, TEXT
httpmethod	<String>	Read-write	Method used in the HTTP request that this callout sends. Mutually exclusive with the full HTTP request expression. Possible values = GET, POST

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
hostexpr	<String>	Read-write	String expression to configure the Host header. Can contain a literal value (for example, 10.101.10.11) or a derived value (for example, http.req.header("Host")). The literal value can be an IP address or a fully qualified domain name. Mutually exclusive with the full HTTP request expression. Minimum length = 1
urlstemexpr	<String>	Read-write	String expression for generating the URL stem. Can contain a literal string (for example, "/mysite/index.html") or an expression that derives the value (for example, http.req.url). Mutually exclusive with the full HTTP request expression. Minimum length = 1
headers	<String[]>	Read-write	One or more headers to insert into the HTTP request. Each header is specified as "name(expr)", where expr is an expression that is evaluated at

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
			runtime to provide the value for the named header. You can configure a maximum of eight headers for an HTTP callout. Mutually exclusive with the full HTTP request expression.
parameters	<String[]>	Read-write	One or more query parameters to insert into the HTTP request URL (for a GET request) or into the request body (for a POST request). Each parameter is specified as "name(expr)", where expr is an expression that is evaluated at run time to provide the value for the named parameter (name=value). The parameter values are URL encoded. Mutually exclusive with the full HTTP request expression.
bodyexpr	<String>	Read-write	An advanced string expression for generating the body of the request. The expression can contain a literal string or an expression that derives the value (for

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
			example, client.ip.src). Mutually exclusive with -fullReqExpr. Minimum length = 1
fullreqexpr	<String>	Read-write	Exact HTTP request, in the form of an expression, which the Citrix ADC sends to the callout agent. If you set this parameter, you must not include HTTP method, host expression, URL stem expression, headers, or parameters. The request expression is constrained by the feature for which the callout is used. For example, an HTTP.RES expression cannot be used in a request-time policy bank or in a TCP content switching policy bank. The Citrix ADC does not check the validity of this request. You must

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
			manually validate the request. Minimum length = 1
scheme	<String>	Read-write	Type of scheme for the callout server. Possible values = http, https
resultexpr	<String>	Read-write	Expression that extracts the callout results from the response sent by the HTTP callout agent. Must be a response based expression, that is, it must begin with HTTP.RES. The operations in this expression must match the return type. For example, if you configure a return type of TEXT, the result expression must be a text based expression. If the return type is NUM, the result expression (resultExpr) must return a numeric value, as in the following example: http.res.body(10000).length. Minimum length = 1
cacheforsecs	<Double>	Read-write	Duration, in seconds, for which the callout response is cached.

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
			The cached responses are stored in an integrated caching content group named "calloutContentGroup". If no duration is configured, the callout responses will not be cached unless normal caching configuration is used to cache them. This parameter takes precedence over any normal caching configuration that would otherwise apply to these responses. Note that the calloutContentGroup definition may not be modified or removed nor may it be used with other cache policies. Minimum value = 1 Maximum value = 31536000
comment	<String>	Read-write	Any comments to preserve information about this HTTP callout.
hits	<Double>	Read-only	Total hits.

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
undefhits	<Double>	Read-only	Total undefs.
svrstate	<String>	Read-only	The state of the service. Possible values = UP, DOWN, UNKNOWN, BUSY, OUT OF SERVICE, GOING OUT OF SERVICE, DOWN WHEN GOING OUT OF SERVICE, NS_EMPTY_STR, Unknown, DISABLED
effectivestate	<String>	Read-only	The effective state of the service. Possible values = UP, DOWN, UNKNOWN, BUSY, OUT OF SERVICE, GOING OUT OF SERVICE, DOWN WHEN GOING OUT OF SERVICE, NS_EMPTY_STR, Unknown, DISABLED
undefreason	<String>	Read-only	Reason for last undef. Possible values = Failed to add service, Vserver not found, Not a HTTP or SSL vserver,

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
			Generated callout request is invalid, Content-Length header not found in callout request, Not enough space to put Content-Length value, Config incomplete, Server is DOWN, Creating callout connection failed, No memory to generate callout request packets, No memory to create callout task, No memory to create callout async, Callout request expression undef, No callout response expression, Skipped callout response eval, Callout response pixl init undef, Callout response expression undef
recursivecallout	<Double>	Read-only	Number of recursive callouts.
__count	<Double>	Read-only	count parameter

Operations

(click to see Properties)

ADD| DELETE| UPDATE| UNSET| GET (ALL)| GET| COUNT

Some options that you can use for each operations:

- **Getting warnings in response:**NITRO allows you to get warnings in an operation by specifying the "warning" query parameter as "yes". For example, to get warnings while connecting to the Citrix ADC appliance, the URL is as follows:

  http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/login?warning=yes

  If any, the warnings are displayed in the response payload with the HTTP code "209 X-NITRO-WARNING".

- **Authenticated access for individual NITRO operations:**NITRO allows you to logon to the Citrix ADC appliance to perform individual operations. You can use this option instead of creating a NITRO session (using the login object) and then using that session to perform all operations,

  To do this, you must specify the username and password in the request header of the NITRO request as follows:

  X-NITRO-USER:<username>

  X-NITRO-PASS:<password>

  Note: In such cases, make sure that the request header DOES not include the following:

  Cookie:NITRO_AUTH_TOKEN=<tokenvalue>

`*Note:`*

Mandatory parameters are marked in redand placeholder content is marked in <green>.

add

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/policyhttpcallout

HTTP Method:POST

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Content-Type:application/json

Request Payload:

```
{"policyhttpcallout":{
<b>"name":<String_value>,
</b>"ipaddress":<String_value>,
"port":<Integer_value>,
"vserver":<String_value>,
"returntype":<String_value>,
"httpmethod":<String_value>,
"hostexpr":<String_value>,
"urlstemexpr":<String_value>,
"headers":<String[]_value>,
"parameters":<String[]_value>,
"bodyexpr":<String_value>,
"fullreqexpr":<String_value>,
"scheme":<String_value>,
"resultexpr":<String_value>,
"cacheforsecs":<Double_value>,
"comment":<String_value>
}}
```

Response:

HTTP Status Code on Success: 201 Created HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error

delete

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/policyhttpcallout/name_value<String>

HTTP Method:DELETE

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue>

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error

update

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/policyhttpcallout

HTTP Method:PUT

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Content-Type:application/json

Request Payload:

```
{"policyhttpcallout":{

<b>"name":<String_value>,

</b>"ipaddress":<String_value>,

"port":<Integer_value>,

"vserver":<String_value>,

"returntype":<String_value>,
```

```
"httpmethod":<String_value>,

"hostexpr":<String_value>,

"urlstemexpr":<String_value>,

"headers":<String[]_value>,

"parameters":<String[]_value>,

"bodyexpr":<String_value>,

"fullreqexpr":<String_value>,

"scheme":<String_value>,

"resultexpr":<String_value>,

"cacheforsecs":<Double_value>,

"comment":<String_value>

}}
```

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error

unset

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/policyhttpcallout?**action=unset**

HTTP Method:POST

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Content-Type:application/json

Request Payload:

```
{"policyhttpcallout":{

<b>"name":<String_value>,

</b>"ipaddress":true,

"port":true,

"vserver":true,

"httpmethod":true,

"hostexpr":true,

"urlstemexpr":true,

"headers":true,

"parameters":true,

"bodyexpr":true,

"fullreqexpr":true,

"resultexpr":true,

"cacheforsecs":true,

"comment":true

}}
```

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error

get (all)

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/policyhttpcallout

Query-parameters:

attrs

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/policyhttpcallout?**attrs=property-name1,property-name2**

Use this query parameter to specify the resource details that you want to retrieve.

filter

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/policyhttpcallout?**filter=property-name1:property-val1,property-name2:property-val2**

Use this query-parameter to get the filtered set of policyhttpcallout resources configured on Citrix ADC. Filtering can be done on any of the properties of the resource.

view

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/policyhttpcallout?**view=summary**

Note: By default, the retrieved results are displayed in detail view (?view=detail).

pagination

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/policyhttpcallout?**pagesize=#no;pageno=#no**

Use this query-parameter to get the policyhttpcallout resources in chunks.

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "policyhttpcallout": [ {

"name":<String_value>,

"ipaddress":<String_value>,

"port":<Integer_value>,

"vserver":<String_value>,

"returntype":<String_value>,

"scheme":<String_value>,

"httpmethod":<String_value>,

"hostexpr":<String_value>,

"urlstemexpr":<String_value>,

"headers":<String[]_value>,

"parameters":<String[]_value>,

"fullreqexpr":<String_value>,

"resultexpr":<String_value>,

"hits":<Double_value>,

"undefhits":<Double_value>,

"svrstate":<String_value>,

"effectivestate":<String_value>,

"undefreason":<String_value>,
```

```
"recursivecallout":<Double_value>,

"bodyexpr":<String_value>,

"cacheforsecs":<Double_value>,

"comment":<String_value>

}]}
```

get

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/policyhttpcallout/name_value<String>

Query-parameters:

attrs

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/policyhttpcallout/name_value<String>?**attrs=property-name1,property-name2**

Use this query parameter to specify the resource details that you want to retrieve.

view

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/policyhttpcallout/name_value<String>?**view=summary**

Note: By default, the retrieved results are displayed in detail view (?view=detail).

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "policyhttpcallout": [ {

"name":<String_value>,

"ipaddress":<String_value>,

"port":<Integer_value>,

"vserver":<String_value>,

"returntype":<String_value>,

"scheme":<String_value>,

"httpmethod":<String_value>,

"hostexpr":<String_value>,

"urlstemexpr":<String_value>,

"headers":<String[]_value>,

"parameters":<String[]_value>,

"fullreqexpr":<String_value>,

"resultexpr":<String_value>,

"hits":<Double_value>,

"undefhits":<Double_value>,

"svrstate":<String_value>,

"effectivestate":<String_value>,

"undefreason":<String_value>,

"recursivecallout":<Double_value>,
```

```
"bodyexpr":<String_value>,

"cacheforsecs":<Double_value>,

"comment":<String_value>

}]}
```

count

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/policyhttpcallout?**count=yes**

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "policyhttpcallout": [ { "__count": "#no"} ] }
```

policymap

Configuration for map policy resource.

Properties

(click to see Operations)

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
mappolicyname	<String>	Read-write	Name for the map policy. Must begin with a letter, number, or the underscore (_) character and must consist only of letters, numbers, and the hash (#), period (.), colon (:), space ( ), at (@), equals (=), hyphen (-), and underscore (_) characters. CLI Users: If the name includes one or more spaces, enclose the name in double or single quotation marks (for example, "my map" or 'my map'). Minimum length = 1
sd	<String>	Read-write	Publicly known source domain name. This is the domain name with which a client request arrives at a reverse proxy virtual server for cache redirection. If you specify a source domain, you must specify a target domain. Minimum length = 1

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
su	<String>	Read-write	Source URL. Specify all or part of the source URL, in the following format: /[[prefix] [*]] [.suffix]. Minimum length = 1
td	<String>	Read-write	Target domain name sent to the server. The source domain name is replaced with this domain name. Minimum length = 1
tu	<String>	Read-write	Target URL. Specify the target URL in the following format: /[[prefix] [*]][.suffix]. Minimum length = 1
targetname	<String>	Read-only	The expression string.
__count	<Double>	Read-only	count parameter

Operations

(click to see Properties)

ADD| DELETE| GET (ALL)| GET| COUNT

Some options that you can use for each operations:

- **Getting warnings in response:**NITRO allows you to get warnings in an operation by specifying the "warning" query parameter as "yes". For example, to get warnings while connecting to the Citrix ADC appliance, the URL is as follows:

  http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/login?warning=yes

If any, the warnings are displayed in the response payload with the HTTP code "209 X-NITRO-WARNING".

- **Authenticated access for individual NITRO operations:**NITRO allows you to logon to the Citrix ADC appliance to perform individual operations. You can use this option instead of creating a NITRO session (using the login object) and then using that session to perform all operations,

  To do this, you must specify the username and password in the request header of the NITRO request as follows:

  X-NITRO-USER:<username>

  X-NITRO-PASS:<password>

  Note: In such cases, make sure that the request header DOES not include the following:

  Cookie:NITRO_AUTH_TOKEN=<tokenvalue>

`*Note:`*

Mandatory parameters are marked in redand placeholder content is marked in <green>.

add

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/policymap

HTTP Method:POST

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Content-Type:application/json

Request Payload:

```
{"policymap":{

<b>"mappolicyname":<String_value>,

</b><b>"sd":<String_value>,

</b>"su":<String_value>,
```

```
"td":<String_value>,

"tu":<String_value>

}}
```

Response:

HTTP Status Code on Success: 201 Created HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error

delete

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/policymap/mappolicyname_value<String>

HTTP Method:DELETE

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue>

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error

get (all)

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/policymap

Query-parameters:

attrs

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/policymap?**attrs=property-name1,property-name2**

Use this query parameter to specify the resource details that you want to retrieve.

filter

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/policymap?**filter=property-name1:property-val1,property-name2:property-val2**

Use this query-parameter to get the filtered set of policymap resources configured on Citrix ADC. Filtering can be done on any of the properties of the resource.

view

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/policymap?**view=summary**

Note: By default, the retrieved results are displayed in detail view (?view=detail).

pagination

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/policymap?**pagesize=#no;pageno=#no**

Use this query-parameter to get the policymap resources in chunks.

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "policymap": [ {

"mappolicyname":<String_value>,

"sd":<String_value>,

"su":<String_value>,

"td":<String_value>,
```

```
"tu":<String_value>,

"targetname":<String_value>

}]}
```

get

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/policymap/mappolicyname_value<String>

Query-parameters:

attrs

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/policymap/mappolicyname_value<String>?**attrs=property-name1,property-name2**

Use this query parameter to specify the resource details that you want to retrieve.

view

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/policymap/mappolicyname_value<String>?**view=summary**

Note: By default, the retrieved results are displayed in detail view (?view=detail).

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "policymap": [ {

"mappolicyname":<String_value>,

"sd":<String_value>,

"su":<String_value>,

"td":<String_value>,

"tu":<String_value>,

"targetname":<String_value>

}]}
```

count

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/policymap?**count=yes**

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "policymap": [ { "__count": "#no"} ] }
```

policyparam

Configuration for policy parameter resource.

Properties

(click to see Operations)

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
timeout	<Double>	Read-write	Maximum time in milliseconds to allow for processing expressions without interruption. If the timeout is reached then the evaluation causes an UNDEF to be raised and no further processing is performed. Minimum value = 1 Maximum value = 5000

Operations

(click to see Properties)

UPDATE| UNSET| GET (ALL)

Some options that you can use for each operations:

- **Getting warnings in response:**NITRO allows you to get warnings in an operation by specifying the "warning" query parameter as "yes". For example, to get warnings while connecting to the Citrix ADC appliance, the URL is as follows:

  http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/login?warning=yes

  If any, the warnings are displayed in the response payload with the HTTP code "209 X-NITRO-WARNING".

- **Authenticated access for individual NITRO operations:**NITRO allows you to logon to the Citrix ADC appliance to perform individual operations. You can use this option instead of creating a NITRO session (using the login object) and then using that session to perform all operations,

  To do this, you must specify the username and password in the request header of the NITRO request as follows:

  X-NITRO-USER:<username>

  X-NITRO-PASS:<password>

  Note: In such cases, make sure that the request header DOES not include the following:

  Cookie:NITRO_AUTH_TOKEN=<tokenvalue>

`*Note:`*

Mandatory parameters are marked in redand placeholder content is marked in <green>.

update

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/policyparam

HTTP Method:PUT

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Content-Type:application/json

Request Payload:

```
{"policyparam":{

"timeout":<Double_value>

}}
```

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error

unset

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/policyparam?**action=unset**

HTTP Method:POST

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Content-Type:application/json

Request Payload:

```
{"policyparam":{

"timeout":true

}}
```

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error

get (all)

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/policyparam

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "policyparam": [ {

"timeout":<Double_value>

}]}
```

policypatset

Configuration for PAT set resource.

Properties

(click to see Operations)

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
name	<String>	Read-write	Unique name of the pattern set. Not case sensitive. Must begin with an ASCII letter or underscore (_) character and must contain only alphanumeric and underscore characters. Must not be the name of an existing named expression, pattern set, dataset, string

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
			map, or HTTP callout. Minimum length = 1
indextype	<String>	Read-write	Index type. Possible values = Auto-generated, User-defined
comment	<String>	Read-write	Any comments to preserve information about this patset.
description	<String>	Read-only	Description of the patset.
__count	<Double>	Read-only	count parameter

Operations

(click to see Properties)

ADD| DELETE| GET (ALL)| GET| COUNT

Some options that you can use for each operations:

- **Getting warnings in response:**NITRO allows you to get warnings in an operation by specifying the "warning" query parameter as "yes". For example, to get warnings while connecting to the Citrix ADC appliance, the URL is as follows:

  http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/login?warning=yes

  If any, the warnings are displayed in the response payload with the HTTP code "209 X-NITRO-WARNING".

- **Authenticated access for individual NITRO operations:**NITRO allows you to logon to the Citrix ADC appliance to perform individual operations. You can use this option instead of creating a NITRO session (using the login object) and then using that session to perform all operations,

  To do this, you must specify the username and password in the request header of the NITRO request as follows:

X-NITRO-USER:<username>

X-NITRO-PASS:<password>

Note: In such cases, make sure that the request header DOES not include the following:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue>

`*Note:`*

Mandatory parameters are marked in redand placeholder content is marked in <green>.

add

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/policypatset

HTTP Method:POST

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Content-Type:application/json

Request Payload:

```
{"policypatset":{

<b>"name":<String_value>,

</b>"indextype":<String_value>,

"comment":<String_value>

}}
```

Response:

HTTP Status Code on Success: 201 Created HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error

delete

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/policypatset/name_value<String>

HTTP Method:DELETE

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue>

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error

get (all)

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/policypatset

Query-parameters:

attrs

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/policypatset?**attrs=property-name1,property-name2**

Use this query parameter to specify the resource details that you want to retrieve.

filter

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/policypatset?**filter=property-name1:property-val1,property-name2:property-val2**

Use this query-parameter to get the filtered set of policypatset resources configured on Citrix ADC. Filtering can be done on any of the properties of the resource.

view

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/policypatset?**view=summary**

Note: By default, the retrieved results are displayed in detail view (?view=detail).

pagination

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/policypatset?**pagesize=#no;pageno=#no**

Use this query-parameter to get the policypatset resources in chunks.

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "policypatset": [ {

"name":<String_value>,

"description":<String_value>,

"indextype":<String_value>,

"comment":<String_value>

}]}
```

get

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/policypatset/name_value<String>

Query-parameters:

attrs

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/policypatset/name_value<String>?**attrs=property-name1,property-name2**

Use this query parameter to specify the resource details that you want to retrieve.

view

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/policypatset/name_value<String>?**view=summary**

Note: By default, the retrieved results are displayed in detail view (?view=detail).

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "policypatset": [ {

"name":<String_value>,

"description":<String_value>,

"indextype":<String_value>,

"comment":<String_value>

}]}
```

count

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/policypatset?**count=yes**

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "policypatset": [ { "__count": "#no"} ] }
```

policypatset_binding

Binding object which returns the resources bound to policypatset.

Properties

(click to see Operations)

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
name	<String>	Read-write	Name of the pattern set for which to display the detailed information. If a name is not provided, a list of all pattern sets configured on the appliance is shown. Minimum length = 1
policypatset_pattern_binding	<policypatset_pattern_binding[]>	Read-only	pattern that can be bound to policypatset.

Operations

(click to see Properties)

GET| GET (ALL)

Some options that you can use for each operations:

- **Getting warnings in response:**NITRO allows you to get warnings in an operation by specifying the "warning" query parameter as "yes". For example, to get warnings while connecting to the Citrix ADC appliance, the URL is as follows:

  http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/login?warning=yes

  If any, the warnings are displayed in the response payload with the HTTP code "209 X-NITRO-WARNING".

- **Authenticated access for individual NITRO operations:**NITRO allows you to logon to the Citrix ADC appliance to perform individual operations. You can use this option instead of creating a NITRO session (using the login object) and then using that session to perform all operations,

  To do this, you must specify the username and password in the request header of the NITRO request as follows:

  X-NITRO-USER:<username>

  X-NITRO-PASS:<password>

  Note: In such cases, make sure that the request header DOES not include the following:

  Cookie:NITRO_AUTH_TOKEN=<tokenvalue>

`*Note:`*

Mandatory parameters are marked in redand placeholder content is marked in <green>.

get

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/policypatset_binding/name_value<String>

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "policypatset_binding": [ {

"name":<String_value>,

"policypatset_pattern_binding":<policypatset_pattern_binding[]_value>

}]}
```

get (all)

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/policypatset_binding

Query-parameters:

bulkbindings

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/policypatset_binding?**bulkbindings=yes**

NITRO allows you to fetch bindings in bulk.

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "policypatset_binding": [ {

"name":<String_value>,

"policypatset_pattern_binding":<policypatset_pattern_binding[]_value>

}]}
```

policypatset_pattern_binding

Binding object showing the pattern that can be bound to policypatset.

Properties

(click to see Operations)

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
String	<String>	Read-write	String of characters that constitutes a pattern. For more information about the characters that can be used, refer to the character set parameter. Note: Minimum length for pattern sets used in rewrite actions of type REPLACE_ALL, DELETE_ALL, INSERT_AFTER_ALL, and INSERT_BEFORE_ALL, is three characters.

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
name	<String>	Read-write	Name of the pattern set to which to bind the string. Minimum length = 1
charset	<String>	Read-write	Character set associated with the characters in the string. Note: UTF-8 characters can be entered directly (if the UI supports it) or can be encoded as a sequence of hexadecimal bytes '\xNN'. For example, the UTF-8 character '?' can be encoded as '\xC3\xBC'. Possible values = ASCII, UTF_8
index	<Double>	Read-write	The index of the string associated with the patset.
__count	<Double>	Read-write	count parameter

Operations

(click to see Properties)

ADD:| DELETE:| GET| GET (ALL)| COUNT

Some options that you can use for each operations:

- **Getting warnings in response:**NITRO allows you to get warnings in an operation by specifying the "warning" query parameter as "yes". For example, to get warnings while connecting to the Citrix ADC appliance, the URL is as follows:

  http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/login?warning=yes

If any, the warnings are displayed in the response payload with the HTTP code "209 X-NITRO-WARNING".

- **Authenticated access for individual NITRO operations:**NITRO allows you to logon to the Citrix ADC appliance to perform individual operations. You can use this option instead of creating a NITRO session (using the login object) and then using that session to perform all operations,

  To do this, you must specify the username and password in the request header of the NITRO request as follows:

  X-NITRO-USER:<username>

  X-NITRO-PASS:<password>

  Note: In such cases, make sure that the request header DOES not include the following:

  Cookie:NITRO_AUTH_TOKEN=<tokenvalue>

`*Note:`*

Mandatory parameters are marked in redand placeholder content is marked in <green>.

add

URL:http://<Citrix-ADC-IP-address(NSIP)/nitro/v1/config/policypatset_pattern_binding

HTTP Method:PUT

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Content-Type:application/json

Request Payload:

```
{

"policypatset_pattern_binding":{

<b>"name":<String_value>,

</b><b>"String":<String_value>,
```

```
</b>"index":<Double_value>,

"charset":<String_value>

}}
```

Response:

HTTP Status Code on Success: 201 Created HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error

delete

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/policypatset_pattern_binding/name_value<String>

Query-parameters:

args

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/policypatset_pattern_binding/name_value<String>?**args=String:<String_value>**

HTTP Method:DELETE

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue>

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error

get

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/policypatset_pattern_binding/name_value<String>

Query-parameters:

filter

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/policypatset_pattern_binding/name_value<String>?**filter=property-name1:property-value1,property-name2:property-value2**

Use this query-parameter to get the filtered set of policypatset_pattern_binding resources configured on Citrix ADC. Filtering can be done on any of the properties of the resource.

pagination

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/policypatset_pattern_binding/name_value<String>?**pagesize=#no;pageno=#no**

Use this query-parameter to get the policypatset_pattern_binding resources in chunks.

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "policypatset_pattern_binding": [ {

"String":<String_value>,

"name":<String_value>,

"charset":<String_value>,

"index":<Double_value>

}]}
```

get (all)

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/policypatset_pattern_binding

Query-parameters:

bulkbindings

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/policypatset_pattern_binding?**bulkbindings=yes**

NITRO allows you to fetch bindings in bulk.

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "policypatset_pattern_binding": [ {

"String":<String_value>,

"name":<String_value>,

"charset":<String_value>,

"index":<Double_value>

}]}
```

count

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/policypatset_pattern_binding/name_value<String>?**count=yes**

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{"policypatset_pattern_binding": [ { "__count": "#no"} ] }
```

policystringmap

Configuration for string map resource.

Properties

(click to see Operations)

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
name	<String>	Read-write	Unique name for the string map. Not case sensitive. Must begin with an ASCII letter or underscore (_) character, and must

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
			consist only of ASCII alphanumeric or underscore characters. Must not begin with 're' or 'xp' or be a word reserved for use as an expression qualifier prefix (such as HTTP) or enumeration value (such as ASCII). Must not be the name of an existing named expression, pattern set, dataset, string map, or HTTP callout. Minimum length = 1
comment	<String>	Read-write	Comments associated with the string map.
__count	<Double>	Read-only	count parameter

Operations

(click to see Properties)

ADD| DELETE| UPDATE| UNSET| GET (ALL)| GET| COUNT

Some options that you can use for each operations:

- **Getting warnings in response:**NITRO allows you to get warnings in an operation by specifying the "warning" query parameter as "yes". For example, to get warnings while connecting to the Citrix ADC appliance, the URL is as follows:

  http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/login?warning=yes

  If any, the warnings are displayed in the response payload with the HTTP code "209 X-NITRO-WARNING".

- **Authenticated access for individual NITRO operations:**NITRO allows you to logon to the Citrix ADC appliance to perform individual operations. You can use this option instead of creating a NITRO session (using the login object) and then using that session to perform all operations,

  To do this, you must specify the username and password in the request header of the NITRO request as follows:

  X-NITRO-USER:<username>

  X-NITRO-PASS:<password>

  Note: In such cases, make sure that the request header DOES not include the following:

  Cookie:NITRO_AUTH_TOKEN=<tokenvalue>

`*Note:`*

Mandatory parameters are marked in redand placeholder content is marked in <green>.

add

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/policystringmap

HTTP Method:POST

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Content-Type:application/json

Request Payload:

```
{"policystringmap":{

<b>"name":<String_value>,

</b>"comment":<String_value>

}}
```

Response:

HTTP Status Code on Success: 201 Created HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error

delete

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/policystringmap/name_value<String>

HTTP Method:DELETE

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue>

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error

update

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/policystringmap

HTTP Method:PUT

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Content-Type:application/json

Request Payload:

```
{"policystringmap":{

<b>"name":<String_value>,

</b>"comment":<String_value>

}}
```

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error

unset

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/policystringmap?**action=unset**

HTTP Method:POST

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Content-Type:application/json

Request Payload:

```
{"policystringmap":{

<b>"name":<String_value>,

</b>"comment":true

}}
```

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error

get (all)

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/policystringmap

Query-parameters:

attrs

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/policystringmap?**attrs=property-name1,property-name2**

Use this query parameter to specify the resource details that you want to retrieve.

filter

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/policystringmap?**filter=property-name1:property-val1,property-name2:property-val2**

Use this query-parameter to get the filtered set of policystringmap resources configured on Citrix ADC. Filtering can be done on any of the properties of the resource.

view

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/policystringmap?**view=summary**

Note: By default, the retrieved results are displayed in detail view (?view=detail).

pagination

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/policystringmap?**pagesize=#no;pageno=#no**

Use this query-parameter to get the policystringmap resources in chunks.

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "policystringmap": [ {

"name":<String_value>,

"comment":<String_value>

}]}
```

get

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/policystringmap/name_value<String>

Query-parameters:

attrs

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/policystringmap/name_value<String>?**attrs=property-name1,property-name2**

Use this query parameter to specify the resource details that you want to retrieve.

view

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/policystringmap/name_value<String>?**view=summary**

Note: By default, the retrieved results are displayed in detail view (?view=detail).

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "policystringmap": [ {

"name":<String_value>,

"comment":<String_value>
```

```
}]}
```

count

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/policystringmap?**count=yes**

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "policystringmap": [ { "__count": "#no"} ] }
```

policystringmap_binding

Binding object which returns the resources bound to policystringmap.

Properties

(click to see Operations)

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
name	<String>	Read-write	Name of the string map to display. If a name is not provided, a list of all

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
			the configured string maps is shown. Minimum length = 1
policystringmap_pattern_binding	<policystringmap_pattern_binding[]>	Read-only	pattern that can be bound to policystringmap.

Operations

(click to see Properties)

GET| GET (ALL)

Some options that you can use for each operations:

- **Getting warnings in response:**NITRO allows you to get warnings in an operation by specifying the "warning" query parameter as "yes". For example, to get warnings while connecting to the Citrix ADC appliance, the URL is as follows:

  http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/login?warning=yes

  If any, the warnings are displayed in the response payload with the HTTP code "209 X-NITRO-WARNING".
- **Authenticated access for individual NITRO operations:**NITRO allows you to logon to the Citrix ADC appliance to perform individual operations. You can use this option instead of creating a NITRO session (using the login object) and then using that session to perform all operations,

  To do this, you must specify the username and password in the request header of the NITRO request as follows:

  X-NITRO-USER:<username>

  X-NITRO-PASS:<password>

  Note: In such cases, make sure that the request header DOES not include the following:

  Cookie:NITRO_AUTH_TOKEN=<tokenvalue>

`*Note:` *

Mandatory parameters are marked in redand placeholder content is marked in <green>.

get

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/policystringmap_binding/name_value<String>

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "policystringmap_binding": [ {

"name":<String_value>,

"policystringmap_pattern_binding":<policystringmap_pattern_binding[]_value
>

}]}
```

get (all)

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/policystringmap_binding

Query-parameters:

bulkbindings

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/policystringmap_binding?**bulkbindings=yes**

NITRO allows you to fetch bindings in bulk.

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "policystringmap_binding": [ {

"name":<String_value>,

"policystringmap_pattern_binding":<policystringmap_pattern_binding[]_value
>

}]}
```

policystringmap_pattern_binding

Binding object showing the pattern that can be bound to policystringmap.

Properties

(click to see Operations)

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
value	<String>	Read-write	Character string constituting the value associated with the key. This value is

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
			returned when processed data matches the associated key. Refer to the key parameter for details of the value character set. Minimum length = 1
name	<String>	Read-write	Name of the string map to which to bind the key-value pair. Minimum length = 1
key	<String>	Read-write	Character string constituting the key to be bound to the string map. The key is matched against the data processed by the operation that uses the string map. The default character set is ASCII. UTF-8 characters can be included if the character set is UTF-8. UTF-8 characters can be entered directly (if the UI supports it) or can be encoded as a sequence of hexadecimal bytes '\xNN'. For example, the UTF-8 character '?' can be encoded as '\xC3\xBC'. Minimum length = 1

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
__count	<Double>	Read-write	count parameter

Operations

(click to see Properties)

ADD:| DELETE:| GET| GET (ALL)| COUNT

Some options that you can use for each operations:

- **Getting warnings in response:**NITRO allows you to get warnings in an operation by specifying the "warning" query parameter as "yes". For example, to get warnings while connecting to the Citrix ADC appliance, the URL is as follows:

  http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/login?warning=yes

  If any, the warnings are displayed in the response payload with the HTTP code "209 X-NITRO-WARNING".
- **Authenticated access for individual NITRO operations:**NITRO allows you to logon to the Citrix ADC appliance to perform individual operations. You can use this option instead of creating a NITRO session (using the login object) and then using that session to perform all operations,

  To do this, you must specify the username and password in the request header of the NITRO request as follows:

  X-NITRO-USER:<username>

  X-NITRO-PASS:<password>

  Note: In such cases, make sure that the request header DOES not include the following:

  Cookie:NITRO_AUTH_TOKEN=<tokenvalue>

`*Note:`*

Mandatory parameters are marked in redand placeholder content is marked in <green>.

add

URL:http://<Citrix-ADC-IP-address(NSIP)/nitro/v1/config/policystringmap_pattern_binding

HTTP Method:PUT

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Content-Type:application/json

Request Payload:

```
{

"policystringmap_pattern_binding":{

<b>"name":<String_value>,

</b><b>"key":<String_value>,

</b><b>"value":<String_value></b>

}}
```

Response:

HTTP Status Code on Success: 201 Created HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error

delete

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/policystringmap_pattern_binding/name_value<String>

Query-parameters:

args

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/policystringmap_pattern_binding/name_value<String>?**args=key:<String_value>**

HTTP Method:DELETE

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue>

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error

get

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/policystringmap_pattern_binding/name_value<String>

Query-parameters:

filter

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/policystringmap_pattern_binding/name_value<String>?**filter=property-name1:property-value1,property-name2:property-value2**

Use this query-parameter to get the filtered set of policystringmap_pattern_binding resources configured on Citrix ADC. Filtering can be done on any of the properties of the resource.

pagination

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/policystringmap_pattern_binding/name_value<String>?**pagesize=#no;pageno=#no**

Use this query-parameter to get the policystringmap_pattern_binding resources in chunks.

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "policystringmap_pattern_binding": [ {

"value":<String_value>,

"name":<String_value>,

"key":<String_value>

}]}
```

get (all)

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/policystringmap_pattern_binding

Query-parameters:

bulkbindings

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/policystringmap_pattern_binding?**bulkbindings=yes**

NITRO allows you to fetch bindings in bulk.

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "policystringmap_pattern_binding": [ {

"value":<String_value>,

"name":<String_value>,

"key":<String_value>

}]}
```

count

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/policystringmap_pattern_binding/name_value<String>?**count=yes**

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{"policystringmap_pattern_binding": [ { "__count": "#no"} ] }
```

policyurlset

Configuration for URL set resource.

Properties

(click to see Operations)

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
name	<String>	Read-write	Unique name of the url set. Not case sensitive. Must begin with an ASCII letter or underscore (_) character and must contain only alphanumeric and underscore characters. Must not be the name of an existing named expression, pattern set, dataset, string map, or HTTP callout. Minimum length = 1 Maximum length = 127
comment	<String>	Read-write	Any comments to preserve information about this url set.
imported	<Boolean>	Read-write	when set, display shows all imported urlsets. Default value: 0
overwrite	<Boolean>	Read-write	Overwrites the existing file. Default value: 0
delimiter	<String>	Read-write	CSV file record delimiter. Default value: 44

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
rowseparator	<String>	Read-write	CSV file row separator. Default value: 10
url	<String>	Read-write	URL (protocol, host, path and file name) from where the CSV (comma separated file) file will be imported or exported. Each record/line will one entry within the urlset. The first field contains the URL pattern, subsequent fields contains the metadata, if available. HTTP, HTTPS and FTP protocols are supported. NOTE: The operation fails if the destination HTTPS server requires client certificate authentication for access. Minimum length = 1 Maximum length = 2047
interval	<Double>	Read-write	The interval, in seconds, rounded down to the nearest 15 minutes, at which the update of urlset occurs.

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
			Default value: 0 Minimum value = 0 Maximum value = 2592000
privateset	<Boolean>	Read-write	Prevent this urlset from being exported. Default value: 0
subdomainexactmatch	<Boolean>	Read-write	Force exact subdomain matching, ex. given an entry 'google.com' in the urlset, a request to 'news.google.com' won't match, if subdomainExactMatch is set. Default value: 0
canaryurl	<String>	Read-write	Add this URL to this urlset. Used for testing when contents of urlset is kept confidential. Minimum length = 1 Maximum length = 2047
patterncount	<Double>	Read-only	Number of patterns in this urlset. Default value: 0
__count	<Double>	Read-only	count parameter

Operations

(click to see Properties)

ADD| DELETE| IMPORT| CHANGE| EXPORT| GET (ALL)| GET| COUNT

Some options that you can use for each operations:

- **Getting warnings in response:**NITRO allows you to get warnings in an operation by specifying the "warning" query parameter as "yes". For example, to get warnings while connecting to the Citrix ADC appliance, the URL is as follows:

  http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/login?warning=yes

  If any, the warnings are displayed in the response payload with the HTTP code "209 X-NITRO-WARNING".

- **Authenticated access for individual NITRO operations:**NITRO allows you to logon to the Citrix ADC appliance to perform individual operations. You can use this option instead of creating a NITRO session (using the login object) and then using that session to perform all operations,

  To do this, you must specify the username and password in the request header of the NITRO request as follows:

  X-NITRO-USER:<username>

  X-NITRO-PASS:<password>

  Note: In such cases, make sure that the request header DOES not include the following:

  Cookie:NITRO_AUTH_TOKEN=<tokenvalue>

`*Note:`*

Mandatory parameters are marked in redand placeholder content is marked in <green>.

add

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/policyurlset

HTTP Method:POST

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Content-Type:application/json

Request Payload:

```
{"policyurlset":{

<b>"name":<String_value>,

</b>"comment":<String_value>

}}
```

Response:

HTTP Status Code on Success: 201 Created HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error

delete

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/policyurlset/name_value<String>

HTTP Method:DELETE

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue>

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error

Import

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/policyurlset?**action=Import**

HTTP Method:POST

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Content-Type:application/json

Request Payload:

```
{"policyurlset":{

<b>"name":<String_value>,

</b>"overwrite":<Boolean_value>,

"delimiter":<String_value>,

"rowseparator":<String_value>,

<b>"url":<String_value>,

</b>"interval":<Double_value>,

"privateset":<Boolean_value>,

"subdomainexactmatch":<Boolean_value>,

"canaryurl":<String_value>

}}
```

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error

change

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/policyurlset?**action=update**

HTTP Method:POST

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Content-Type:application/json

Request Payload:

```
{"policyurlset":{

<b>"name":<String_value>

</b>}}
```

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error

export

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/policyurlset?**action=export**

HTTP Method:POST

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Content-Type:application/json

Request Payload:

```
{"policyurlset":{

<b>"name":<String_value>,

</b><b>"url":<String_value>

</b>}}
```

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error

get (all)

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/policyurlset

Query-parameters:

args

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/policyurlset?**args=name:<String_value>,imported:<Boolean_value>**

Use this query-parameter to get policyurlset resources based on additional properties.

attrs

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/policyurlset?**attrs=property-name1,property-name2**

Use this query parameter to specify the resource details that you want to retrieve.

filter

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/policyurlset?**filter=property-name1:property-val1,property-name2:property-val2**

Use this query-parameter to get the filtered set of policyurlset resources configured on Citrix ADC. Filtering can be done on any of the properties of the resource.

view

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/policyurlset?**view=summary**

Note: By default, the retrieved results are displayed in detail view (?view=detail).

pagination

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/policyurlset?**pagesize=#no;pageno=#no**

Use this query-parameter to get the policyurlset resources in chunks.

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "policyurlset": [ {

name:<String_value>,imported:<Boolean_value>"comment":<String_value>,

"patterncount":<Double_value>,

"url":<String_value>,

"delimiter":<String_value>,

"rowseparator":<String_value>,

"interval":<Double_value>,

"canaryurl":<String_value>

}]}
```

get

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/policyurlset/name_value<String>

Query-parameters:

attrs

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/policyurlset/name_value<String>?**attrs=property-name1,property-name2**

Use this query parameter to specify the resource details that you want to retrieve.

view

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/policyurlset/name_value<String>?**view=summary**

Note: By default, the retrieved results are displayed in detail view (?view=detail).

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "policyurlset": [ {

name:<String_value>,imported:<Boolean_value>"comment":<String_value>,

"patterncount":<Double_value>,

"url":<String_value>,

"delimiter":<String_value>,

"rowseparator":<String_value>,

"interval":<Double_value>,

"canaryurl":<String_value>

}]}
```

count

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/policyurlset?**count=yes**

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "policyurlset": [ { "__count": "#no"} ] }
```

Priority queuing

Priority Queuing

Priority Queuing configuration. Priority Queuing ensures that key customer web sites get the timely responses they need. System provides priority-based queuing, which provides different levels of service to customers.

NAME	DESCRIPTION
pqbinding	Configuration for PQ bindings
pqpolicy	Configuration for PQ policy

pqbinding

Configuration for PQ bindings resource.

Properties

(click to see Operations)

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
vservername	<String>	Read-write	Name of the load balancing virtual server for which to display the priority queuing policy. Minimum length = 1
policyname	<String>	Read-only	The name of the priority queuing policy.
rule	<String>	Read-only	The condition for applying the policy.
priority	<Double>	Read-only	The priority of queuing the request.
weight	<Double>	Read-only	Weight.
qdepth	<Double>	Read-only	Queue Depth.
polqdepth	<Double>	Read-only	Policy Queue Depth.
hits	<Double>	Read-only	Total number of hits.
__count	<Double>	Read-only	count parameter

Operations

(click to see Properties)

GET (ALL)| COUNT

Some options that you can use for each operations:

- **Getting warnings in response:**NITRO allows you to get warnings in an operation by specifying the "warning" query parameter as "yes". For example, to get warnings while connecting to the Citrix ADC appliance, the URL is as follows:

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/login?warning=yes

If any, the warnings are displayed in the response payload with the HTTP code "209 X-NITRO-WARNING".

- **Authenticated access for individual NITRO operations:**NITRO allows you to logon to the Citrix ADC appliance to perform individual operations. You can use this option instead of creating a NITRO session (using the login object) and then using that session to perform all operations,

  To do this, you must specify the username and password in the request header of the NITRO request as follows:

  X-NITRO-USER:<username>

  X-NITRO-PASS:<password>

  Note: In such cases, make sure that the request header DOES not include the following:

  Cookie:NITRO_AUTH_TOKEN=<tokenvalue>

`*Note:`*

Mandatory parameters are marked in redand placeholder content is marked in <green>.

get (all)

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/pqbinding

Query-parameters:

args

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/pqbinding?**args=vservername:<String_value>,**

Use this query-parameter to get pqbinding resources based on additional properties.

attrs

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/pqbinding?**attrs=property-name1,property-name2**

Use this query parameter to specify the resource details that you want to retrieve.

filter

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/pqbinding?**filter=property-name1:property-val1,property-name2:property-val2**

Use this query-parameter to get the filtered set of pqbinding resources configured on Citrix ADC. Filtering can be done on any of the properties of the resource.

view

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/pqbinding?**view=summary**

Note: By default, the retrieved results are displayed in detail view (?view=detail).

pagination

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/pqbinding?**pagesize=#no;pageno=#no**

Use this query-parameter to get the pqbinding resources in chunks.

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "pqbinding": [ {

<b>vservername:<String_value>,</b>"policyname":<String_value>,

"rule":<String_value>,

"priority":<Double_value>,

"weight":<Double_value>,
```

```
"qdepth":<Double_value>,

"polqdepth":<Double_value>,

"hits":<Double_value>

}]}
```

count

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/pqbinding?**args=vservername:<String_value>,;count=yes**

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "pqbinding": [ { "__count": "#no"} ] }
```

pqpolicy

Configuration for PQ policy resource.

Properties

(click to see Operations)

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
policyname	<String>	Read-write	Name for the priority queuing policy. Must begin with a letter, number, or the underscore symbol (_). Other characters allowed, after the first character, are the hyphen (-), period (.) hash (#), space ( ), at (@), equals (=), and colon (:) characters. Minimum length = 1
rule	<String>	Read-write	Expression or name of a named expression, against which the request is evaluated. The priority queuing policy is applied if the rule evaluates to true. Note: * On the command line interface, if the expression includes blank spaces, the entire expression must be enclosed in double quotation marks. * If the expression itself includes double quotation marks, you must

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
			escape the quotations by using the \ character. * Alternatively, you can use single quotation marks to enclose the rule, in which case you will not have to escape the double quotation marks. * Maximum length of a string literal in the expression is 255 characters. A longer string can be split into smaller strings of up to 255 characters each, and the smaller strings concatenated with the + operator. For example, you can create a 500-character string as follows: '"<string of 255 characters>" + "<string of 245 characters>"'.
priority	<Double>	Read-write	Priority for queuing the request. If server resources are not available for a

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
			request that matches the configured rule, this option specifies a priority for queuing the request until the server resources are available again. Enter the value of positive_integer as 1, 2 or 3. The highest priority level is 1 and the lowest priority value is 3. Minimum value = 1 Maximum value = 3
weight	<Double>	Read-write	Weight of the priority. Each priority is assigned a weight according to which it is served when server resources are available. The weight for a higher priority request must be set higher than that of a lower priority request. To prevent delays for low-priority requests across multiple priority levels, you can configure weighted queuing for serving requests. The default weights for the priorities

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
			are: * Gold - Priority 1 - Weight 3 * Silver - Priority 2 - Weight 2 * Bronze - Priority 3 - Weight 1 Specify the weights as 0 through 101. A weight of 0 indicates that the particular priority level should be served only when there are no requests in any of the priority queues. A weight of 101 specifies a weight of infinity. This means that this priority level is served irrespective of the number of clients waiting in other priority queues. Minimum value = 0 Maximum value = 101
qdepth	<Double>	Read-write	Queue depth threshold value.

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
			When the queue size (number of requests in the queue) on the virtual server to which this policy is bound, increases to the specified qDepth value, subsequent requests are dropped to the lowest priority level. Default value: 0 Minimum value = 0 Maximum value = 4294967294
polqdepth	<Double>	Read-write	Policy queue depth threshold value. When the policy queue size (number of requests in all the queues belonging to this policy) increases to the specified polqDepth value, subsequent requests are dropped to the lowest priority level. Default value: 0 Minimum value = 0 Maximum value = 4294967294
hits	<Double>	Read-only	Total number of hits.
__count	<Double>	Read-only	count parameter

Operations

(click to see Properties)

ADD| DELETE| UPDATE| UNSET| GET (ALL)| GET| COUNT

Some options that you can use for each operations:

- **Getting warnings in response:**NITRO allows you to get warnings in an operation by specifying the "warning" query parameter as "yes". For example, to get warnings while connecting to the Citrix ADC appliance, the URL is as follows:

  http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/login?warning=yes

  If any, the warnings are displayed in the response payload with the HTTP code "209 X-NITRO-WARNING".

- **Authenticated access for individual NITRO operations:**NITRO allows you to logon to the Citrix ADC appliance to perform individual operations. You can use this option instead of creating a NITRO session (using the login object) and then using that session to perform all operations,

  To do this, you must specify the username and password in the request header of the NITRO request as follows:

  X-NITRO-USER:<username>

  X-NITRO-PASS:<password>

  Note: In such cases, make sure that the request header DOES not include the following:

  Cookie:NITRO_AUTH_TOKEN=<tokenvalue>

`*Note:`*

Mandatory parameters are marked in redand placeholder content is marked in <green>.

add

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/pqpolicy

HTTP Method:POST

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Content-Type:application/json

Request Payload:

```
{"pqpolicy":{

<b>"policyname":<String_value>,

</b><b>"rule":<String_value>,

</b><b>"priority":<Double_value>,

</b>"weight":<Double_value>,

"qdepth":<Double_value>,

"polqdepth":<Double_value>

}}
```

Response:

HTTP Status Code on Success: 201 Created HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error

delete

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/pqpolicy/policyname_value<String>

HTTP Method:DELETE

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue>

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error

update

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/pqpolicy

HTTP Method:PUT

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Content-Type:application/json

Request Payload:

```
{"pqpolicy":{

<b>"policyname":<String_value>,

</b>"weight":<Double_value>,

"qdepth":<Double_value>,

"polqdepth":<Double_value>

}}
```

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error

unset

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/pqpolicy?**action=unset**

HTTP Method:POST

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Content-Type:application/json

Request Payload:

```
{"pqpolicy":{

<b>"policyname":<String_value>,

</b>"weight":true,

"qdepth":true,

"polqdepth":true

}}
```

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error

get (all)

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/pqpolicy

Query-parameters:

attrs

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/pqpolicy?**attrs=property-name1,property-name2**

Use this query parameter to specify the resource details that you want to retrieve.

filter

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/pqpolicy?**filter=property-name1:property-val1,property-name2:property-val2**

Use this query-parameter to get the filtered set of pqpolicy resources configured on Citrix ADC. Filtering can be done on any of the properties of the resource.

view

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/pqpolicy?**view=summary**

Note: By default, the retrieved results are displayed in detail view (?view=detail).

pagination

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/pqpolicy?**pagesize=#no;pageno=#no**

Use this query-parameter to get the pqpolicy resources in chunks.

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "pqpolicy": [ {

"policyname":<String_value>,

"rule":<String_value>,

"priority":<Double_value>,

"weight":<Double_value>,

"qdepth":<Double_value>,

"polqdepth":<Double_value>,

"hits":<Double_value>

}]}
```

get

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/pqpolicy/policyname_value<String>

Query-parameters:

attrs

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/pqpolicy/policyname_value<String>?**attrs=property-name1,property-name2**

Use this query parameter to specify the resource details that you want to retrieve.

view

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/pqpolicy/policyname_value<String>?**view=summary**

Note: By default, the retrieved results are displayed in detail view (?view=detail).

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "pqpolicy": [ {

"policyname":<String_value>,

"rule":<String_value>,
```

```
"priority":<Double_value>,

"weight":<Double_value>,

"qdepth":<Double_value>,

"polqdepth":<Double_value>,

"hits":<Double_value>

}]}
```

count

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/pqpolicy?**count=yes**

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "pqpolicy": [ { "__count": "#no"} ] }
```

Protocol

Protocol

Protocol Configuration.

NAME	DESCRIPTION
protocolhttpband	Configuration for HTTP request/response band

protocolhttpband

Configuration for HTTP request/response band resource.

Properties

(click to see Operations)

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
reqbandsize	<Integer>	Read-write	Band size, in bytes, for HTTP request band statistics. For example, if you specify a band size of 100 bytes, statistics will be maintained and displayed for the following size ranges: 0 - 99 bytes 100 - 199 bytes 200 - 299 bytes and so on. Default value: 100 Minimum value = 50

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
respbandsize	<Integer>	Read-write	Band size, in bytes, for HTTP response band statistics. For example, if you specify a band size of 100 bytes, statistics will be maintained and displayed for the following size ranges: 0 - 99 bytes 100 - 199 bytes 200 - 299 bytes and so on. Default value: 1024 Minimum value = 50
type	<String>	Read-write	Type of statistics to display. Possible values = REQUEST, RESPONSE
nodeid	<Double>	Read-write	Unique number that identifies the cluster node. Minimum value = 0 Maximum value = 31
bandrange	<Integer>	Read-only	The range of the HTTP request/ response size, in bytes.

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
numberofbands	<Integer>	Read-only	The total number of http bands.
totalbandsize	<Double[]>	Read-only	The total size of all HTTP requests/ responses in this size range.
avgbandsize	<Double[]>	Read-only	The average size of all HTTP requests/ responses in this size range.
avgbandsizenew	<Double[]>	Read-only	The average size of all HTTP requests/ responses in this size range.
banddata	<Double[]>	Read-only	The total size of all HTTP requests/ responses in this size range, expressed as a percentage of the total size of all HTTP requests/responses.
banddatanew	<Double[]>	Read-only	The total size of all HTTP requests/ responses in this size range, expressed as a percentage of the total size of all HTTP requests/responses.
accesscount	<Double[]>	Read-only	The number of HTTP requests/responses in this size range.

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
accessratio	<Double[]>	Read-only	The number of HTTP requests/responses in this size range, expressed as a percentage of the total number of HTTP requests/ responses.
accessrationew	<Double[]>	Read-only	The number of HTTP requests/responses in this size range, expressed as a percentage of the total number of HTTP requests/ responses.
totals	<Integer[]>	Read-only	The total of totalBandSize, avgBandSize, BandData, accessCount, accessRatio respectively.

Operations

(click to see Properties)

UPDATE| UNSET| CLEAR| GET (ALL)

Some options that you can use for each operations:

- **Getting warnings in response:**NITRO allows you to get warnings in an operation by specifying the "warning" query parameter as "yes". For example, to get warnings while connecting to the Citrix ADC appliance, the URL is as follows:

  http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/login?warning=yes

If any, the warnings are displayed in the response payload with the HTTP code "209 X-NITRO-WARNING".

- **Authenticated access for individual NITRO operations:**NITRO allows you to logon to the Citrix ADC appliance to perform individual operations. You can use this option instead of creating a NITRO session (using the login object) and then using that session to perform all operations,

  To do this, you must specify the username and password in the request header of the NITRO request as follows:

  X-NITRO-USER:<username>

  X-NITRO-PASS:<password>

  Note: In such cases, make sure that the request header DOES not include the following:

  Cookie:NITRO_AUTH_TOKEN=<tokenvalue>

`*Note:`*

Mandatory parameters are marked in redand placeholder content is marked in <green>.

update

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/protocolhttpband

HTTP Method:PUT

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Content-Type:application/json

Request Payload:

```
{"protocolhttpband":{

"reqbandsize":<Integer_value>,

"respbandsize":<Integer_value>

}}
```

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error

unset

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/protocolhttpband?**action=unset**

HTTP Method:POST

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Content-Type:application/json

Request Payload:

```
{"protocolhttpband":{

"reqbandsize":true,

"respbandsize":true

}}
```

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error

clear

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/protocolhttpband?**action=clear**

HTTP Method:POST

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Content-Type:application/json

Request Payload:

```
{"protocolhttpband":{

<b>"type":<String_value>

</b>}}
```

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error

get (all)

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/protocolhttpband

Query-parameters:

args

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/protocolhttpband?**args=type:<String_value>,nodeid:<Double_value>**

Use this query-parameter to get protocolhttpband resources based on additional properties.

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "protocolhttpband": [ {

<b>type:<String_value>,</
b>nodeid:<Double_value>"bandrange":<Integer_value>,

"numberofbands":<Integer_value>,

"totalbandsize":<Double[]_value>,

"avgbandsize":<Double[]_value>,

"avgbandsizenew":<Double[]_value>,

"banddata":<Double[]_value>,

"banddatanew":<Double[]_value>,

"accesscount":<Double[]_value>,

"accessratio":<Double[]_value>,

"accessrationew":<Double[]_value>,

"totals":<Integer[]_value>,

"reqbandsize":<Integer_value>,

"respbandsize":<Integer_value>

}]}
```

RDP

Rdp

RDP configuration.

NAME	DESCRIPTION
rdpclientprofile	Configuration for RDP clientprofile
rdpconnections	Configuration for active rdp connections
rdpserverprofile	Configuration for RDP serverprofile

rdpclientprofile

Configuration for RDP clientprofile resource.

Properties

(click to see Operations)

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
name	<String>	Read-write	The name of the rdp profile. Minimum length = 1
rdpurloverride	<String>	Read-write	This setting determines whether the RDP parameters supplied in the vpn url override those specified in the RDP profile. Default value: ENABLE Possible values = ENABLE, DISABLE
redirectclipboard	<String>	Read-write	This setting corresponds to the Clipboard check box on the Local Resources tab under Options in RDC.

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
			Default value: ENABLE Possible values = ENABLE, DISABLE
redirectdrives	<String>	Read-write	This setting corresponds to the selections for Drives under More on the Local Resources tab under Options in RDC. Default value: DISABLE Possible values = ENABLE, DISABLE
redirectprinters	<String>	Read-write	This setting corresponds to the selection in the Printers check box on the Local Resources tab under Options in RDC. Default value: ENABLE Possible values = ENABLE, DISABLE
redirectcomports	<String>	Read-write	This setting corresponds to the selections for comports under More on the Local Resources tab under Options in RDC.

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
			Default value: DISABLE Possible values = ENABLE, DISABLE
redirectpnpdevices	<String>	Read-write	This setting corresponds to the selections for pnpdevices under More on the Local Resources tab under Options in RDC. Default value: DISABLE Possible values = ENABLE, DISABLE
keyboardhook	<String>	Read-write	This setting corresponds to the selection in the Keyboard drop-down list on the Local Resources tab under Options in RDC. Default value: InFullScreenMode Possible values = OnLocal, OnRemote, InFullScreenMode
audiocapturemode	<String>	Read-write	This setting corresponds to the selections in the Remote audio area on the Local

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
			Resources tab under Options in RDC. Default value: DISABLE Possible values = ENABLE, DISABLE
videoplaybackmode	<String>	Read-write	This setting determines if Remote Desktop Connection (RDC) will use RDP efficient multimedia streaming for video playback. Default value: ENABLE Possible values = ENABLE, DISABLE
multimonitorsupport	<String>	Read-write	Enable/Disable Multiple Monitor Support for Remote Desktop Connection (RDC). Default value: ENABLE Possible values = ENABLE, DISABLE
rdpcookievalidity	<Double>	Read-write	RDP cookie validity period. RDP cookie validity time is applicable for new connection and also for any re-connection that might happen, mostly due to

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
			network disruption or during fail-over. Default value: 60 Minimum value = 1 Maximum value = 86400
addusernameinrdpfile	<String>	Read-write	Add username in rdp file. Default value: NO Possible values = YES, NO
rdpfilename	<String>	Read-write	RDP file name to be sent to End User. Minimum length = 1
rdphost	<String>	Read-write	Fully-qualified domain name (FQDN) of the RDP Listener. Maximum length = 252
rdplistener	<String>	Read-write	IP address (or) Fully-qualified domain name(FQDN) of the RDP Listener with the port in the format IP:Port (or) FQDN:Port. Maximum length = 255
rdpcustomparams	<String>	Read-write	Option for RDP custom parameters settings (if any). Custom params

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
			needs to be separated by ';'. Default value: 0 Minimum length = 1
psk	<String>	Read-write	Pre shared key value. Default value: 0
randomizerdpfilename	<String>	Read-write	Will generate unique filename everytime rdp file is downloaded by appending output of time() function in the format <rdpfileName>_<time>.rdp. This tries to avoid the pop-up for replacement of existing rdp file during each rdp connection launch, hence providing better end-user experience. Default value: NO Possible values = YES, NO
rdplinkattribute	<String>	Read-write	Citrix Gateway allows the configuration of rdpLinkAttribute parameter which can be used to fetch a list of RDP servers(IP/FQDN) that a user can access, from an

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
			Authentication server attribute(Example: LDAP, SAML). Based on the list received, the RDP links will be generated and displayed to the user. Note: The Attribute mentioned in the rdpLinkAttribute should be fetched through corresponding authentication method.
__count	<Double>	Read-only	count parameter

Operations

(click to see Properties)

ADD| UPDATE| UNSET| DELETE| GET (ALL)| GET| COUNT

Some options that you can use for each operations:

- **Getting warnings in response:**NITRO allows you to get warnings in an operation by specifying the "warning" query parameter as "yes". For example, to get warnings while connecting to the Citrix ADC appliance, the URL is as follows:

  http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/login?warning=yes

  If any, the warnings are displayed in the response payload with the HTTP code "209 X-NITRO-WARNING".

- **Authenticated access for individual NITRO operations:**NITRO allows you to logon to the Citrix ADC appliance to perform individual operations. You can use this option instead of creating a NITRO session (using the login object) and then using that session to perform all operations,

To do this, you must specify the username and password in the request header of the NITRO request as follows:

X-NITRO-USER:<username>

X-NITRO-PASS:<password>

Note: In such cases, make sure that the request header DOES not include the following:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue>

`*Note:`*

Mandatory parameters are marked in redand placeholder content is marked in <green>.

add

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/rdpclientprofile

HTTP Method:POST

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Content-Type:application/json

Request Payload:

```
{"rdpclientprofile":{

<b>"name":<String_value>,

</b>"rdpurloverride":<String_value>,

"redirectclipboard":<String_value>,

"redirectdrives":<String_value>,

"redirectprinters":<String_value>,

"redirectcomports":<String_value>,

"redirectpnpdevices":<String_value>,
```

```
"keyboardhook":<String_value>,

"audiocapturemode":<String_value>,

"videoplaybackmode":<String_value>,

"multimonitorsupport":<String_value>,

"rdpcookievalidity":<Double_value>,

"addusernameinrdpfile":<String_value>,

"rdpfilename":<String_value>,

"rdphost":<String_value>,

"rdplistener":<String_value>,

"rdpcustomparams":<String_value>,

"psk":<String_value>,

"randomizerdpfilename":<String_value>,

"rdplinkattribute":<String_value>

}}
```

Response:

HTTP Status Code on Success: 201 Created HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error

update

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/rdpclientprofile

HTTP Method:PUT

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Content-Type:application/json

Request Payload:

```
{"rdpclientprofile":{

<b>"name":<String_value>,

</b>"rdpurloverride":<String_value>,

"redirectclipboard":<String_value>,

"redirectdrives":<String_value>,

"redirectprinters":<String_value>,

"redirectcomports":<String_value>,

"redirectpnpdevices":<String_value>,

"keyboardhook":<String_value>,

"audiocapturemode":<String_value>,

"videoplaybackmode":<String_value>,

"multimonitorsupport":<String_value>,

"rdpcookievalidity":<Double_value>,

"addusernameinrdpfile":<String_value>,

"rdpfilename":<String_value>,

"rdphost":<String_value>,

"rdplistener":<String_value>,

"rdpcustomparams":<String_value>,

"psk":<String_value>,
```

```
"randomizerdpfilename":<String_value>,

"rdplinkattribute":<String_value>

}}
```

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error

unset

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/rdpclientprofile?**action=unset**

HTTP Method:POST

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Content-Type:application/json

Request Payload:

```
{"rdpclientprofile":{

<b>"name":<String_value>,

</b>"rdpurloverride":true,

"redirectclipboard":true,

"redirectdrives":true,

"redirectprinters":true,

"redirectcomports":true,

"redirectpnpdevices":true,
```

```
"keyboardhook":true,

"audiocapturemode":true,

"videoplaybackmode":true,

"multimonitorsupport":true,

"rdpcookievalidity":true,

"addusernameinrdpfile":true,

"rdpfilename":true,

"rdphost":true,

"rdplistener":true,

"rdpcustomparams":true,

"psk":true,

"randomizerdpfilename":true,

"rdplinkattribute":true

}}
```

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error

delete

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/rdpclientprofile/name_value<String>

HTTP Method:DELETE

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue>

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error

get (all)

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/rdpclientprofile

Query-parameters:

attrs

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/rdpclientprofile?**attrs=property-name1,property-name2**

Use this query parameter to specify the resource details that you want to retrieve.

filter

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/rdpclientprofile?**filter=property-name1:property-val1,property-name2:property-val2**

Use this query-parameter to get the filtered set of rdpclientprofile resources configured on Citrix ADC. Filtering can be done on any of the properties of the resource.

view

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/rdpclientprofile?**view=summary**

Note: By default, the retrieved results are displayed in detail view (?view=detail).

pagination

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/rdpclientprofile?**pagesize=#no;pageno=#no**

Use this query-parameter to get the rdpclientprofile resources in chunks.

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "rdpclientprofile": [ {

"name":<String_value>,

"rdpurloverride":<String_value>,

"redirectclipboard":<String_value>,

"redirectdrives":<String_value>,

"redirectprinters":<String_value>,

"redirectcomports":<String_value>,

"redirectpnpdevices":<String_value>,

"keyboardhook":<String_value>,

"audiocapturemode":<String_value>,

"videoplaybackmode":<String_value>,

"multimonitorsupport":<String_value>,

"rdpcookievalidity":<Double_value>,

"addusernameinrdpfile":<String_value>,

"rdpfilename":<String_value>,

"rdphost":<String_value>,
```

```
"rdplistener":<String_value>,

"rdpcustomparams":<String_value>,

"psk":<String_value>,

"randomizerdpfilename":<String_value>,

"rdplinkattribute":<String_value>

}]}
```

get

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/rdpclientprofile/name_value<String>

Query-parameters:

attrs

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/rdpclientprofile/name_value<String>?**attrs=property-name1,property-name2**

Use this query parameter to specify the resource details that you want to retrieve.

view

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/rdpclientprofile/name_value<String>?**view=summary**

Note: By default, the retrieved results are displayed in detail view (?view=detail).

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "rdpclientprofile": [ {

"name":<String_value>,

"rdpurloverride":<String_value>,

"redirectclipboard":<String_value>,

"redirectdrives":<String_value>,

"redirectprinters":<String_value>,

"redirectcomports":<String_value>,

"redirectpnpdevices":<String_value>,

"keyboardhook":<String_value>,

"audiocapturemode":<String_value>,

"videoplaybackmode":<String_value>,

"multimonitorsupport":<String_value>,

"rdpcookievalidity":<Double_value>,

"addusernameinrdpfile":<String_value>,

"rdpfilename":<String_value>,

"rdphost":<String_value>,

"rdplistener":<String_value>,

"rdpcustomparams":<String_value>,

"psk":<String_value>,
```

```
"randomizerdpfilename":<String_value>,

"rdplinkattribute":<String_value>

}]}
```

count

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/rdpclientprofile?**count=yes**

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "rdpclientprofile": [ { "__count": "#no"} ] }
```

rdpconnections

Configuration for active rdp connections resource.

Properties

(click to see Operations)

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
username	<String>	Read-write	User name for which to display connections. Minimum length = 1
all	<Boolean>	Read-write	Terminate all active rdpconnections.
endpointip	<String>	Read-only	The client IP address.
endpointport	<Integer>	Read-only	The client port. Range 1 - 65535 * in CLI is represented as 65535 in NITRO API
targetip	<String>	Read-only	The Server IP address.
targetport	<Integer>	Read-only	The server port. Range 1 - 65535 * in CLI is represented as 65535 in NITRO API
peid	<Double>	Read-only	Core id of the session owner.
__count	<Double>	Read-only	count parameter

Operations

(click to see Properties)

KILL| GET (ALL)| COUNT

Some options that you can use for each operations:

- **Getting warnings in response:**NITRO allows you to get warnings in an operation by specifying the "warning" query parameter as "yes". For example, to get warnings while connecting to the Citrix ADC appliance, the URL is as follows:

  http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/login?warning=yes

  If any, the warnings are displayed in the response payload with the HTTP code "209 X-NITRO-WARNING".

- **Authenticated access for individual NITRO operations:**NITRO allows you to logon to the Citrix ADC appliance to perform individual operations. You can use this option instead of creating a NITRO session (using the login object) and then using that session to perform all operations,

  To do this, you must specify the username and password in the request header of the NITRO request as follows:

  X-NITRO-USER:<username>

  X-NITRO-PASS:<password>

  Note: In such cases, make sure that the request header DOES not include the following:

  Cookie:NITRO_AUTH_TOKEN=<tokenvalue>

`*Note:`*

Mandatory parameters are marked in redand placeholder content is marked in <green>.

kill

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/rdpconnections?**action=kill**

HTTP Method:POST

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Content-Type:application/json

Request Payload:

```
{"rdpconnections":{

"username":<String_value>,

"all":<Boolean_value>

}}
```

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error

get (all)

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/rdpconnections

Query-parameters:

args

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/rdpconnections?**args=username:<String_value>**

Use this query-parameter to get rdpconnections resources based on additional properties.

attrs

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/rdpconnections?**attrs=property-name1,property-name2**

Use this query parameter to specify the resource details that you want to retrieve.

filter

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/rdpconnections?**filter=property-name1:property-val1,property-name2:property-val2**

Use this query-parameter to get the filtered set of rdpconnections resources configured on Citrix ADC. Filtering can be done on any of the properties of the resource.

view

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/rdpconnections?**view=summary**

Note: By default, the retrieved results are displayed in detail view (?view=detail).

pagination

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/rdpconnections?**pagesize=#no;pageno=#no**

Use this query-parameter to get the rdpconnections resources in chunks.

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "rdpconnections": [ {

username:<String_value>"endpointip":<String_value>,

"endpointport":<Integer_value>,

"targetip":<String_value>,

"targetport":<Integer_value>,

"peid":<Double_value>

}]}
```

count

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/rdpconnections?**count=yes**

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "rdpconnections": [ { "__count": "#no"} ] }
```

rdpserverprofile

Configuration for RDP serverprofile resource.

Properties

(click to see Operations)

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
name	<String>	Read-write	The name of the rdp server profile. Minimum length = 1 Maximum length = 32
rdpip	<String>	Read-write	IPv4 or IPv6 address of RDP listener. This

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
			terminates client RDP connections. Minimum length = 1
rdpport	<Double>	Read-write	TCP port on which the RDP connection is established. Default value: 3389 Minimum value = 1 Maximum value = 65535
psk	<String>	Read-write	Pre shared key value.
rdpredirection	<String>	Read-write	Enable/Disable RDP redirection support. This needs to be enabled in presence of connection broker or session directory with IP cookie(msts cookie) based redirection support. Default value: DISABLE Possible values = ENABLE, DISABLE
__count	<Double>	Read-only	count parameter

Operations

(click to see Properties)

ADD| UPDATE| UNSET| DELETE| GET (ALL)| GET| COUNT

Some options that you can use for each operations:

- **Getting warnings in response:**NITRO allows you to get warnings in an operation by specifying the "warning" query parameter as "yes". For example, to get warnings while connecting to the Citrix ADC appliance, the URL is as follows:

  http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/login?warning=yes

  If any, the warnings are displayed in the response payload with the HTTP code "209 X-NITRO-WARNING".

- **Authenticated access for individual NITRO operations:**NITRO allows you to logon to the Citrix ADC appliance to perform individual operations. You can use this option instead of creating a NITRO session (using the login object) and then using that session to perform all operations,

  To do this, you must specify the username and password in the request header of the NITRO request as follows:

  X-NITRO-USER:<username>

  X-NITRO-PASS:<password>

  Note: In such cases, make sure that the request header DOES not include the following:

  Cookie:NITRO_AUTH_TOKEN=<tokenvalue>

`*Note:`*

Mandatory parameters are marked in redand placeholder content is marked in <green>.

add

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/rdpserverprofile

HTTP Method:POST

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Content-Type:application/json

Request Payload:

```
{"rdpserverprofile":{
```

```
<b>"name":<String_value>,

</b>"rdpip":<String_value>,

"rdpport":<Double_value>,

<b>"psk":<String_value>,

</b>"rdpredirection":<String_value>

}}
```

Response:

HTTP Status Code on Success: 201 Created HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error

update

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/rdpserverprofile

HTTP Method:PUT

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Content-Type:application/json

Request Payload:

```
{"rdpserverprofile":{

<b>"name":<String_value>,

</b>"rdpip":<String_value>,

"rdpport":<Double_value>,

"psk":<String_value>,
```

```
"rdpredirection":<String_value>

}}
```

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error

unset

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/rdpserverprofile?**action=unset**

HTTP Method:POST

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Content-Type:application/json

Request Payload:

```
{"rdpserverprofile":{

<b>"name":<String_value>,

</b>"rdpip":true,

"rdpport":true,

"psk":true,

"rdpredirection":true

}}
```

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error

delete

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/rdpserverprofile/name_value<String>

HTTP Method:DELETE

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue>

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error

get (all)

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/rdpserverprofile

Query-parameters:

attrs

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/rdpserverprofile?**attrs=property-name1,property-name2**

Use this query parameter to specify the resource details that you want to retrieve.

filter

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/rdpserverprofile?**filter=property-name1:property-val1,property-name2:property-val2**

Use this query-parameter to get the filtered set of rdpserverprofile resources configured on Citrix ADC. Filtering can be done on any of the properties of the resource.

view

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/rdpserverprofile?**view=summary**

Note: By default, the retrieved results are displayed in detail view (?view=detail).

pagination

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/rdpserverprofile?**pagesize=#no;pageno=#no**

Use this query-parameter to get the rdpserverprofile resources in chunks.

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "rdpserverprofile": [ {

"name":<String_value>,

"rdpip":<String_value>,

"rdpport":<Double_value>,

"psk":<String_value>,

"rdpredirection":<String_value>

}]}
```

get

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/rdpserverprofile/name_value<String>

Query-parameters:

attrs

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/rdpserverprofile/name_value<String>?**attrs=property-name1,property-name2**

Use this query parameter to specify the resource details that you want to retrieve.

view

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/rdpserverprofile/name_value<String>?**view=summary**

Note: By default, the retrieved results are displayed in detail view (?view=detail).

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "rdpserverprofile": [ {

"name":<String_value>,

"rdpip":<String_value>,

"rdpport":<Double_value>,

"psk":<String_value>,

"rdpredirection":<String_value>
```

```
}]}
```

count

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/rdpserverprofile?**count=yes**

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "rdpserverprofile": [ { "__count": "#no"} ] }
```

Reputation

Reputation

Reputation services configuration.

NAME	DESCRIPTION
reputationsettings	Configuration for Reputation service settings

reputationsettings

Configuration for Reputation service settings resource.

Properties

(click to see Operations)

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
proxyserver	<String>	Read-write	Proxy server IP to get Reputation data. Minimum length = 1
proxyport	<Double>	Read-write	Proxy server port.

Operations

(click to see Properties)

UPDATE| UNSET| GET (ALL)

Some options that you can use for each operations:

- **Getting warnings in response:**NITRO allows you to get warnings in an operation by specifying the "warning" query parameter as "yes". For example, to get warnings while connecting to the Citrix ADC appliance, the URL is as follows:

  http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/login?warning=yes

  If any, the warnings are displayed in the response payload with the HTTP code "209 X-NITRO-WARNING".

- **Authenticated access for individual NITRO operations:**NITRO allows you to logon to the Citrix ADC appliance to perform individual operations. You can use this option instead of creating a NITRO session (using the login object) and then using that session to perform all operations,

  To do this, you must specify the username and password in the request header of the NITRO request as follows:

  X-NITRO-USER:<username>

  X-NITRO-PASS:<password>

  Note: In such cases, make sure that the request header DOES not include the following:

  Cookie:NITRO_AUTH_TOKEN=<tokenvalue>

`*Note:` *

Mandatory parameters are marked in redand placeholder content is marked in <green>.

update

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/reputationsettings

HTTP Method:PUT

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Content-Type:application/json

Request Payload:

```
{"reputationsettings":{

"proxyserver":<String_value>,

"proxyport":<Double_value>

}}
```

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error

unset

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/reputationsettings?**action=unset**

HTTP Method:POST

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Content-Type:application/json

Request Payload:

```
{"reputationsettings":{

"proxyserver":true,

"proxyport":true

}}
```

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error

get (all)

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/reputationsettings

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "reputationsettings": [ {

"proxyserver":<String_value>,

"proxyport":<Double_value>
```

```
}]}
```

Responder

Responder

Responder configuration.

NAME	DESCRIPTION
responderaction	Configuration for responder action
responderglobal_binding	Binding object showing the resources that can be bound to responderglobal
responderglobal_responderpolicy_binding	Binding object showing the responderpolicy that can be bound to responderglobal
responderhtmlpage	Configuration for Responder HTML page
responderparam	Configuration for responser parameter
responderpolicy	Configuration for responder policy
responderpolicy_binding	Binding object showing the resources that can be bound to responderpolicy
responderpolicy_crvserver_binding	Binding object showing the crvserver that can be bound to responderpolicy
responderpolicy_csvserver_binding	Binding object showing the csvserver that can be bound to responderpolicy
responderpolicy_lbvserver_binding	Binding object showing the lbvserver that can be bound to responderpolicy
responderpolicy_responderglobal_binding	Binding object showing the responderglobal that can be bound to responderpolicy

NAME	DESCRIPTION
responderpolicy_responderpolicylabel_binding	Binding object showing the responderpolicylabel that can be bound to responderpolicy
responderpolicylabel	Configuration for responder policy label
responderpolicylabel_binding	Binding object showing the resources that can be bound to responderpolicylabel
responderpolicylabel_policybinding_binding	Binding object showing the policybinding that can be bound to responderpolicylabel
responderpolicylabel_responderpolicy_binding	Binding object showing the responderpolicy that can be bound to responderpolicylabel

responderaction

Configuration for responder action resource.

Properties

(click to see Operations)

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
name	<String>	Read-write	Name for the responder action. Must begin with a letter, number, or the underscore character (_), and must contain only letters, numbers, and the hyphen (-), period (.) hash (#), space ( ), at (@), equals (=), colon (:), and underscore characters. Can be changed after the

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
			responder policy is added. The following requirement applies only to the Citrix ADC CLI: If the name includes one or more spaces, enclose the name in double or single quotation marks (for example, "my responder action" or 'my responder action').
type	<String>	Read-write	Type of responder action. Available settings function as follows: * respondwith <target> - Respond to the request with the expression specified as the target. * respondwithhtmlpage - Respond to the request with the uploaded HTML page object specified as the target.

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
			* redirect - Redirect the request to the URL specified as the target. * sqlresponse_ok - Send an SQL OK response. * sqlresponse_error - Send an SQL ERROR response. Possible values = noop, respondwith, redirect, respondwithhtmlpage, sqlresponse_ok, sqlresponse_error
target	<String>	Read-write	Expression specifying what to respond with. Typically a URL for redirect policies or a default-syntax expression. In addition to Citrix ADC default-syntax expressions that refer to information in the request, a stringbuilder expression can contain text and HTML, and simple escape codes that define new lines and paragraphs. Enclose each stringbuilder

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
			expression element (either a Citrix ADC default-syntax expression or a string) in double quotation marks. Use the plus (+) character to join the elements. Examples: 1) Respondwith expression that sends an HTTP 1.1 200 OK response: "HTTP/1.1 200 OK\r\n\r\n" 2) Redirect expression that redirects user to the specified web host and appends the request URL to the redirect. "http://backupsite2.com" + HTTP.REQ.URL 3) Respondwith expression that sends an HTTP 1.1 404 Not Found response with the request URL

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
			included in the response: "HTTP/1.1 404 Not Found\r\n\r\n"+ "HTTP.REQ.URL.HTTP_URL_SAFE" + "does not exist on the web server." The following requirement applies only to the Citrix ADC CLI: Enclose the entire expression in single quotation marks. (Citrix ADC expression elements should be included inside the single quotation marks for the entire expression, but do not need to be enclosed in double quotation marks.).
htmlpage	<String>	Read-write	For respondwithhtmlpage policies, name of the HTML page object to use as the response. You must first import the page object. Minimum length = 1

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
bypasssafetycheck	<String>	Read-write	Bypass the safety check, allowing potentially unsafe expressions. An unsafe expression in a response is one that contains references to request elements that might not be present in all requests. If a response refers to a missing request element, an empty string is used instead. Default value: NO Possible values = YES, NO
comment	<String>	Read-write	Comment. Any type of information about this responder action.
responsestatuscode	<Double>	Read-write	HTTP response status code, for example 200, 302, 404, etc. The default value for the redirect action type is 302 and for respondwithhtmlpage is 200. Minimum value = 100 Maximum value = 599

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
reasonphrase	<String>	Read-write	Expression specifying the reason phrase of the HTTP response. The reason phrase may be a string literal with quotes or a PI expression. For example: "Invalid URL: " + HTTP.REQ.URL. Minimum length = 1
newname	<String>	Read-write	New name for the responder action. Must begin with a letter, number, or the underscore character (_), and must contain only letters, numbers, and the hyphen (-), period (.) hash (#), space ( ), at (@), equals (=), colon (:), and underscore characters. The following requirement applies only to the Citrix ADC CLI: If the name includes one or more spaces, enclose the name in double or single quotation

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
			marks (for example, "my responder action" or my responder action'). Minimum length = 1
hits	<Double>	Read-only	The number of times the action has been taken.
referencecount	<Double>	Read-only	The number of references to the action.
undefhits	<Double>	Read-only	The number of times the action resulted in UNDEF.
__count	<Double>	Read-only	count parameter

Operations

(click to see Properties)

ADD| DELETE| UPDATE| UNSET| RENAME| GET (ALL)| GET| COUNT

Some options that you can use for each operations:

- **Getting warnings in response:**NITRO allows you to get warnings in an operation by specifying the "warning" query parameter as "yes". For example, to get warnings while connecting to the Citrix ADC appliance, the URL is as follows:

  http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/login?warning=yes

  If any, the warnings are displayed in the response payload with the HTTP code "209 X-NITRO-WARNING".
- **Authenticated access for individual NITRO operations:**NITRO allows you to logon to the Citrix ADC appliance to perform individual operations. You can use this option instead of

creating a NITRO session (using the login object) and then using that session to perform all operations,

To do this, you must specify the username and password in the request header of the NITRO request as follows:

X-NITRO-USER:<username>

X-NITRO-PASS:<password>

Note: In such cases, make sure that the request header DOES not include the following:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue>

`*Note:`*

Mandatory parameters are marked in redand placeholder content is marked in <green>.

add

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/responderaction

HTTP Method:POST

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Content-Type:application/json

Request Payload:

```
{"responderaction":{

<b>"name":<String_value>,

</b><b>"type":<String_value>,

</b>"target":<String_value>,

"htmlpage":<String_value>,

"bypasssafetycheck":<String_value>,

"comment":<String_value>,
```

```
"responsestatuscode":<Double_value>,

"reasonphrase":<String_value>

}}
```

Response:

HTTP Status Code on Success: 201 Created HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error

delete

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/responderaction/name_value<String>

HTTP Method:DELETE

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue>

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error

update

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/responderaction

HTTP Method:PUT

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Content-Type:application/json

Request Payload:

```
{"responderaction":{
```

```
<b>"name":<String_value>,

</b>"target":<String_value>,

"bypasssafetycheck":<String_value>,

"htmlpage":<String_value>,

"responsestatuscode":<Double_value>,

"reasonphrase":<String_value>,

"comment":<String_value>

}}
```

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error

unset

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/responderaction?**action=unset**

HTTP Method:POST

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Content-Type:application/json

Request Payload:

```
{"responderaction":{

<b>"name":<String_value>,

</b>"responsestatuscode":true,
```

```
"reasonphrase":true,

"comment":true

}}
```

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error

rename

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/responderaction?**action=rename**

HTTP Method:POST

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Content-Type:application/json

Request Payload:

```
{"responderaction":{

<b>"name":<String_value>,

</b><b>"newname":<String_value>

</b>}}
```

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error

get (all)

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/responderaction

Query-parameters:

attrs

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/responderaction?**attrs=property-name1,property-name2**

Use this query parameter to specify the resource details that you want to retrieve.

filter

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/responderaction?**filter=property-name1:property-val1,property-name2:property-val2**

Use this query-parameter to get the filtered set of responderaction resources configured on Citrix ADC. Filtering can be done on any of the properties of the resource.

view

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/responderaction?**view=summary**

Note: By default, the retrieved results are displayed in detail view (?view=detail).

pagination

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/responderaction?**pagesize=#no;pageno=#no**

Use this query-parameter to get the responderaction resources in chunks.

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "responderaction": [ {

"name":<String_value>,

"type":<String_value>,

"target":<String_value>,

"htmlpage":<String_value>,

"bypasssafetycheck":<String_value>,

"hits":<Double_value>,

"referencecount":<Double_value>,

"undefhits":<Double_value>,

"comment":<String_value>,

"responsestatuscode":<Double_value>,

"reasonphrase":<String_value>

}]}
```

get

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/responderaction/name_value<String>

Query-parameters:

attrs

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/responderaction/name_value<String>?**attrs=property-name1,property-name2**

Use this query parameter to specify the resource details that you want to retrieve.

view

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/responderaction/name_value<String>?**view=summary**

Note: By default, the retrieved results are displayed in detail view (?view=detail).

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "responderaction": [ {

"name":<String_value>,

"type":<String_value>,

"target":<String_value>,

"htmlpage":<String_value>,

"bypasssafetycheck":<String_value>,

"hits":<Double_value>,

"referencecount":<Double_value>,

"undefhits":<Double_value>,
```

```
"comment":<String_value>,

"responsestatuscode":<Double_value>,

"reasonphrase":<String_value>

}]}
```

count

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/responderaction?**count=yes**

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "responderaction": [ { "__count": "#no"} ] }
```

responderglobal_binding

Binding object which returns the resources bound to responderglobal.

Properties

(click to see Operations)

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
responderglobal_responderpolicy_binding	<responderglobal_responderpolicy_binding[]>	Read-only	responderpolicy that can be bound to responderglobal.

Operations

(click to see Properties)

GET| GET (ALL)

Some options that you can use for each operations:

- **Getting warnings in response:**NITRO allows you to get warnings in an operation by specifying the "warning" query parameter as "yes". For example, to get warnings while connecting to the Citrix ADC appliance, the URL is as follows:

  http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/login?warning=yes

  If any, the warnings are displayed in the response payload with the HTTP code "209 X-NITRO-WARNING".

- **Authenticated access for individual NITRO operations:**NITRO allows you to logon to the Citrix ADC appliance to perform individual operations. You can use this option instead of creating a NITRO session (using the login object) and then using that session to perform all operations,

  To do this, you must specify the username and password in the request header of the NITRO request as follows:

  X-NITRO-USER:<username>

  X-NITRO-PASS:<password>

  Note: In such cases, make sure that the request header DOES not include the following:

  Cookie:NITRO_AUTH_TOKEN=<tokenvalue>

`*Note:`*

Mandatory parameters are marked in redand placeholder content is marked in <green>.

get

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/responderglobal_binding

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "responderglobal_binding": [ {

"responderglobal_responderpolicy_binding":<responderglobal_responderpolicy
_binding[]_value>

}]}
```

get (all)

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/responderglobal_binding

Query-parameters:

bulkbindings

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/responderglobal_binding?**bulkbindings=yes**

NITRO allows you to fetch bindings in bulk.

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "responderglobal_binding": [ {

"responderglobal_responderpolicy_binding":<responderglobal_responderpolicy
_binding[]_value>

}]}
```

responderglobal_responderpolicy_binding

Binding object showing the responderpolicy that can be bound to responderglobal.

Properties

(click to see Operations)

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
priority	<Double>	Read-write	Specifies the priority of the policy.
globalbindtype	<String>	Read-write	. Default value: SYSTEM_GLOBAL Possible values = SYSTEM_GLOBAL,

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
			VPN_GLOBAL, RNAT_GLOBAL
policyname	<String>	Read-write	Name of the responder policy.
labelname	<String>	Read-write	Name of the policy label to invoke. If the current policy evaluates to TRUE, the invoke parameter is set, and Label Type is policylabel.
gotopriorityexpression	<String>	Read-write	Expression specifying the priority of the next policy which will get evaluated if the current policy rule evaluates to TRUE.
invoke	<Boolean>	Read-write	If the current policy evaluates to TRUE, terminate evaluation of policies bound to the current policy label, and then forward the request to the specified virtual server or evaluate the specified policy label.
type	<String>	Read-write	Specifies the bind point whose policies you want to display. Available settings function as follows: * REQ_OVERRIDE - Request override.

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
			Binds the policy to the priority request queue. * REQ_DEFAULT - Binds the policy to the default request queue. * OTHERTCP_REQ_OVERRIDE - Binds the policy to the non-HTTP TCP priority request queue. * OTHERTCP_REQ_DEFAULT - Binds the policy to the non-HTTP TCP default request queue.. * SIPUDP_REQ_OVERRIDE - Binds the policy to the SIP UDP priority response queue.. * SIPUDP_REQ_DEFAULT - Binds the policy to the SIP UDP default response queue. * RADIUS_REQ_OVERRIDE - Binds the policy to the RADIUS priority response queue.. * RADIUS_REQ_DEFAULT - Binds the policy to the RADIUS default response queue. * MSSQL_REQ_OVERRIDE - Binds the policy to the Microsoft SQL

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
			priority response queue.. * MSSQL_REQ_DEFAULT - Binds the policy to the Microsoft SQL default response queue. * MYSQL_REQ_OVERRIDE - Binds the policy to the MySQL priority response queue. * MYSQL_REQ_DEFAULT - Binds the policy to the MySQL default response queue. Possible values = REQ_OVERRIDE, REQ_DEFAULT, OVERRIDE, DEFAULT, OTHERTCP_REQ_OVERRIDE, OTHERTCP_REQ_DEFAULT, SIPUDP_REQ_OVERRIDE, SIPUDP_REQ_DEFAULT, SIPTCP_REQ_OVERRIDE, SIPTCP_REQ_DEFAULT, MSSQL_REQ_OVERRIDE, MSSQL_REQ_DEF

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
			AULT, MYSQL_REQ_OVERRIDE, MYSQL_REQ_DEFAULT, NAT_REQ_OVERRIDE, NAT_REQ_DEFAULT, DIAMETER_REQ_OVERRIDE, DIAMETER_REQ_DEFAULT, RADIUS_REQ_OVERRIDE, RADIUS_REQ_DEFAULT, DNS_REQ_OVERRIDE, DNS_REQ_DEFAULT
labeltype	<String>	Read-write	Type of invocation, Available settings function as follows: * vserver - Forward the request to the specified virtual server. * policylabel - Invoke the specified policy label. Possible values = vserver, policylabel
numpol	<Double>	Read-only	number of polices bound to label.

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
flowtype	<Double>	Read-only	flowtype of the bound responder policy.
__count	<Double>	Read-only	count parameter

Operations

(click to see Properties)

ADD:| DELETE:| GET| GET (ALL)| COUNT

Some options that you can use for each operations:

- **Getting warnings in response:**NITRO allows you to get warnings in an operation by specifying the "warning" query parameter as "yes". For example, to get warnings while connecting to the Citrix ADC appliance, the URL is as follows:

  http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/login?warning=yes

  If any, the warnings are displayed in the response payload with the HTTP code "209 X-NITRO-WARNING".

- **Authenticated access for individual NITRO operations:**NITRO allows you to logon to the Citrix ADC appliance to perform individual operations. You can use this option instead of creating a NITRO session (using the login object) and then using that session to perform all operations,

  To do this, you must specify the username and password in the request header of the NITRO request as follows:

  X-NITRO-USER:<username>

  X-NITRO-PASS:<password>

  Note: In such cases, make sure that the request header DOES not include the following:

  Cookie:NITRO_AUTH_TOKEN=<tokenvalue>

`*Note:` *

Mandatory parameters are marked in redand placeholder content is marked in <green>.

add

URL:http://<Citrix-ADC-IP-address(NSIP)/nitro/v1/config/responderglobal_responderpolicy_binding

HTTP Method:PUT

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Content-Type:application/json

Request Payload:

```
{

"responderglobal_responderpolicy_binding":{

<b>"policyname":<String_value>,

</b><b>"priority":<Double_value>,

</b>"gotopriorityexpression":<String_value>,

"type":<String_value>,

"invoke":<Boolean_value>,

"labeltype":<String_value>,

"labelname":<String_value>

}}
```

Response:

HTTP Status Code on Success: 201 Created HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error

delete

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/responderglobal_responderpolicy_binding

Query-parameters:

args

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/responderglobal_responderpolicy_binding?**args=policyname:<String_value>,type:<String_value>,priority:<Double_value>**

HTTP Method:DELETE

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue>

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error

get

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/responderglobal_responderpolicy_binding

Query-parameters:

filter

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/responderglobal_responderpolicy_binding?**filter=property-name1:property-value1,property-name2:property-value2**

Use this query-parameter to get the filtered set of responderglobal_responderpolicy_binding resources configured on Citrix ADC. Filtering can be done on any of the properties of the resource.

pagination

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/responderglobal_responderpolicy_binding?**pagesize=#no;pageno=#no**

Use this query-parameter to get the responderglobal_responderpolicy_binding resources in chunks.

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "responderglobal_responderpolicy_binding": [ {

"priority":<Double_value>,

"globalbindtype":<String_value>,

"policyname":<String_value>,

"labelname":<String_value>,

"gotopriorityexpression":<String_value>,

"invoke":<Boolean_value>,

"type":<String_value>,

"labeltype":<String_value>,

"numpol":<Double_value>,

"flowtype":<Double_value>

}]}
```

get (all)

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/responderglobal_responderpolicy_binding

Query-parameters:

bulkbindings

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/responderglobal_responderpolicy_binding?**bulkbindings=yes**

NITRO allows you to fetch bindings in bulk.

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "responderglobal_responderpolicy_binding": [ {

"priority":<Double_value>,

"globalbindtype":<String_value>,

"policyname":<String_value>,

"labelname":<String_value>,

"gotopriorityexpression":<String_value>,

"invoke":<Boolean_value>,
```

```
"type":<String_value>,

"labeltype":<String_value>,

"numpol":<Double_value>,

"flowtype":<Double_value>

}]}
```

count

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/responderglobal_responderpolicy_binding?**count=yes**

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "responderglobal_responderpolicy_binding": [ { "__count": "#no"} ] }
```

responderhtmlpage

Configuration for Responder HTML page resource.

Properties

(click to see Operations)

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
src	<String>	Read-write	Local path to and name of, or URL \ (protocol, host, path, and file name\) for, the file in which to store the imported HTML page. NOTE: The import fails if the object to be imported is on an HTTPS server that requires client certificate authentication for access. Minimum length = 1 Maximum length = 2047
name	<String>	Read-write	Name to assign to the HTML page object on the Citrix ADC. Minimum length = 1 Maximum length = 31
comment	<String>	Read-write	Any comments to preserve information about the HTML page object. Maximum length = 128

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
overwrite	<Boolean>	Read-write	Overwrites the existing file.
response	<String>	Read-only	.

Operations

(click to see Properties)

IMPORT| DELETE| CHANGE| GET (ALL)| GET

Some options that you can use for each operations:

- **Getting warnings in response:**NITRO allows you to get warnings in an operation by specifying the "warning" query parameter as "yes". For example, to get warnings while connecting to the Citrix ADC appliance, the URL is as follows:

  http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/login?warning=yes

  If any, the warnings are displayed in the response payload with the HTTP code "209 X-NITRO-WARNING".

- **Authenticated access for individual NITRO operations:**NITRO allows you to logon to the Citrix ADC appliance to perform individual operations. You can use this option instead of creating a NITRO session (using the login object) and then using that session to perform all operations,

  To do this, you must specify the username and password in the request header of the NITRO request as follows:

  X-NITRO-USER:<username>

  X-NITRO-PASS:<password>

  Note: In such cases, make sure that the request header DOES not include the following:

  Cookie:NITRO_AUTH_TOKEN=<tokenvalue>

`*Note:`*

Mandatory parameters are marked in redand placeholder content is marked in <green>.

Import

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/responderhtmlpage?**action=Import**

HTTP Method:POST

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Content-Type:application/json

Request Payload:

```
{"responderhtmlpage":{

"src":<String_value>,

<b>"name":<String_value>,

</b>"comment":<String_value>,

"overwrite":<Boolean_value>

}}
```

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error

delete

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/responderhtmlpage/name_value<String>

HTTP Method:DELETE

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue>

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error

change

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/responderhtmlpage?**action=update**

HTTP Method:POST

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Content-Type:application/json

Request Payload:

```
{"responderhtmlpage":{

<b>"name":<String_value>

</b>}}
```

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error

get (all)

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/responderhtmlpage

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "responderhtmlpage": [ {

"name":<String_value>,

"response":<String_value>

}]}
```

get

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/responderhtmlpage/name_value<String>

Query-parameters:

attrs

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/responderhtmlpage/name_value<String>?**attrs=property-name1,property-name2**

Use this query parameter to specify the resource details that you want to retrieve.

view

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/responderhtmlpage/name_value<String>?**view=summary**

Note: By default, the retrieved results are displayed in detail view (?view=detail).

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "responderhtmlpage": [ {

"name":<String_value>,

"response":<String_value>

}]}
```

responderparam

Configuration for responser parameter resource.

Properties

(click to see Operations)

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
undefaction	<String>	Read-write	Action to perform when policy evaluation creates an UNDEF condition. Available settings function as follows: * NOOP - Send the request to the protected server.

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
			* RESET - Reset the request and notify the user's browser, so that the user can resend the request. * DROP - Drop the request without sending a response to the user. Default value: "NOOP"

Operations

(click to see Properties)

UPDATE| UNSET| GET (ALL)

Some options that you can use for each operations:

- **Getting warnings in response:**NITRO allows you to get warnings in an operation by specifying the "warning" query parameter as "yes". For example, to get warnings while connecting to the Citrix ADC appliance, the URL is as follows:

  http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/login?warning=yes

  If any, the warnings are displayed in the response payload with the HTTP code "209 X-NITRO-WARNING".

- **Authenticated access for individual NITRO operations:**NITRO allows you to logon to the Citrix ADC appliance to perform individual operations. You can use this option instead of creating a NITRO session (using the login object) and then using that session to perform all operations,

  To do this, you must specify the username and password in the request header of the NITRO request as follows:

  X-NITRO-USER:<username>

X-NITRO-PASS:<password>

Note: In such cases, make sure that the request header DOES not include the following:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue>

`*Note:`*

Mandatory parameters are marked in redand placeholder content is marked in <green>.

update

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/responderparam

HTTP Method:PUT

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Content-Type:application/json

Request Payload:

```
{"responderparam":{

"undefaction":<String_value>

}}
```

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error

unset

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/responderparam?**action=unset**

HTTP Method:POST

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Content-Type:application/json

Request Payload:

```
{"responderparam":{

"undefaction":true

}}
```

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error

get (all)

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/responderparam

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "responderparam": [ {

"undefaction":<String_value>

}]}
```

responderpolicy

Configuration for responder policy resource.

Properties

(click to see Operations)

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
name	<String>	Read-write	Name for the responder policy. Must begin with a letter, number, or the underscore character (_), and must contain only letters, numbers, and the hyphen (-), period (.) pound (#), space ( ), at (@), equals (=), colon (:), and underscore characters. Can be changed after the responder policy is added. The following requirement applies only to the Citrix ADC CLI: If the name includes one or more spaces, enclose the name in double or single quotation

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
			marks (for example, "my responder policy" or 'my responder policy').
rule	<String>	Read-write	Expression that the policy uses to determine whether to respond to the specified request.
action	<String>	Read-write	Name of the responder action to perform if the request matches this responder policy. There are also some built-in actions which can be used. These are: * NOOP - Send the request to the protected server instead of responding to it. * RESET - Reset the client connection by closing it. The client program, such as a browser, will handle this and may inform the user. The client may then resend the request if desired.

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
			* DROP - Drop the request without sending a response to the user.
undefaction	<String>	Read-write	Action to perform if the result of policy evaluation is undefined (UNDEF). An UNDEF event indicates an internal error condition. Only the above built-in actions can be used.
comment	<String>	Read-write	Any type of information about this responder policy.
logaction	<String>	Read-write	Name of the messagelog action to use for requests that match this policy.
appflowaction	<String>	Read-write	AppFlow action to invoke for requests that match this policy.
newname	<String>	Read-write	New name for the responder policy. Must begin with a letter, number, or the underscore character (_), and must contain only letters, numbers, and the hyphen (-), period (.) hash (#), space ( ), at (@), equals (=), colon (:), and

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
			underscore characters. The following requirement applies only to the Citrix ADC CLI: If the name includes one or more spaces, enclose the name in double or single quotation marks (for example, "my responder policy" or 'my responder policy'). Minimum length = 1
hits	<Double>	Read-only	Number of hits.
undefhits	<Double>	Read-only	Number of policy UNDEF hits.
__count	<Double>	Read-only	count parameter

Operations

(click to see Properties)

ADD| DELETE| UPDATE| UNSET| RENAME| GET (ALL)| GET| COUNT

Some options that you can use for each operations:

- **Getting warnings in response:**NITRO allows you to get warnings in an operation by specifying the "warning" query parameter as "yes". For example, to get warnings while connecting to the Citrix ADC appliance, the URL is as follows:

  http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/login?warning=yes

If any, the warnings are displayed in the response payload with the HTTP code "209 X-NITRO-WARNING".

- **Authenticated access for individual NITRO operations:**NITRO allows you to logon to the Citrix ADC appliance to perform individual operations. You can use this option instead of creating a NITRO session (using the login object) and then using that session to perform all operations,

  To do this, you must specify the username and password in the request header of the NITRO request as follows:

  X-NITRO-USER:<username>

  X-NITRO-PASS:<password>

  Note: In such cases, make sure that the request header DOES not include the following:

  Cookie:NITRO_AUTH_TOKEN=<tokenvalue>

`*Note:`*

Mandatory parameters are marked in redand placeholder content is marked in <green>.

add

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/responderpolicy

HTTP Method:POST

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Content-Type:application/json

Request Payload:

```
{"responderpolicy":{

<b>"name":<String_value>,

</b><b>"rule":<String_value>,

</b><b>"action":<String_value>,
```

```
</b>"undefaction":<String_value>,

"comment":<String_value>,

"logaction":<String_value>,

"appflowaction":<String_value>

}}
```

Response:

HTTP Status Code on Success: 201 Created HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error

delete

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/responderpolicy/name_value<String>

HTTP Method:DELETE

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue>

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error

update

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/responderpolicy

HTTP Method:PUT

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Content-Type:application/json

Request Payload:

```
{"responderpolicy":{

<b>"name":<String_value>,

</b>"rule":<String_value>,

"action":<String_value>,

"undefaction":<String_value>,

"comment":<String_value>,

"logaction":<String_value>,

"appflowaction":<String_value>

}}
```

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error

unset

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/responderpolicy?**action=unset**

HTTP Method:POST

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Content-Type:application/json

Request Payload:

```
{"responderpolicy":{
```

```
<b>"name":<String_value>,

</b>"undefaction":true,

"comment":true,

"logaction":true,

"appflowaction":true

}}
```

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error

rename

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/responderpolicy?**action=rename**

HTTP Method:POST

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Content-Type:application/json

Request Payload:

```
{"responderpolicy":{

<b>"name":<String_value>,

</b><b>"newname":<String_value>

</b>}}
```

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error

get (all)

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/responderpolicy

Query-parameters:

attrs

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/responderpolicy?**attrs=property-name1,property-name2**

Use this query parameter to specify the resource details that you want to retrieve.

filter

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/responderpolicy?**filter=property-name1:property-val1,property-name2:property-val2**

Use this query-parameter to get the filtered set of responderpolicy resources configured on Citrix ADC. Filtering can be done on any of the properties of the resource.

view

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/responderpolicy?**view=summary**

Note: By default, the retrieved results are displayed in detail view (?view=detail).

pagination

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/responderpolicy?**pagesize=#no;pageno=#no**

Use this query-parameter to get the responderpolicy resources in chunks.

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "responderpolicy": [ {

"name":<String_value>,

"rule":<String_value>,

"action":<String_value>,

"undefaction":<String_value>,

"hits":<Double_value>,

"undefhits":<Double_value>,

"comment":<String_value>,

"logaction":<String_value>,

"appflowaction":<String_value>

}]}
```

get

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/responderpolicy/name_value<String>

Query-parameters:

attrs

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/responderpolicy/name_value<String>?**attrs=property-name1,property-name2**

Use this query parameter to specify the resource details that you want to retrieve.

view

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/responderpolicy/name_value<String>?**view=summary**

Note: By default, the retrieved results are displayed in detail view (?view=detail).

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "responderpolicy": [ {

"name":<String_value>,

"rule":<String_value>,

"action":<String_value>,

"undefaction":<String_value>,

"hits":<Double_value>,

"undefhits":<Double_value>,

"comment":<String_value>,

"logaction":<String_value>,
```

```
"appflowaction":<String_value>

}]}
```

count

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/responderpolicy?**count=yes**

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "responderpolicy": [ { "__count": "#no"} ] }
```

responderpolicylabel

Configuration for responder policy label resource.

Properties

(click to see Operations)

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
labelname	<String>	Read-write	Name for the responder policy label. Must begin

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
			with a letter, number, or the underscore character (_), and must contain only letters, numbers, and the hyphen (-), period (.) hash (#), space ( ), at (@), equals (=), colon (:), and underscore characters. Cannot be changed after the responder policy label is added. The following requirement applies only to the Citrix ADC CLI: If the name includes one or more spaces, enclose the name in double or single quotation marks (for example, "my responder policy label" or my responder policy label').
policylabeltype	<String>	Read-write	Type of responses sent by the policies bound to this policy label. Types are: * HTTP - HTTP responses.

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
			* OTHERTCP - NON-HTTP TCP responses. * SIP_UDP - SIP responses. * RADIUS - RADIUS responses. * MYSQL - SQL responses in MySQL format. * MSSQL - SQL responses in Microsoft SQL format. * NAT - NAT response. Default value: HTTP Possible values = HTTP, OTHERTCP, SIP_UDP, SIP_TCP, MYSQL, MSSQL, NAT, DIAMETER, RADIUS, DNS
comment	<String>	Read-write	Any comments to preserve information about this responder policy label.
newname	<String>	Read-write	New name for the responder policy label. Must begin with a letter, number,

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
			or the underscore character (_), and must contain only letters, numbers, and the hyphen (-), period (.) hash (#), space ( ), at (@), equals (=), colon (:), and underscore characters. Minimum length = 1
numpol	<Double>	Read-only	number of polices bound to label.
hits	<Double>	Read-only	Number of times policy label was invoked.
priority	<Double>	Read-only	Specifies the priority of the policy.
gotopriorityexpression	<String>	Read-only	Expression specifying the priority of the next policy which will get evaluated if the current policy rule evaluates to TRUE.
labeltype	<String>	Read-only	Type of policy label to invoke. Available settings function as follows: * vserver - Invoke an unnamed policy label associated with a virtual server.

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
			* policylabel - Invoke a user-defined policy label. Possible values = vserver, policylabel
invoke_labelname	<String>	Read-only	* If labelType is policylabel, name of the policy label to invoke. * If labelType is reqvserver or resvserver, name of the virtual server.
__count	<Double>	Read-only	count parameter

Operations

(click to see Properties)

ADD| DELETE| RENAME| GET (ALL)| GET| COUNT

Some options that you can use for each operations:

- **Getting warnings in response:**NITRO allows you to get warnings in an operation by specifying the "warning" query parameter as "yes". For example, to get warnings while connecting to the Citrix ADC appliance, the URL is as follows:

  http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/login?warning=yes

  If any, the warnings are displayed in the response payload with the HTTP code "209 X-NITRO-WARNING".
- **Authenticated access for individual NITRO operations:**NITRO allows you to logon to the Citrix ADC appliance to perform individual operations. You can use this option instead of creating a NITRO session (using the login object) and then using that session to perform all operations,

To do this, you must specify the username and password in the request header of the NITRO request as follows:

X-NITRO-USER:<username>

X-NITRO-PASS:<password>

Note: In such cases, make sure that the request header DOES not include the following:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue>

`*Note:`*

Mandatory parameters are marked in redand placeholder content is marked in <green>.

add

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/responderpolicylabel

HTTP Method:POST

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Content-Type:application/json

Request Payload:

```
{"responderpolicylabel":{

<b>"labelname":<String_value>,

</b>"policylabeltype":<String_value>,

"comment":<String_value>

}}
```

Response:

HTTP Status Code on Success: 201 Created HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error

delete

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/responderpolicylabel/labelname_value<String>

HTTP Method:DELETE

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue>

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error

rename

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/responderpolicylabel?**action=rename**

HTTP Method:POST

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Content-Type:application/json

Request Payload:

```
{"responderpolicylabel":{

<b>"labelname":<String_value>,

</b><b>"newname":<String_value>

</b>}}
```

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error

get (all)

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/responderpolicylabel

Query-parameters:

attrs

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/responderpolicylabel?**attrs=property-name1,property-name2**

Use this query parameter to specify the resource details that you want to retrieve.

filter

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/responderpolicylabel?**filter=property-name1:property-val1,property-name2:property-val2**

Use this query-parameter to get the filtered set of responderpolicylabel resources configured on Citrix ADC. Filtering can be done on any of the properties of the resource.

view

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/responderpolicylabel?**view=summary**

Note: By default, the retrieved results are displayed in detail view (?view=detail).

pagination

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/responderpolicylabel?**pagesize=#no;pageno=#no**

Use this query-parameter to get the responderpolicylabel resources in chunks.

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "responderpolicylabel": [ {

"labelname":<String_value>,

"policylabeltype":<String_value>,

"numpol":<Double_value>,

"hits":<Double_value>,

"priority":<Double_value>,

"gotopriorityexpression":<String_value>,

"labeltype":<String_value>,

"invoke_labelname":<String_value>,

"comment":<String_value>

}]}
```

get

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/responderpolicylabel/labelname_value<String>

Query-parameters:

attrs

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/responderpolicylabel/labelname_value<String>?**attrs=property-name1,property-name2**

Use this query parameter to specify the resource details that you want to retrieve.

view

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/responderpolicylabel/labelname_value<String>?**view=summary**

Note: By default, the retrieved results are displayed in detail view (?view=detail).

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "responderpolicylabel": [ {

"labelname":<String_value>,

"policylabeltype":<String_value>,

"numpol":<Double_value>,

"hits":<Double_value>,

"priority":<Double_value>,

"gotopriorityexpression":<String_value>,

"labeltype":<String_value>,

"invoke_labelname":<String_value>,

"comment":<String_value>

}]}
```

count

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/responderpolicylabel?**count=yes**

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "responderpolicylabel": [ { "__count": "#no"} ] }
```

responderpolicylabel_binding

Binding object which returns the resources bound to responderpolicylabel.

Properties

(click to see Operations)

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
labelname	<String>	Read-write	Name of the responder policy label.
responderpolicylabel_policybinding_binding	<responderpolicylabel_policybinding_binding[]>	Read-only	policybinding that can be bound to responderpolicylabel.

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
responderpolicylabel_responderpolicy_binding	<responderpolicylabel_responderpolicy_binding[]>	Read-only	responderpolicy that can be bound to responderpolicylabel.

Operations

(click to see Properties)

GET| GET (ALL)

Some options that you can use for each operations:

- **Getting warnings in response:**NITRO allows you to get warnings in an operation by specifying the "warning" query parameter as "yes". For example, to get warnings while connecting to the Citrix ADC appliance, the URL is as follows:

  http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/login?warning=yes

  If any, the warnings are displayed in the response payload with the HTTP code "209 X-NITRO-WARNING".

- **Authenticated access for individual NITRO operations:**NITRO allows you to logon to the Citrix ADC appliance to perform individual operations. You can use this option instead of creating a NITRO session (using the login object) and then using that session to perform all operations,

  To do this, you must specify the username and password in the request header of the NITRO request as follows:

  X-NITRO-USER:<username>

  X-NITRO-PASS:<password>

  Note: In such cases, make sure that the request header DOES not include the following:

  Cookie:NITRO_AUTH_TOKEN=<tokenvalue>

`*Note:` *

Mandatory parameters are marked in redand placeholder content is marked in <green>.

get

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/responderpolicylabel_binding/labelname_value<String>

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "responderpolicylabel_binding": [ {

"labelname":<String_value>,

"responderpolicylabel_responderpolicy_binding":<responderpolicylabel_respo
nderpolicy_binding[]_value>,

"responderpolicylabel_policybinding_binding":<responderpolicylabel_policyb
inding_binding[]_value>

}]}
```

get (all)

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/responderpolicylabel_binding

Query-parameters:

bulkbindings

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/responderpolicylabel_binding?**bulkbindings=yes**

NITRO allows you to fetch bindings in bulk.

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "responderpolicylabel_binding": [ {

"labelname":<String_value>,

"responderpolicylabel_responderpolicy_binding":<responderpolicylabel_respo
nderpolicy_binding[]_value>,

"responderpolicylabel_policybinding_binding":<responderpolicylabel_policyb
inding_binding[]_value>

}]}
```

responderpolicylabel_policybinding_binding

Binding object showing the policybinding that can be bound to responderpolicylabel.

Properties

(click to see Operations)

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
priority	<Double>	Read-write	Specifies the priority of the policy.
policyname	<String>	Read-write	Name of the responder policy.
labelname	<String>	Read-write	Name of the responder policy label to which to bind the policy.
invoke_labelname	<String>	Read-write	* If labelType is policylabel, name of the policy label to invoke. * If labelType is reqvserver or resvserver, name of the virtual server.
gotopriorityexpression	<String>	Read-write	Expression specifying the priority of the next policy which will get evaluated if the current policy rule evaluates to TRUE.
invoke	<Boolean>	Read-write	If the current policy evaluates to TRUE, terminate evaluation of policies bound to the current policy label and evaluate the specified policy label.
labeltype	<String>	Read-write	Type of policy label to invoke. Available settings function as follows: * vserver - Invoke an unnamed

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
			policy label associated with a virtual server. * policylabel - Invoke a user-defined policy label. Possible values = vserver, policylabel
__count	<Double>	Read-write	count parameter

Operations

(click to see Properties)

GET| GET (ALL)| COUNT

Some options that you can use for each operations:

- **Getting warnings in response:**NITRO allows you to get warnings in an operation by specifying the "warning" query parameter as "yes". For example, to get warnings while connecting to the Citrix ADC appliance, the URL is as follows:

  http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/login?warning=yes

  If any, the warnings are displayed in the response payload with the HTTP code "209 X-NITRO-WARNING".

- **Authenticated access for individual NITRO operations:**NITRO allows you to logon to the Citrix ADC appliance to perform individual operations. You can use this option instead of creating a NITRO session (using the login object) and then using that session to perform all operations,

  To do this, you must specify the username and password in the request header of the NITRO request as follows:

  X-NITRO-USER:<username>

  X-NITRO-PASS:<password>

  Note: In such cases, make sure that the request header DOES not include the following:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue>

`*Note:`*

Mandatory parameters are marked in redand placeholder content is marked in <green>.

get

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/responderpolicylabel_policybinding_binding/labelname_value<String>

Query-parameters:

filter

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/responderpolicylabel_policybinding_binding/labelname_value<String>?**filter=property-name1:property-value1,property-name2:property-value2**

Use this query-parameter to get the filtered set of responderpolicylabel_policybinding_binding resources configured on Citrix ADC. Filtering can be done on any of the properties of the resource.

pagination

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/responderpolicylabel_policybinding_binding/labelname_value<String>?**pagesize=#no;pageno=#no**

Use this query-parameter to get the responderpolicylabel_policybinding_binding resources in chunks.

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "responderpolicylabel_policybinding_binding": [ {

"priority":<Double_value>,

"policyname":<String_value>,

"labelname":<String_value>,

"invoke_labelname":<String_value>,

"gotopriorityexpression":<String_value>,

"invoke":<Boolean_value>,

"labeltype":<String_value>

}]}
```

get (all)

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/responderpolicylabel_policybinding_binding

Query-parameters:

bulkbindings

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/responderpolicylabel_policybinding_binding?**bulkbindings=yes**

NITRO allows you to fetch bindings in bulk.

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "responderpolicylabel_policybinding_binding": [ {

"priority":<Double_value>,

"policyname":<String_value>,

"labelname":<String_value>,

"invoke_labelname":<String_value>,

"gotopriorityexpression":<String_value>,

"invoke":<Boolean_value>,

"labeltype":<String_value>

}]}
```

count

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/responderpolicylabel_policybinding_binding/labelname_value<String>?**count=yes**

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{"responderpolicylabel_policybinding_binding": [ { "__count": "#no"} ] }
```

responderpolicylabel_responderpolicy_binding

Binding object showing the responderpolicy that can be bound to responderpolicylabel.

Properties

(click to see Operations)

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
priority	<Double>	Read-write	Specifies the priority of the policy.
policyname	<String>	Read-write	Name of the responder policy.
labelname	<String>	Read-write	Name of the responder policy label to which to bind the policy.
invoke_labelname	<String>	Read-write	* If labelType is policylabel, name of the policy label to invoke. * If labelType is reqvserver or resvserver, name of the virtual server.

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
gotopriorityexpression	<String>	Read-write	Expression specifying the priority of the next policy which will get evaluated if the current policy rule evaluates to TRUE.
invoke	<Boolean>	Read-write	If the current policy evaluates to TRUE, terminate evaluation of policies bound to the current policy label and evaluate the specified policy label.
labeltype	<String>	Read-write	Type of policy label to invoke. Available settings function as follows: * vserver - Invoke an unnamed policy label associated with a virtual server. * policylabel - Invoke a user-defined policy label. Possible values = vserver, policylabel
__count	<Double>	Read-write	count parameter

Operations

(click to see Properties)

ADD:| DELETE:| GET| GET (ALL)| COUNT

Some options that you can use for each operations:

- **Getting warnings in response:**NITRO allows you to get warnings in an operation by specifying the "warning" query parameter as "yes". For example, to get warnings while connecting to the Citrix ADC appliance, the URL is as follows:

  http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/login?warning=yes

  If any, the warnings are displayed in the response payload with the HTTP code "209 X-NITRO-WARNING".

- **Authenticated access for individual NITRO operations:**NITRO allows you to logon to the Citrix ADC appliance to perform individual operations. You can use this option instead of creating a NITRO session (using the login object) and then using that session to perform all operations,

  To do this, you must specify the username and password in the request header of the NITRO request as follows:

  X-NITRO-USER:<username>

  X-NITRO-PASS:<password>

  Note: In such cases, make sure that the request header DOES not include the following:

  Cookie:NITRO_AUTH_TOKEN=<tokenvalue>

`*Note:`*

Mandatory parameters are marked in redand placeholder content is marked in <green>.

add

URL:http://<Citrix-ADC-IP-address(NSIP)/nitro/v1/config/responderpolicylabel_responderpolicy_binding

HTTP Method:PUT

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Content-Type:application/json

Request Payload:

```
{
```

```
"responderpolicylabel_responderpolicy_binding":{

<b>"labelname":<String_value>,

</b><b>"policyname":<String_value>,

</b><b>"priority":<Double_value>,

</b>"gotopriorityexpression":<String_value>,

"invoke":<Boolean_value>,

"labeltype":<String_value>,

"invoke_labelname":<String_value>

}}
```

Response:

HTTP Status Code on Success: 201 Created HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error

delete

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/responderpolicylabel_responderpolicy_binding/labelname_value<String>

Query-parameters:

args

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/responderpolicylabel_responderpolicy_binding/labelname_value<String>?**args=policyname:<String_value>,priority:<Double_value>**

HTTP Method:DELETE

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue>

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error

get

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/responderpolicylabel_responderpolicy_binding/labelname_value<String>

Query-parameters:

filter

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/responderpolicylabel_responderpolicy_binding/labelname_value<String>?**filter=property-name1:property-value1,property-name2:property-value2**

Use this query-parameter to get the filtered set of responderpolicylabel_responderpolicy_binding resources configured on Citrix ADC. Filtering can be done on any of the properties of the resource.

pagination

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/responderpolicylabel_responderpolicy_binding/labelname_value<String>?**pagesize=#no;pageno=#no**

Use this query-parameter to get the responderpolicylabel_responderpolicy_binding resources in chunks.

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "responderpolicylabel_responderpolicy_binding": [ {

"priority":<Double_value>,

"policyname":<String_value>,

"labelname":<String_value>,

"invoke_labelname":<String_value>,

"gotopriorityexpression":<String_value>,

"invoke":<Boolean_value>,

"labeltype":<String_value>

}]}
```

get (all)

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/responderpolicylabel_responderpolicy_binding

Query-parameters:

bulkbindings

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/responderpolicylabel_responderpolicy_binding?**bulkbindings=yes**

NITRO allows you to fetch bindings in bulk.

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "responderpolicylabel_responderpolicy_binding": [ {

"priority":<Double_value>,

"policyname":<String_value>,

"labelname":<String_value>,

"invoke_labelname":<String_value>,

"gotopriorityexpression":<String_value>,

"invoke":<Boolean_value>,

"labeltype":<String_value>

}]}
```

count

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/responderpolicylabel_responderpolicy_binding/labelname_value<String>?**count=yes**

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{"responderpolicylabel_responderpolicy_binding": [ { "__count": "#no"} ] }
```

responderpolicy_binding

Binding object which returns the resources bound to responderpolicy.

Properties

(click to see Operations)

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
name	<String>	Read-write	Name of the responder policy for which to display settings.
responderpolicy_responderpolicylabel_binding	<responderpolicy_responderpolicylabel_binding[]>	Read-only	responderpolicylabel that can be bound to responderpolicy.
responderpolicy_csvserver_binding	<responderpolicy_csvserver_binding[]>	Read-only	csvserver that can be bound to responderpolicy.
responderpolicy_responderglobal_binding	<responderpolicy_responderglobal_binding[]>	Read-only	responderglobal that can be bound to responderpolicy.
responderpolicy_lbvserver_binding	<responderpolicy_lbvserver_binding[]>	Read-only	lbvserver that can be bound to responderpolicy.

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
responderpolicy_crvserver_binding	<responderpolicy_crvserver_binding[]>	Read-only	crvserver that can be bound to responderpolicy.

Operations

(click to see Properties)

GET| GET (ALL)

Some options that you can use for each operations:

- **Getting warnings in response:**NITRO allows you to get warnings in an operation by specifying the "warning" query parameter as "yes". For example, to get warnings while connecting to the Citrix ADC appliance, the URL is as follows:

  http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/login?warning=yes

  If any, the warnings are displayed in the response payload with the HTTP code "209 X-NITRO-WARNING".

- **Authenticated access for individual NITRO operations:**NITRO allows you to logon to the Citrix ADC appliance to perform individual operations. You can use this option instead of creating a NITRO session (using the login object) and then using that session to perform all operations,

  To do this, you must specify the username and password in the request header of the NITRO request as follows:

  X-NITRO-USER:<username>

  X-NITRO-PASS:<password>

  Note: In such cases, make sure that the request header DOES not include the following:

  Cookie:NITRO_AUTH_TOKEN=<tokenvalue>

`*Note:`*

Mandatory parameters are marked in redand placeholder content is marked in <green>.

get

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/responderpolicy_binding/name_value<String>

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "responderpolicy_binding": [ {

"name":<String_value>,

"responderpolicy_crvserver_binding":<responderpolicy_crvserver_binding[]_value>,

"responderpolicy_responderglobal_binding":<responderpolicy_responderglobal_binding[]_value>,

"responderpolicy_csvserver_binding":<responderpolicy_csvserver_binding[]_value>,

"responderpolicy_lbvserver_binding":<responderpolicy_lbvserver_binding[]_value>,

"responderpolicy_responderpolicylabel_binding":<responderpolicy_responderpolicylabel_binding[]_value>

}]}
```

get (all)

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/responderpolicy_binding

Query-parameters:

bulkbindings

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/responderpolicy_binding?**bulkbindings=yes**

NITRO allows you to fetch bindings in bulk.

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "responderpolicy_binding": [ {

"name":<String_value>,

"responderpolicy_crvserver_binding":<responderpolicy_crvserver_binding[]_v
alue>,

"responderpolicy_responderglobal_binding":<responderpolicy_responderglobal
_binding[]_value>,

"responderpolicy_csvserver_binding":<responderpolicy_csvserver_binding[]_v
alue>,
```

```
"responderpolicy_lbvserver_binding":<responderpolicy_lbvserver_binding[]_value>,

"responderpolicy_responderpolicylabel_binding":<responderpolicy_responderpolicylabel_binding[]_value>

}]}
```

responderpolicy_crvserver_binding

Binding object showing the crvserver that can be bound to responderpolicy.

Properties

(click to see Operations)

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
boundto	<String>	Read-write	Location where policy is bound.
name	<String>	Read-write	Name of the responder policy for which to display settings.
priority	<Double>	Read-only	Specifies the priority of the policy.
labelname	<String>	Read-only	Name of the label to invoke if the current policy rule evaluates to TRUE.
gotopriorityexpression	<String>	Read-only	Expression specifying the priority of the next policy which will get evaluated if the current policy rule evaluates to TRUE.

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
labeltype	<String>	Read-only	Type of policy label invocation. Possible values = reqvserver, resvserver, policylabel
activepolicy	<Integer>	Read-only	Indicates whether policy is bound or not.
__count	<Double>	Read-only	count parameter

Operations

(click to see Properties)

GET| GET (ALL)| COUNT

Some options that you can use for each operations:

- **Getting warnings in response:**NITRO allows you to get warnings in an operation by specifying the "warning" query parameter as "yes". For example, to get warnings while connecting to the Citrix ADC appliance, the URL is as follows:

  http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/login?warning=yes

  If any, the warnings are displayed in the response payload with the HTTP code "209 X-NITRO-WARNING".

- **Authenticated access for individual NITRO operations:**NITRO allows you to logon to the Citrix ADC appliance to perform individual operations. You can use this option instead of creating a NITRO session (using the login object) and then using that session to perform all operations,

  To do this, you must specify the username and password in the request header of the NITRO request as follows:

  X-NITRO-USER:<username>

  X-NITRO-PASS:<password>

Note: In such cases, make sure that the request header DOES not include the following:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue>

`*Note:`*

Mandatory parameters are marked in redand placeholder content is marked in <green>.

get

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/responderpolicy_crvserver_binding/name_value<String>

Query-parameters:

filter

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/responderpolicy_crvserver_binding/name_value<String>?**filter=property-name1:property-value1,property-name2:property-value2**

Use this query-parameter to get the filtered set of responderpolicy_crvserver_binding resources configured on Citrix ADC. Filtering can be done on any of the properties of the resource.

pagination

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/responderpolicy_crvserver_binding/name_value<String>?**pagesize=#no;pageno=#no**

Use this query-parameter to get the responderpolicy_crvserver_binding resources in chunks.

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "responderpolicy_crvserver_binding": [ {

"boundto":<String_value>,

"name":<String_value>,

"priority":<Double_value>,

"labelname":<String_value>,

"gotopriorityexpression":<String_value>,

"labeltype":<String_value>,

"activepolicy":<Integer_value>

}]}
```

get (all)

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/responderpolicy_crvserver_binding

Query-parameters:

bulkbindings

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/responderpolicy_crvserver_binding?**bulkbindings=yes**

NITRO allows you to fetch bindings in bulk.

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "responderpolicy_crvserver_binding": [ {

"boundto":<String_value>,

"name":<String_value>,

"priority":<Double_value>,

"labelname":<String_value>,

"gotopriorityexpression":<String_value>,

"labeltype":<String_value>,

"activepolicy":<Integer_value>

}]}
```

count

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/responderpolicy_crvserver_binding/name_value<String>?**count=yes**

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{"responderpolicy_crvserver_binding": [ { "__count": "#no"} ] }
```

responderpolicy_csvserver_binding

Binding object showing the csvserver that can be bound to responderpolicy.

Properties

(click to see Operations)

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
boundto	<String>	Read-write	Location where policy is bound.
name	<String>	Read-write	Name of the responder policy for which to display settings.
priority	<Double>	Read-only	Specifies the priority of the policy.
labelname	<String>	Read-only	Name of the label to invoke if the current policy rule evaluates to TRUE.
gotopriorityexpression	<String>	Read-only	Expression specifying the priority of the next policy which will get

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
			evaluated if the current policy rule evaluates to TRUE.
labeltype	<String>	Read-only	Type of policy label invocation. Possible values = reqvserver, resvserver, policylabel
activepolicy	<Integer>	Read-only	Indicates whether policy is bound or not.
__count	<Double>	Read-only	count parameter

Operations

(click to see Properties)

GET| GET (ALL)| COUNT

Some options that you can use for each operations:

- **Getting warnings in response:**NITRO allows you to get warnings in an operation by specifying the "warning" query parameter as "yes". For example, to get warnings while connecting to the Citrix ADC appliance, the URL is as follows:

  http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/login?warning=yes

  If any, the warnings are displayed in the response payload with the HTTP code "209 X-NITRO-WARNING".

- **Authenticated access for individual NITRO operations:**NITRO allows you to logon to the Citrix ADC appliance to perform individual operations. You can use this option instead of creating a NITRO session (using the login object) and then using that session to perform all operations,

  To do this, you must specify the username and password in the request header of the NITRO request as follows:

X-NITRO-USER:<username>

X-NITRO-PASS:<password>

Note: In such cases, make sure that the request header DOES not include the following:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue>

`*Note:`*

Mandatory parameters are marked in redand placeholder content is marked in <green>.

get

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/responderpolicy_csvserver_binding/name_value<String>

Query-parameters:

filter

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/responderpolicy_csvserver_binding/name_value<String>?**filter=property-name1:property-value1,property-name2:property-value2**

Use this query-parameter to get the filtered set of responderpolicy_csvserver_binding resources configured on Citrix ADC. Filtering can be done on any of the properties of the resource.

pagination

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/responderpolicy_csvserver_binding/name_value<String>?**pagesize=#no;pageno=#no**

Use this query-parameter to get the responderpolicy_csvserver_binding resources in chunks.

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "responderpolicy_csvserver_binding": [ {

"boundto":<String_value>,

"name":<String_value>,

"priority":<Double_value>,

"labelname":<String_value>,

"gotopriorityexpression":<String_value>,

"labeltype":<String_value>,

"activepolicy":<Integer_value>

}]}
```

get (all)

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/responderpolicy_csvserver_binding

Query-parameters:

bulkbindings

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/responderpolicy_csvserver_binding?**bulkbindings=yes**

NITRO allows you to fetch bindings in bulk.

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "responderpolicy_csvserver_binding": [ {

"boundto":<String_value>,

"name":<String_value>,

"priority":<Double_value>,

"labelname":<String_value>,

"gotopriorityexpression":<String_value>,

"labeltype":<String_value>,

"activepolicy":<Integer_value>

}]}
```

count

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/responderpolicy_csvserver_binding/name_value<String>?**count=yes**

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{"responderpolicy_csvserver_binding": [ { "__count": "#no"} ] }
```

responderpolicy_lbvserver_binding

Binding object showing the lbvserver that can be bound to responderpolicy.

Properties

(click to see Operations)

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
boundto	<String>	Read-write	Location where policy is bound.
name	<String>	Read-write	Name of the responder policy for which to display settings.
priority	<Double>	Read-only	Specifies the priority of the policy.
labelname	<String>	Read-only	Name of the label to invoke if the current policy rule evaluates to TRUE.
gotopriorityexpression	<String>	Read-only	Expression specifying the priority of the next policy which will get

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
			evaluated if the current policy rule evaluates to TRUE.
labeltype	<String>	Read-only	Type of policy label invocation. Possible values = reqvserver, resvserver, policylabel
activepolicy	<Integer>	Read-only	Indicates whether policy is bound or not.
__count	<Double>	Read-only	count parameter

Operations

(click to see Properties)

GET| GET (ALL)| COUNT

Some options that you can use for each operations:

- **Getting warnings in response:**NITRO allows you to get warnings in an operation by specifying the "warning" query parameter as "yes". For example, to get warnings while connecting to the Citrix ADC appliance, the URL is as follows:

  http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/login?warning=yes

  If any, the warnings are displayed in the response payload with the HTTP code "209 X-NITRO-WARNING".

- **Authenticated access for individual NITRO operations:**NITRO allows you to logon to the Citrix ADC appliance to perform individual operations. You can use this option instead of creating a NITRO session (using the login object) and then using that session to perform all operations,

  To do this, you must specify the username and password in the request header of the NITRO request as follows:

X-NITRO-USER:<username>

X-NITRO-PASS:<password>

Note: In such cases, make sure that the request header DOES not include the following:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue>

`*Note:`*

Mandatory parameters are marked in redand placeholder content is marked in <green>.

get

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/responderpolicy_lbvserver_binding/name_value<String>

Query-parameters:

filter

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/responderpolicy_lbvserver_binding/name_value<String>?**filter=property-name1:property-value1,property-name2:property-value2**

Use this query-parameter to get the filtered set of responderpolicy_lbvserver_binding resources configured on Citrix ADC. Filtering can be done on any of the properties of the resource.

pagination

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/responderpolicy_lbvserver_binding/name_value<String>?**pagesize=#no;pageno=#no**

Use this query-parameter to get the responderpolicy_lbvserver_binding resources in chunks.

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "responderpolicy_lbvserver_binding": [ {

"boundto":<String_value>,

"name":<String_value>,

"priority":<Double_value>,

"labelname":<String_value>,

"gotopriorityexpression":<String_value>,

"labeltype":<String_value>,

"activepolicy":<Integer_value>

}]}
```

get (all)

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/responderpolicy_lbvserver_binding

Query-parameters:

bulkbindings

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/responderpolicy_lbvserver_binding?**bulkbindings=yes**

NITRO allows you to fetch bindings in bulk.

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "responderpolicy_lbvserver_binding": [ {

"boundto":<String_value>,

"name":<String_value>,

"priority":<Double_value>,

"labelname":<String_value>,

"gotopriorityexpression":<String_value>,

"labeltype":<String_value>,

"activepolicy":<Integer_value>

}]}
```

count

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/responderpolicy_lbvserver_binding/name_value<String>?**count=yes**

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{"responderpolicy_lbvserver_binding": [ { "__count": "#no"} ] }
```

responderpolicy_responderglobal_binding

Binding object showing the responderglobal that can be bound to responderpolicy.

Properties

(click to see Operations)

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
boundto	<String>	Read-write	Location where policy is bound.
name	<String>	Read-write	Name of the responder policy for which to display settings.
priority	<Double>	Read-only	Specifies the priority of the policy.
labelname	<String>	Read-only	Name of the label to invoke if the current policy rule evaluates to TRUE.
gotopriorityexpression	<String>	Read-only	Expression specifying the priority of the next policy which will get

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
			evaluated if the current policy rule evaluates to TRUE.
labeltype	<String>	Read-only	Type of policy label invocation. Possible values = reqvserver, resvserver, policylabel
activepolicy	<Integer>	Read-only	Indicates whether policy is bound or not.
__count	<Double>	Read-only	count parameter

Operations

(click to see Properties)

GET| GET (ALL)| COUNT

Some options that you can use for each operations:

- **Getting warnings in response:**NITRO allows you to get warnings in an operation by specifying the "warning" query parameter as "yes". For example, to get warnings while connecting to the Citrix ADC appliance, the URL is as follows:

  http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/login?warning=yes

  If any, the warnings are displayed in the response payload with the HTTP code "209 X-NITRO-WARNING".

- **Authenticated access for individual NITRO operations:**NITRO allows you to logon to the Citrix ADC appliance to perform individual operations. You can use this option instead of creating a NITRO session (using the login object) and then using that session to perform all operations,

  To do this, you must specify the username and password in the request header of the NITRO request as follows:

X-NITRO-USER:<username>

X-NITRO-PASS:<password>

Note: In such cases, make sure that the request header DOES not include the following:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue>

`*Note:`*

Mandatory parameters are marked in redand placeholder content is marked in <green>.

get

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/responderpolicy_responderglobal_binding/name_value<String>

Query-parameters:

filter

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/responderpolicy_responderglobal_binding/name_value<String>?**filter=property-name1:property-value1,property-name2:property-value2**

Use this query-parameter to get the filtered set of responderpolicy_responderglobal_binding resources configured on Citrix ADC. Filtering can be done on any of the properties of the resource.

pagination

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/responderpolicy_responderglobal_binding/name_value<String>?**pagesize=#no;pageno=#no**

Use this query-parameter to get the responderpolicy_responderglobal_binding resources in chunks.

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "responderpolicy_responderglobal_binding": [ {

"boundto":<String_value>,

"name":<String_value>,

"priority":<Double_value>,

"labelname":<String_value>,

"gotopriorityexpression":<String_value>,

"labeltype":<String_value>,

"activepolicy":<Integer_value>

}]}
```

get (all)

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/responderpolicy_responderglobal_binding

Query-parameters:

bulkbindings

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/responderpolicy_responderglobal_binding?**bulkbindings=yes**

NITRO allows you to fetch bindings in bulk.

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "responderpolicy_responderglobal_binding": [ {

"boundto":<String_value>,

"name":<String_value>,

"priority":<Double_value>,

"labelname":<String_value>,

"gotopriorityexpression":<String_value>,

"labeltype":<String_value>,

"activepolicy":<Integer_value>

}]}
```

count

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/responderpolicy_responderglobal_binding/name_value<String>?**count=yes**

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{"responderpolicy_responderglobal_binding": [ { "__count": "#no"} ] }
```

responderpolicy_responderpolicylabel_binding

Binding object showing the responderpolicylabel that can be bound to responderpolicy.

Properties

(click to see Operations)

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
boundto	<String>	Read-write	Location where policy is bound.
name	<String>	Read-write	Name of the responder policy for which to display settings.
priority	<Double>	Read-only	Specifies the priority of the policy.
labelname	<String>	Read-only	Name of the label to invoke if the current policy rule evaluates to TRUE.
gotopriorityexpression	<String>	Read-only	Expression specifying the priority of the next policy which will get

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
			evaluated if the current policy rule evaluates to TRUE.
labeltype	<String>	Read-only	Type of policy label invocation. Possible values = reqvserver, resvserver, policylabel
activepolicy	<Integer>	Read-only	Indicates whether policy is bound or not.
__count	<Double>	Read-only	count parameter

Operations

(click to see Properties)

GET| GET (ALL)| COUNT

Some options that you can use for each operations:

- **Getting warnings in response:**NITRO allows you to get warnings in an operation by specifying the "warning" query parameter as "yes". For example, to get warnings while connecting to the Citrix ADC appliance, the URL is as follows:

  http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/login?warning=yes

  If any, the warnings are displayed in the response payload with the HTTP code "209 X-NITRO-WARNING".

- **Authenticated access for individual NITRO operations:**NITRO allows you to logon to the Citrix ADC appliance to perform individual operations. You can use this option instead of creating a NITRO session (using the login object) and then using that session to perform all operations,

  To do this, you must specify the username and password in the request header of the NITRO request as follows:

X-NITRO-USER:<username>

X-NITRO-PASS:<password>

Note: In such cases, make sure that the request header DOES not include the following:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue>

`*Note:`*

Mandatory parameters are marked in redand placeholder content is marked in <green>.

get

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/responderpolicy_responderpolicylabel_binding/name_value<String>

Query-parameters:

filter

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/responderpolicy_responderpolicylabel_binding/name_value<String>?**filter=property-name1:property-value1,property-name2:property-value2**

Use this query-parameter to get the filtered set of responderpolicy_responderpolicylabel_binding resources configured on Citrix ADC. Filtering can be done on any of the properties of the resource.

pagination

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/responderpolicy_responderpolicylabel_binding/name_value<String>?**pagesize=#no;pageno=#no**

Use this query-parameter to get the responderpolicy_responderpolicylabel_binding resources in chunks.

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "responderpolicy_responderpolicylabel_binding": [ {

"boundto":<String_value>,

"name":<String_value>,

"priority":<Double_value>,

"labelname":<String_value>,

"gotopriorityexpression":<String_value>,

"labeltype":<String_value>,

"activepolicy":<Integer_value>

}]}
```

get (all)

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/responderpolicy_responderpolicylabel_binding

Query-parameters:

bulkbindings

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/responderpolicy_responderpolicylabel_binding?**bulkbindings=yes**

NITRO allows you to fetch bindings in bulk.

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "responderpolicy_responderpolicylabel_binding": [ {

"boundto":<String_value>,

"name":<String_value>,

"priority":<Double_value>,

"labelname":<String_value>,

"gotopriorityexpression":<String_value>,

"labeltype":<String_value>,

"activepolicy":<Integer_value>

}]}
```

count

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/responderpolicy_responderpolicylabel_binding/name_value<String>?**count=yes**

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{"responderpolicy_responderpolicylabel_binding": [ { "__count": "#no"} ] }
```

Rewrite

Rewrite

Rewrite configuration.

NAME	DESCRIPTION
rewriteaction	Configuration for rewrite action
rewriteglobal_binding	Binding object showing the resources that can be bound to rewriteglobal
rewriteglobal_rewritepolicy_binding	Binding object showing the rewritepolicy that can be bound to rewriteglobal
rewriteparam	Configuration for rewrite parameter
rewritepolicy	Configuration for rewrite policy
rewritepolicy_binding	Binding object showing the resources that can be bound to rewritepolicy
rewritepolicy_csvserver_binding	Binding object showing the csvserver that can be bound to rewritepolicy

NAME	DESCRIPTION
rewritepolicy_lbvserver_binding	Binding object showing the lbvserver that can be bound to rewritepolicy
rewritepolicy_rewriteglobal_binding	Binding object showing the rewriteglobal that can be bound to rewritepolicy
rewritepolicy_rewritepolicylabel_binding	Binding object showing the rewritepolicylabel that can be bound to rewritepolicy
rewritepolicylabel	Configuration for rewrite policy label
rewritepolicylabel_binding	Binding object showing the resources that can be bound to rewritepolicylabel
rewritepolicylabel_policybinding_binding	Binding object showing the policybinding that can be bound to rewritepolicylabel
rewritepolicylabel_rewritepolicy_binding	Binding object showing the rewritepolicy that can be bound to rewritepolicylabel

rewriteaction

Configuration for rewrite action resource.

Properties

(click to see Operations)

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
name	<String>	Read-write	Name for the user-defined rewrite action. Must begin with a letter, number, or the underscore character (_), and must contain only letters, numbers, and the hyphen (-), period (.) hash (#), space ( ), at (@), equals (=), colon (:),

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
			and underscore characters. Can be changed after the rewrite policy is added. The following requirement applies only to the Citrix ADC CLI: If the name includes one or more spaces, enclose the name in double or single quotation marks (for example, "my rewrite action" or 'my rewrite action').
type	<String>	Read-write	Type of user-defined rewrite action. The information that you provide for, and the effect of, each type are as follows:: * REPLACE <target> <string_builder_expr>. Replaces the string with the string-builder expression. * REPLACE_ALL <target> <string_builder_expr1> -(pattern|

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
			search) <string_builder_expr2>. In the request or response specified by <target>, replaces all occurrences of the string defined by <string_builder_expr1> with the string defined by <string_builder_expr2>. You can use a PCRE-format pattern or the search facility to find the strings to be replaced. * REPLACE_HTTP_RES <string_builder_expr>. Replaces the complete HTTP response with the string defined by the string-builder expression. * REPLACE_SIP_RES <target> - Replaces the complete SIP response with the

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
			string specified by <target>. * INSERT_HTTP_HEADER <header_string_builder_expr> <contents_string_builder_expr>. Inserts the HTTP header specified by <header_string_builder_expr> and header contents specified by <contents_string_builder_expr>. * DELETE_HTTP_HEADER <target>. Deletes the HTTP header specified by <target>. * CORRUPT_HTTP_HEADER <target>. Replaces the header name of all occurrences of the HTTP header specified by <target> with a corrupted name, so that it will not be

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
			recognized by the receiver Example: MY_HEADER is changed to MHEY_ADER. * INSERT_BEFORE <string_builder_expr1> <string_builder_expr1>. Finds the string specified in <string_builder_expr1> and inserts the string in <string_builder_expr2> before it. * INSERT_BEFORE_ALL <target> <string_builder_expr1> -(pattern|search) <string_builder_expr2>. In the request or response specified by <target>, locates all occurrences of the string specified in <string_builder_expr1> and inserts the string specified in <string_builder_expr2> before each. You can use a

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
			PCRE-format pattern or the search facility to find the strings. * INSERT_AFTER <string_builder_expr1> <string_builder_expr2>. Finds the string specified in <string_builder_expr1>, and inserts the string specified in <string_builder_expr2> after it. * INSERT_AFTER_ALL <target> <string_builder_expr1> -(pattern|search) <string_builder_expr>. In the request or response specified by <target>, locates all occurrences of the string specified by <string_builder_expr1> and inserts the string specified by <string_builder_expr2> after each. You can use a PCRE-format pattern or

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
			the search facility to find the strings. * DELETE <target>. Finds and deletes the specified target. * DELETE_ALL <target> -(pattern|search) <string_builder_expr>. In the request or response specified by <target>, locates and deletes all occurrences of the string specified by <string_builder_expr>. You can use a PCRE-format pattern or the search facility to find the strings. * REPLACE_DIAMETER_HEADER_FIELD <target> <field value>. In the request or response modify the header field specified by <target>. Use Diameter.req.flags.SET(<flag>) or Diameter.req.flags.UNSET<flag> as

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
			'stringbuilderexpression' to set or unset flags. * REPLACE_DNS_HEADER_FIELD <target>. In the request or response modify the header field specified by <target>. * REPLACE_DNS_ANSWER_SECTION <target>. Replace the DNS answer section in the response. This is currently applicable for A and AAAA records only. Use DNS.NEW_RRSET_A ; DNS.NEW_RRSET_AAAA expressions to configure the new answer section . Possible values = noop, delete, insert_http_header, delete_http_header, corrupt_http_header, insert_before, insert_after,

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
			replace, replace_http_res, delete_all, replace_all, insert_before_all, insert_after_all, clientless_vpn_encode, clientless_vpn_encode_all, clientless_vpn_decode, clientless_vpn_decode_all, insert_sip_header, delete_sip_header, corrupt_sip_header, replace_sip_res, replace_diameter_header_field, replace_dns_header_field, replace_dns_answer_section
target	<String>	Read-write	Expression that specifies which part of the request or response to rewrite. Minimum length = 1
stringbuilderexpr	<String>	Read-write	Expression that specifies the content to insert into the request or response at the specified location, or that

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
			replaces the specified string.
pattern	<String>	Read-write	DEPRECATED in favor of -search: Pattern that is used to match multiple strings in the request or response. The pattern may be a string literal (without quotes) or a PCRE-format regular expression with a delimiter that consists of any printable ASCII non-alphanumeric character except for the underscore (_) and space ( ) that is not otherwise used in the expression. Example: re~https?://|HTTPS?://~ The preceding regular expression can use the tilde (~) as the delimiter because that character does not appear in the regular expression itself. Used in the INSERT_BEFORE_ALL, INSERT_AFTER_ALL, REPLACE_ALL, and DELETE_ALL action types.

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
search	<String>	Read-write	Search facility that is used to match multiple strings in the request or response. Used in the INSERT_BEFORE_ALL, INSERT_AFTER_ALL, REPLACE_ALL, and DELETE_ALL action types. The following search types are supported: * Text ("text(string)") - A literal string. Example: -search text("hello") * Regular expression ("regex(re<delimiter>regular exp<delimiter>)") - Pattern that is used to match multiple strings in the request or response. The pattern may be a PCRE-format regular expression with a delimiter that consists of any printable ASCII non-alphanumeric character except for the underscore (_)

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
			and space ( ) that is not otherwise used in the expression. Example: -search regex(re~\^hello*~) The preceding regular expression can use the tilde (~) as the delimiter because that character does not appear in the regular expression itself. * XPath ("xpath(xp<delimiter>xpath expression<delimiter>)") - An XPath expression to search XML. The delimiter has the same rules as for regex. Example: -search xpath(xp%/a/b%) * JSON ("xpath_json(xp<delimiter>xpath expression<delimiter>)") - An XPath expression to search JSON. The delimiter has the same rules as for

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
			regex. Example: -search xpath_json(xp%/a/b%) NOTE: JSON searches use the same syntax as XPath searches, but operate on JSON files instead of standard XML files. * HTML ("xpath_html(xp<delimiter>xpath expression<delimiter>)") - An XPath expression to search HTML. The delimiter has the same rules as for regex. Example: -search xpath_html(xp%/html/body%) NOTE: HTML searches use the same syntax as XPath searches, but operate on HTML files instead of standard XML files; HTML 5 rules for the file syntax are

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
			used; HTML 4 and later are supported. * Patset ("patset(patset)") - A predefined pattern set. Example: -search patset("patset1"). * Datset ("dataset(dataset)") - A predefined dataset. Example: -search dataset("dataset1"). * AVP ("avp(avp number)") - AVP number that is used to match multiple AVPs in a Diameter/ Radius Message. Example: -search avp(999) Note: for all these the TARGET prefix can be used in the replacement expression to specify the text that was selected by the -search parameter, optionally adjusted

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
			by the -refineSearch parameter. Example: TARGET.BEFORE_STR(",").
bypasssafetycheck	<String>	Read-write	Bypass the safety check and allow unsafe expressions. An unsafe expression is one that contains references to message elements that might not be present in all messages. If an expression refers to a missing request element, an empty string is used instead. Default value: NO Possible values = YES, NO
refinesearch	<String>	Read-write	Specify additional criteria to refine the results of the search. Always starts with the "extend(m,n)" operation, where 'm' specifies number of bytes to the left of selected data and 'n' specifies number of bytes to the right

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
			of selected data to extend the selected area. You can use refineSearch only on body expressions, and for the INSERT_BEFORE_ALL, INSERT_AFTER_ALL, REPLACE_ALL, and DELETE_ALL action types. Example: -refineSearch 'EXTEND(10, 20).REGEX_SELECT(re~0x[0-9a-zA-Z]+~).
comment	<String>	Read-write	Comment. Can be used to preserve information about this rewrite action.
newname	<String>	Read-write	New name for the rewrite action. Must begin with a letter, number, or the underscore character (_), and must contain only letters, numbers, and the hyphen (-), period (.) hash (#),

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
			space ( ), at (@), equals (=), colon (:), and underscore characters. Can be changed after the rewrite policy is added. The following requirement applies only to the Citrix ADC CLI: If the name includes one or more spaces, enclose the name in double or single quotation marks (for example, "my rewrite action" or 'my rewrite action'). Minimum length = 1
hits	<Double>	Read-only	The number of times the action has been taken.
undefhits	<Double>	Read-only	The number of times the action resulted in UNDEF.
referencecount	<Double>	Read-only	The number of references to the action.
description	<String>	Read-only	Description of the action.

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
isdefault	<Boolean>	Read-only	A value of true is returned if it is a default rewriteaction.
__count	<Double>	Read-only	count parameter

Operations

(click to see Properties)

ADD| DELETE| UPDATE| UNSET| RENAME| GET (ALL)| GET| COUNT

Some options that you can use for each operations:

- **Getting warnings in response:**NITRO allows you to get warnings in an operation by specifying the "warning" query parameter as "yes". For example, to get warnings while connecting to the Citrix ADC appliance, the URL is as follows:

  http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/login?warning=yes

  If any, the warnings are displayed in the response payload with the HTTP code "209 X-NITRO-WARNING".

- **Authenticated access for individual NITRO operations:**NITRO allows you to logon to the Citrix ADC appliance to perform individual operations. You can use this option instead of creating a NITRO session (using the login object) and then using that session to perform all operations,

  To do this, you must specify the username and password in the request header of the NITRO request as follows:

  X-NITRO-USER:<username>

  X-NITRO-PASS:<password>

  Note: In such cases, make sure that the request header DOES not include the following:

  Cookie:NITRO_AUTH_TOKEN=<tokenvalue>

`*Note:`*

Mandatory parameters are marked in redand placeholder content is marked in <green>.

add

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/rewriteaction

HTTP Method:POST

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Content-Type:application/json

Request Payload:

```
{"rewriteaction":{

<b>"name":<String_value>,

</b><b>"type":<String_value>,

</b><b>"target":<String_value>,

</b>"stringbuilderexpr":<String_value>,

"pattern":<String_value>,

"search":<String_value>,

"bypasssafetycheck":<String_value>,

"refinesearch":<String_value>,

"comment":<String_value>

}}
```

Response:

HTTP Status Code on Success: 201 Created HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error

delete

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/rewriteaction/name_value<String>

HTTP Method:DELETE

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue>

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error

update

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/rewriteaction

HTTP Method:PUT

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Content-Type:application/json

Request Payload:

```
{"rewriteaction":{

<b>"name":<String_value>,

</b>"target":<String_value>,

"stringbuilderexpr":<String_value>,

"bypasssafetycheck":<String_value>,

"pattern":<String_value>,

"search":<String_value>,

"bypasssafetycheck":<String_value>,
```

```
"refinesearch":<String_value>,

"comment":<String_value>

}}
```

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error

unset

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/rewriteaction?**action=unset**

HTTP Method:POST

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Content-Type:application/json

Request Payload:

```
{"rewriteaction":{

<b>"name":<String_value>,

</b>"stringbuilderexpr":true,

"refinesearch":true,

"comment":true

}}
```

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error

rename

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/rewriteaction?**action=rename**

HTTP Method:POST

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Content-Type:application/json

Request Payload:

```
{"rewriteaction":{

<b>"name":<String_value>,

</b><b>"newname":<String_value>

</b>}}
```

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error

get (all)

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/rewriteaction

Query-parameters:

attrs

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/rewriteaction?**attrs=property-name1,property-name2**

Use this query parameter to specify the resource details that you want to retrieve.

filter

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/rewriteaction?**filter=property-name1:property-val1,property-name2:property-val2**

Use this query-parameter to get the filtered set of rewriteaction resources configured on Citrix ADC. Filtering can be done on any of the properties of the resource.

view

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/rewriteaction?**view=summary**

Note: By default, the retrieved results are displayed in detail view (?view=detail).

pagination

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/rewriteaction?**pagesize=#no;pageno=#no**

Use this query-parameter to get the rewriteaction resources in chunks.

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "rewriteaction": [ {

"name":<String_value>,

"type":<String_value>,

"target":<String_value>,
```

```
"stringbuilderexpr":<String_value>,

"pattern":<String_value>,

"search":<String_value>,

"bypasssafetycheck":<String_value>,

"refinesearch":<String_value>,

"hits":<Double_value>,

"undefhits":<Double_value>,

"referencecount":<Double_value>,

"description":<String_value>,

"isdefault":<Boolean_value>,

"comment":<String_value>

}]}
```

get

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/rewriteaction/name_value<String>

Query-parameters:

attrs

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/rewriteaction/name_value<String>?**attrs=property-name1,property-name2**

Use this query parameter to specify the resource details that you want to retrieve.

view

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/rewriteaction/name_value<String>?**view=summary**

Note: By default, the retrieved results are displayed in detail view (?view=detail).

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "rewriteaction": [ {

"name":<String_value>,

"type":<String_value>,

"target":<String_value>,

"stringbuilderexpr":<String_value>,

"pattern":<String_value>,

"search":<String_value>,

"bypasssafetycheck":<String_value>,

"refinesearch":<String_value>,

"hits":<Double_value>,

"undefhits":<Double_value>,

"referencecount":<Double_value>,

"description":<String_value>,
```

```
"isdefault":<Boolean_value>,

"comment":<String_value>

}]}
```

count

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/rewriteaction?**count=yes**

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "rewriteaction": [ { "__count": "#no"} ] }
```

rewriteglobal_binding

Binding object which returns the resources bound to rewriteglobal.

Properties

(click to see Operations)

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
rewriteglobal_rewrite policy_binding	<rewriteglobal_rewrit epolicy_binding[]>	Read-only	rewritepolicy that can be bound to rewriteglobal.

Operations

(click to see Properties)

GET| GET (ALL)

Some options that you can use for each operations:

- **Getting warnings in response:**NITRO allows you to get warnings in an operation by specifying the "warning" query parameter as "yes". For example, to get warnings while connecting to the Citrix ADC appliance, the URL is as follows:

  http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/login?warning=yes

  If any, the warnings are displayed in the response payload with the HTTP code "209 X-NITRO-WARNING".

- **Authenticated access for individual NITRO operations:**NITRO allows you to logon to the Citrix ADC appliance to perform individual operations. You can use this option instead of creating a NITRO session (using the login object) and then using that session to perform all operations,

  To do this, you must specify the username and password in the request header of the NITRO request as follows:

  X-NITRO-USER:<username>

  X-NITRO-PASS:<password>

  Note: In such cases, make sure that the request header DOES not include the following:

  Cookie:NITRO_AUTH_TOKEN=<tokenvalue>

`*Note:`*

Mandatory parameters are marked in redand placeholder content is marked in <green>.

get

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/rewriteglobal_binding

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "rewriteglobal_binding": [ {

"rewriteglobal_rewritepolicy_binding":<rewriteglobal_rewritepolicy_binding
[]_value>

}]}
```

get (all)

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/rewriteglobal_binding

Query-parameters:

bulkbindings

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/rewriteglobal_binding?**bulkbindings=yes**

NITRO allows you to fetch bindings in bulk.

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "rewriteglobal_binding": [ {

"rewriteglobal_rewritepolicy_binding":<rewriteglobal_rewritepolicy_binding
[]_value>

}]}
```

rewriteglobal_rewritepolicy_binding

Binding object showing the rewritepolicy that can be bound to rewriteglobal.

Properties

(click to see Operations)

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
priority	<Double>	Read-write	Specifies the priority of the policy.
globalbindtype	<String>	Read-write	. Default value: SYSTEM_GLOBAL Possible values = SYSTEM_GLOBAL,

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
			VPN_GLOBAL, RNAT_GLOBAL
policyname	<String>	Read-write	Name of the rewrite policy.
labelname	<String>	Read-write	* If labelType is policylabel, name of the policy label to invoke. * If labelType is reqvserver or resvserver, name of the virtual server to which to forward the request of response.
gotopriorityexpression	<String>	Read-write	Expression specifying the priority of the next policy which will get evaluated if the current policy rule evaluates to TRUE.
invoke	<Boolean>	Read-write	Terminate evaluation of policies bound to the current policy label, and then forward the request to the specified virtual server or evaluate the specified policy label.
type	<String>	Read-write	The bindpoint to which to policy is bound. Possible values = REQ_OVERRIDE, REQ_DEFAULT, RES_OVERRIDE,

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
			RES_DEFAULT, OTHERTCP_REQ_OVERRIDE, OTHERTCP_REQ_DEFAULT, OTHERTCP_RES_OVERRIDE, OTHERTCP_RES_DEFAULT, SIPUDP_REQ_OVERRIDE, SIPUDP_REQ_DEFAULT, SIPUDP_RES_OVERRIDE, SIPUDP_RES_DEFAULT, SIPTCP_REQ_OVERRIDE, SIPTCP_REQ_DEFAULT, SIPTCP_RES_OVERRIDE, SIPTCP_RES_DEFAULT, DIAMETER_REQ_OVERRIDE, DIAMETER_REQ_DEFAULT, DIAMETER_RES_OVERRIDE, DIAMETER_RES_DEFAULT, RADIUS_REQ_OVERRIDE, RADIUS_REQ_DE

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
			FAULT, RADIUS_RES_OVERRIDE, RADIUS_RES_DEFAULT, DNS_REQ_OVERRIDE, DNS_REQ_DEFAULT, DNS_RES_OVERRIDE, DNS_RES_DEFAULT
labeltype	<String>	Read-write	Type of invocation. Available settings function as follows: * reqvserver - Forward the request to the specified request virtual server. * resvserver - Forward the response to the specified response virtual server. * policylabel - Invoke the specified policy label. Possible values = reqvserver, resvserver, policylabel
flowtype	<Double>	Read-only	flowtype of the bound rewrite policy.

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
numpol	<Double>	Read-only	The number of policies bound to the bindpoint.
__count	<Double>	Read-only	count parameter

Operations

(click to see Properties)

ADD:| DELETE:| GET| GET (ALL)| COUNT

Some options that you can use for each operations:

- **Getting warnings in response:**NITRO allows you to get warnings in an operation by specifying the "warning" query parameter as "yes". For example, to get warnings while connecting to the Citrix ADC appliance, the URL is as follows:

  http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/login?warning=yes

  If any, the warnings are displayed in the response payload with the HTTP code "209 X-NITRO-WARNING".

- **Authenticated access for individual NITRO operations:**NITRO allows you to logon to the Citrix ADC appliance to perform individual operations. You can use this option instead of creating a NITRO session (using the login object) and then using that session to perform all operations,

  To do this, you must specify the username and password in the request header of the NITRO request as follows:

  X-NITRO-USER:<username>

  X-NITRO-PASS:<password>

  Note: In such cases, make sure that the request header DOES not include the following:

  Cookie:NITRO_AUTH_TOKEN=<tokenvalue>

`*Note:` *

Mandatory parameters are marked in redand placeholder content is marked in <green>.

add

URL:http://<Citrix-ADC-IP-address(NSIP)/nitro/v1/config/rewriteglobal_rewritepolicy_binding

HTTP Method:PUT

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Content-Type:application/json

Request Payload:

```
{

"rewriteglobal_rewritepolicy_binding":{

<b>"policyname":<String_value>,

</b><b>"priority":<Double_value>,

</b>"gotopriorityexpression":<String_value>,

"type":<String_value>,

"invoke":<Boolean_value>,

"labeltype":<String_value>,

"labelname":<String_value>

}}
```

Response:

HTTP Status Code on Success: 201 Created HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error

delete

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/rewriteglobal_rewritepolicy_binding

Query-parameters:

args

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/rewriteglobal_rewritepolicy_binding?**args=policyname:<String_value>,type:<String_value>,priority:<Double_value>**

HTTP Method:DELETE

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue>

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error

get

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/rewriteglobal_rewritepolicy_binding

Query-parameters:

filter

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/rewriteglobal_rewritepolicy_binding?**filter=property-name1:property-value1,property-name2:property-value2**

Use this query-parameter to get the filtered set of rewriteglobal_rewritepolicy_binding resources configured on Citrix ADC. Filtering can be done on any of the properties of the resource.

pagination

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/rewriteglobal_rewritepolicy_binding?**pagesize=#no;pageno=#no**

Use this query-parameter to get the rewriteglobal_rewritepolicy_binding resources in chunks.

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "rewriteglobal_rewritepolicy_binding": [ {

"priority":<Double_value>,

"globalbindtype":<String_value>,

"policyname":<String_value>,

"labelname":<String_value>,

"gotopriorityexpression":<String_value>,

"invoke":<Boolean_value>,

"type":<String_value>,

"labeltype":<String_value>,

"flowtype":<Double_value>,

"numpol":<Double_value>

}]}
```

get (all)

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/rewriteglobal_rewritepolicy_binding

Query-parameters:

bulkbindings

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/rewriteglobal_rewritepolicy_binding?**bulkbindings=yes**

NITRO allows you to fetch bindings in bulk.

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "rewriteglobal_rewritepolicy_binding": [ {

"priority":<Double_value>,

"globalbindtype":<String_value>,

"policyname":<String_value>,

"labelname":<String_value>,

"gotopriorityexpression":<String_value>,

"invoke":<Boolean_value>,

"type":<String_value>,

"labeltype":<String_value>,

"flowtype":<Double_value>,

"numpol":<Double_value>
```

```
}]}
```

count

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/rewriteglobal_rewritepolicy_binding?**count=yes**

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "rewriteglobal_rewritepolicy_binding": [ { "__count": "#no"} ] }
```

rewriteparam

Configuration for rewrite parameter resource.

Properties

(click to see Operations)

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
undefaction	<String>	Read-write	Action to perform if the result of policy evaluation is undefined (UNDEF).

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
			An UNDEF event indicates an error condition in evaluating the expression. Available settings function as follows: * NOREWRITE - Do not modify the message. * RESET - Reset the connection and notify the user's browser, so that the user can resend the request. * DROP - Drop the message without sending a response to the user. Default value: "NOREWRITE"
timeout	<Double>	Read-write	Maximum time in milliseconds to allow for processing all the policies and their selected actions without interruption. If the timeout is reached then the evaluation causes an UNDEF to be raised and no further processing is performed. Note that

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
			some rewrites may have already been performed. Minimum value = 1 Maximum value = 5000

Operations

(click to see Properties)

UPDATE| UNSET| GET (ALL)

Some options that you can use for each operations:

- **Getting warnings in response:**NITRO allows you to get warnings in an operation by specifying the "warning" query parameter as "yes". For example, to get warnings while connecting to the Citrix ADC appliance, the URL is as follows:

  http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/login?warning=yes

  If any, the warnings are displayed in the response payload with the HTTP code "209 X-NITRO-WARNING".

- **Authenticated access for individual NITRO operations:**NITRO allows you to logon to the Citrix ADC appliance to perform individual operations. You can use this option instead of creating a NITRO session (using the login object) and then using that session to perform all operations,

  To do this, you must specify the username and password in the request header of the NITRO request as follows:

  X-NITRO-USER:<username>

  X-NITRO-PASS:<password>

  Note: In such cases, make sure that the request header DOES not include the following:

  Cookie:NITRO_AUTH_TOKEN=<tokenvalue>

`*Note:` *

Mandatory parameters are marked in redand placeholder content is marked in <green>.

update

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/rewriteparam

HTTP Method:PUT

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Content-Type:application/json

Request Payload:

```
{"rewriteparam":{

"undefaction":<String_value>,

"timeout":<Double_value>

}}
```

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error

unset

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/rewriteparam?**action=unset**

HTTP Method:POST

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Content-Type:application/json

Request Payload:

```
{"rewriteparam":{

"undefaction":true,

"timeout":true

}}
```

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error

get (all)

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/rewriteparam

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "rewriteparam": [ {

"undefaction":<String_value>,

"timeout":<Double_value>
```

```
}]}
```

rewritepolicy

Configuration for rewrite policy resource.

Properties

(click to see Operations)

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
name	<String>	Read-write	Name for the rewrite policy. Must begin with a letter, number, or the underscore character (_), and must contain only letters, numbers, and the hyphen (-), period (.) hash (#), space ( ), at (@), equals (=), colon (:), and underscore characters. Can be changed after the rewrite policy is added. The following requirement applies only to the Citrix ADC CLI: If the name includes one or more spaces, enclose the name in double or single quotation marks (for example, "my rewrite policy"

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
			or 'my rewrite policy').
rule	<String>	Read-write	Expression against which traffic is evaluated. The following requirements apply only to the Citrix ADC CLI: * If the expression includes one or more spaces, enclose the entire expression in double quotation marks. * If the expression itself includes double quotation marks, escape the quotations by using the \ character. * Alternatively, you can use single quotation marks to enclose the rule, in which case you do not have to escape the double quotation marks.
action	<String>	Read-write	Name of the rewrite action to perform if the request or

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
			response matches this rewrite policy. There are also some built-in actions which can be used. These are: * NOREWRITE - Send the request from the client to the server or response from the server to the client without making any changes in the message. * RESET - Resets the client connection by closing it. The client program, such as a browser, will handle this and may inform the user. The client may then resend the request if desired. * DROP - Drop the request without sending a response to the user.
undefaction	<String>	Read-write	Action to perform if the result of policy evaluation is undefined (UNDEF).

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
			An UNDEF event indicates an internal error condition. Only the above built-in actions can be used.
comment	<String>	Read-write	Any comments to preserve information about this rewrite policy.
logaction	<String>	Read-write	Name of messagelog action to use when a request matches this policy.
newname	<String>	Read-write	New name for the rewrite policy. Must begin with a letter, number, or the underscore character (_), and must contain only letters, numbers, and the hyphen (-), period (.) hash (#), space ( ), at (@), equals (=), colon (:), and underscore characters. The following requirement applies only to the Citrix ADC CLI: If the name includes one or more spaces, enclose the

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
			name in double or single quotation marks (for example, "my rewrite policy" or 'my rewrite policy'). Minimum length = 1
hits	<Double>	Read-only	Number of hits.
undefhits	<Double>	Read-only	Number of Undef hits.
description	<String>	Read-only	Description of the policy.
isdefault	<Boolean>	Read-only	A value of true is returned if it is a default rewritepolicy.
__count	<Double>	Read-only	count parameter

Operations

(click to see Properties)

ADD| DELETE| UPDATE| UNSET| RENAME| GET (ALL)| GET| COUNT

Some options that you can use for each operations:

- **Getting warnings in response:**NITRO allows you to get warnings in an operation by specifying the "warning" query parameter as "yes". For example, to get warnings while connecting to the Citrix ADC appliance, the URL is as follows:

  http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/login?warning=yes

  If any, the warnings are displayed in the response payload with the HTTP code "209 X-NITRO-WARNING".

- **Authenticated access for individual NITRO operations:**NITRO allows you to logon to the Citrix ADC appliance to perform individual operations. You can use this option instead of creating a NITRO session (using the login object) and then using that session to perform all operations,

  To do this, you must specify the username and password in the request header of the NITRO request as follows:

  X-NITRO-USER:<username>

  X-NITRO-PASS:<password>

  Note: In such cases, make sure that the request header DOES not include the following:

  Cookie:NITRO_AUTH_TOKEN=<tokenvalue>

`*Note:`*

Mandatory parameters are marked in redand placeholder content is marked in <green>.

add

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/rewritepolicy

HTTP Method:POST

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Content-Type:application/json

Request Payload:

```
{"rewritepolicy":{

<b>"name":<String_value>,

</b><b>"rule":<String_value>,

</b><b>"action":<String_value>,

</b>"undefaction":<String_value>,

"comment":<String_value>,
```

```
"logaction":<String_value>

}}
```

Response:

HTTP Status Code on Success: 201 Created HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error

delete

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/rewritepolicy/name_value<String>

HTTP Method:DELETE

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue>

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error

update

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/rewritepolicy

HTTP Method:PUT

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Content-Type:application/json

Request Payload:

```
{"rewritepolicy":{

<b>"name":<String_value>,
```

```
</b>"rule":<String_value>,

"action":<String_value>,

"undefaction":<String_value>,

"comment":<String_value>,

"logaction":<String_value>

}}
```

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error

unset

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/rewritepolicy?**action=unset**

HTTP Method:POST

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Content-Type:application/json

Request Payload:

```
{"rewritepolicy":{

<b>"name":<String_value>,

</b>"undefaction":true,

"comment":true,

"logaction":true
```

```
}}
```

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error

rename

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/rewritepolicy?**action=rename**

HTTP Method:POST

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Content-Type:application/json

Request Payload:

```
{"rewritepolicy":{

<b>"name":<String_value>,

</b><b>"newname":<String_value>

</b>}}
```

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error

get (all)

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/rewritepolicy

Query-parameters:

attrs

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/rewritepolicy?**attrs=property-name1,property-name2**

Use this query parameter to specify the resource details that you want to retrieve.

filter

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/rewritepolicy?**filter=property-name1:property-val1,property-name2:property-val2**

Use this query-parameter to get the filtered set of rewritepolicy resources configured on Citrix ADC. Filtering can be done on any of the properties of the resource.

view

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/rewritepolicy?**view=summary**

Note: By default, the retrieved results are displayed in detail view (?view=detail).

pagination

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/rewritepolicy?**pagesize=#no;pageno=#no**

Use this query-parameter to get the rewritepolicy resources in chunks.

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "rewritepolicy": [ {

"name":<String_value>,

"rule":<String_value>,

"action":<String_value>,

"undefaction":<String_value>,

"hits":<Double_value>,

"undefhits":<Double_value>,

"description":<String_value>,

"comment":<String_value>,

"logaction":<String_value>,

"isdefault":<Boolean_value>

}]}
```

get

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/rewritepolicy/name_value<String>

Query-parameters:

attrs

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/rewritepolicy/name_value<String>?**attrs=property-name1,property-name2**

Use this query parameter to specify the resource details that you want to retrieve.

view

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/rewritepolicy/name_value<String>?**view=summary**

Note: By default, the retrieved results are displayed in detail view (?view=detail).

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "rewritepolicy": [ {

"name":<String_value>,

"rule":<String_value>,

"action":<String_value>,

"undefaction":<String_value>,

"hits":<Double_value>,

"undefhits":<Double_value>,

"description":<String_value>,

"comment":<String_value>,

"logaction":<String_value>,

"isdefault":<Boolean_value>

}]}
```

count

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/rewritepolicy?**count=yes**

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "rewritepolicy": [ { "__count": "#no"} ] }
```

rewritepolicylabel

Configuration for rewrite policy label resource.

Properties

(click to see Operations)

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
labelname	<String>	Read-write	Name for the rewrite policy label. Must begin with a letter, number, or the underscore character (_), and must contain only letters,

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
			numbers, and the hyphen (-), period (.) hash (#), space ( ), at (@), equals (=), colon (:), and underscore characters. Cannot be changed after the rewrite policy label is added. The following requirement applies only to the Citrix ADC CLI: If the name includes one or more spaces, enclose the name in double or single quotation marks (for example, "my rewrite policy label" or 'my rewrite policy label').
transform	<String>	Read-write	Types of transformations allowed by the policies bound to the label. For Rewrite, the following types are supported: * http_req - HTTP requests * http_res - HTTP responses

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
			* othertcp_req - Non-HTTP TCP requests * othertcp_res - Non-HTTP TCP responses * url - URLs * text - Text strings * clientless_vpn_req - Citrix ADC clientless VPN requests * clientless_vpn_res - Citrix ADC clientless VPN responses * sipudp_req - SIP requests * sipudp_res - SIP responses * diameter_req - DIAMETER requests * diameter_res - DIAMETER responses * radius_req - RADIUS requests

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
			* radius_res - RADIUS responses * dns_req - DNS requests * dns_res - DNS responses. Possible values = http_req, http_res, othertcp_req, othertcp_res, url, text, clientless_vpn_req, clientless_vpn_res, sipudp_req, sipudp_res, siptcp_req, siptcp_res, diameter_req, diameter_res, radius_req, radius_res, dns_req, dns_res
comment	<String>	Read-write	Any comments to preserve information about this rewrite policy label.
newname	<String>	Read-write	New name for the rewrite policy label. Must begin with a letter, number, or the underscore character (_), and must contain only

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
			letters, numbers, and the hyphen (-), period (.) hash (#), space ( ), at (@), equals (=), colon (:), and underscore characters. The following requirement applies only to the Citrix ADC CLI: If the name includes one or more spaces, enclose the name in double or single quotation marks (for example, "my policy label" or 'my policy label'). Minimum length = 1
numpol	<Double>	Read-only	Number of polices bound to label.
hits	<Double>	Read-only	Number of times policy label was invoked.
priority	<Double>	Read-only	Specifies the priority of the policy.
gotopriorityexpression	<String>	Read-only	Expression specifying the priority of the next policy which will get evaluated if the

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
			current policy rule evaluates to TRUE.
labeltype	<String>	Read-only	Type of invocation. Available settings function as follows: * reqvserver - Forward the request to the specified request virtual server. * resvserver - Forward the response to the specified response virtual server. * policylabel - Invoke the specified policy label. Possible values = reqvserver, resvserver, policylabel
invoke_labelname	<String>	Read-only	* If labelType is policylabel, name of the policy label to invoke. * If labelType is reqvserver or resvserver, name of the virtual server to which to forward the request or response.

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
flowtype	<Double>	Read-only	Flowtype of the bound rewrite policy.
description	<String>	Read-only	Description of the policylabel.
isdefault	<Boolean>	Read-only	A value of true is returned if it is a default rewritepolicylabel.
__count	<Double>	Read-only	count parameter

Operations

(click to see Properties)

ADD| DELETE| RENAME| GET (ALL)| GET| COUNT

Some options that you can use for each operations:

- **Getting warnings in response:**NITRO allows you to get warnings in an operation by specifying the "warning" query parameter as "yes". For example, to get warnings while connecting to the Citrix ADC appliance, the URL is as follows:

  http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/login?warning=yes

  If any, the warnings are displayed in the response payload with the HTTP code "209 X-NITRO-WARNING".

- **Authenticated access for individual NITRO operations:**NITRO allows you to logon to the Citrix ADC appliance to perform individual operations. You can use this option instead of creating a NITRO session (using the login object) and then using that session to perform all operations,

  To do this, you must specify the username and password in the request header of the NITRO request as follows:

  X-NITRO-USER:<username>

  X-NITRO-PASS:<password>

Note: In such cases, make sure that the request header DOES not include the following:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue>

`*Note:` *

Mandatory parameters are marked in redand placeholder content is marked in <green>.

add

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/rewritepolicylabel

HTTP Method:POST

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Content-Type:application/json

Request Payload:

```
{"rewritepolicylabel":{

<b>"labelname":<String_value>,

</b><b>"transform":<String_value>,

</b>"comment":<String_value>

}}
```

Response:

HTTP Status Code on Success: 201 Created HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error

delete

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/rewritepolicylabel/labelname_value<String>

HTTP Method:DELETE

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue>

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error

rename

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/rewritepolicylabel?**action=rename**

HTTP Method:POST

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Content-Type:application/json

Request Payload:

```
{"rewritepolicylabel":{

<b>"labelname":<String_value>,

</b><b>"newname":<String_value>

</b>}}
```

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error

get (all)

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/rewritepolicylabel

Query-parameters:

attrs

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/rewritepolicylabel?**attrs=property-name1,property-name2**

Use this query parameter to specify the resource details that you want to retrieve.

filter

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/rewritepolicylabel?**filter=property-name1:property-val1,property-name2:property-val2**

Use this query-parameter to get the filtered set of rewritepolicylabel resources configured on Citrix ADC. Filtering can be done on any of the properties of the resource.

view

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/rewritepolicylabel?**view=summary**

Note: By default, the retrieved results are displayed in detail view (?view=detail).

pagination

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/rewritepolicylabel?**pagesize=#no;pageno=#no**

Use this query-parameter to get the rewritepolicylabel resources in chunks.

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "rewritepolicylabel": [ {

"labelname":<String_value>,

"transform":<String_value>,

"numpol":<Double_value>,

"hits":<Double_value>,

"priority":<Double_value>,

"gotopriorityexpression":<String_value>,

"labeltype":<String_value>,

"invoke_labelname":<String_value>,

"flowtype":<Double_value>,

"description":<String_value>,

"isdefault":<Boolean_value>,

"comment":<String_value>

}]}
```

get

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/rewritepolicylabel/labelname_value<String>

Query-parameters:

attrs

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/rewritepolicylabel/labelname_value<String>?**attrs=property-name1,property-name2**

Use this query parameter to specify the resource details that you want to retrieve.

view

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/rewritepolicylabel/labelname_value<String>?**view=summary**

Note: By default, the retrieved results are displayed in detail view (?view=detail).

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "rewritepolicylabel": [ {

"labelname":<String_value>,

"transform":<String_value>,

"numpol":<Double_value>,

"hits":<Double_value>,

"priority":<Double_value>,

"gotopriorityexpression":<String_value>,

"labeltype":<String_value>,

"invoke_labelname":<String_value>,
```

```
"flowtype":<Double_value>,

"description":<String_value>,

"isdefault":<Boolean_value>,

"comment":<String_value>

}]}
```

count

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/rewritepolicylabel?**count=yes**

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "rewritepolicylabel": [ { "__count": "#no"} ] }
```

rewritepolicylabel_binding

Binding object which returns the resources bound to rewritepolicylabel.

Properties

(click to see Operations)

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
labelname	<String>	Read-write	Name of the rewrite policy label.
rewritepolicylabel_policybinding_binding	<rewritepolicylabel_policybinding_binding[]>	Read-only	policybinding that can be bound to rewritepolicylabel.
rewritepolicylabel_rewritepolicy_binding	<rewritepolicylabel_rewritepolicy_binding[]>	Read-only	rewritepolicy that can be bound to rewritepolicylabel.

Operations

(click to see Properties)

GET| GET (ALL)

Some options that you can use for each operations:

- **Getting warnings in response:**NITRO allows you to get warnings in an operation by specifying the "warning" query parameter as "yes". For example, to get warnings while connecting to the Citrix ADC appliance, the URL is as follows:

  http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/login?warning=yes

  If any, the warnings are displayed in the response payload with the HTTP code "209 X-NITRO-WARNING".

- **Authenticated access for individual NITRO operations:**NITRO allows you to logon to the Citrix ADC appliance to perform individual operations. You can use this option instead of creating a NITRO session (using the login object) and then using that session to perform all operations,

  To do this, you must specify the username and password in the request header of the NITRO request as follows:

  X-NITRO-USER:<username>

  X-NITRO-PASS:<password>

  Note: In such cases, make sure that the request header DOES not include the following:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue>

`*Note:` *

Mandatory parameters are marked in redand placeholder content is marked in <green>.

get

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/rewritepolicylabel_binding/labelname_value<String>

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "rewritepolicylabel_binding": [ {

"labelname":<String_value>,

"rewritepolicylabel_policybinding_binding":<rewritepolicylabel_policybindi
ng_binding[]_value>,

"rewritepolicylabel_rewritepolicy_binding":<rewritepolicylabel_rewritepoli
cy_binding[]_value>

}]}
```

get (all)

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/rewritepolicylabel_binding

Query-parameters:

bulkbindings

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/rewritepolicylabel_binding?**bulkbindings=yes**

NITRO allows you to fetch bindings in bulk.

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "rewritepolicylabel_binding": [ {

"labelname":<String_value>,

"rewritepolicylabel_policybinding_binding":<rewritepolicylabel_policybindi
ng_binding[]_value>,

"rewritepolicylabel_rewritepolicy_binding":<rewritepolicylabel_rewritepoli
cy_binding[]_value>

}]}
```

rewritepolicylabel_policybinding_binding

Binding object showing the policybinding that can be bound to rewritepolicylabel.

Properties

(click to see Operations)

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
priority	<Double>	Read-write	Specifies the priority of the policy.
policyname	<String>	Read-write	Name of the rewrite policy to bind to the policy label.
labelname	<String>	Read-write	Name of the rewrite policy label to which to bind the policy.
invoke_labelname	<String>	Read-write	* If labelType is policylabel, name of the policy label to invoke. * If labelType is reqvserver or resvserver, name of the virtual server to which to forward the request or response.
gotopriorityexpression	<String>	Read-write	Expression specifying the priority of the next policy which will get evaluated if the current policy rule evaluates to TRUE.
invoke	<Boolean>	Read-write	Suspend evaluation of policies bound to the current policy label, and then forward the request to the specified virtual server or

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
			evaluate the specified policy label.
labeltype	<String>	Read-write	Type of invocation. Available settings function as follows: * reqvserver - Forward the request to the specified request virtual server. * resvserver - Forward the response to the specified response virtual server. * policylabel - Invoke the specified policy label. Possible values = reqvserver, resvserver, policylabel
__count	<Double>	Read-write	count parameter

Operations

(click to see Properties)

GET| GET (ALL)| COUNT

Some options that you can use for each operations:

- **Getting warnings in response:**NITRO allows you to get warnings in an operation by specifying the "warning" query parameter as "yes". For example, to get warnings while connecting to the Citrix ADC appliance, the URL is as follows:

  http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/login?warning=yes

  If any, the warnings are displayed in the response payload with the HTTP code "209 X-NITRO-WARNING".

- **Authenticated access for individual NITRO operations:**NITRO allows you to logon to the Citrix ADC appliance to perform individual operations. You can use this option instead of creating a NITRO session (using the login object) and then using that session to perform all operations,

  To do this, you must specify the username and password in the request header of the NITRO request as follows:

  X-NITRO-USER:<username>

  X-NITRO-PASS:<password>

  Note: In such cases, make sure that the request header DOES not include the following:

  Cookie:NITRO_AUTH_TOKEN=<tokenvalue>

`*Note:`*

Mandatory parameters are marked in redand placeholder content is marked in <green>.

get

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/rewritepolicylabel_policybinding_binding/labelname_value<String>

Query-parameters:

filter

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/rewritepolicylabel_policybinding_binding/labelname_value<String>?**filter=property-name1:property-value1,property-name2:property-value2**

Use this query-parameter to get the filtered set of rewritepolicylabel_policybinding_binding resources configured on Citrix ADC. Filtering can be done on any of the properties of the resource.

pagination

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/rewritepolicylabel_policybinding_binding/labelname_value<String>?**pagesize=#no;pageno=#no**

Use this query-parameter to get the rewritepolicylabel_policybinding_binding resources in chunks.

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "rewritepolicylabel_policybinding_binding": [ {

"priority":<Double_value>,

"policyname":<String_value>,

"labelname":<String_value>,

"invoke_labelname":<String_value>,

"gotopriorityexpression":<String_value>,

"invoke":<Boolean_value>,

"labeltype":<String_value>

}]}
```

get (all)

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/rewritepolicylabel_policybinding_binding

Query-parameters:

bulkbindings

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/rewritepolicylabel_policybinding_binding?**bulkbindings=yes**

NITRO allows you to fetch bindings in bulk.

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "rewritepolicylabel_policybinding_binding": [ {

"priority":<Double_value>,

"policyname":<String_value>,

"labelname":<String_value>,

"invoke_labelname":<String_value>,

"gotopriorityexpression":<String_value>,

"invoke":<Boolean_value>,

"labeltype":<String_value>

}]}
```

count

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/rewritepolicylabel_policybinding_binding/labelname_value<String>?**count=yes**

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{"rewritepolicylabel_policybinding_binding": [ { "__count": "#no"} ] }
```

rewritepolicylabel_rewritepolicy_binding

Binding object showing the rewritepolicy that can be bound to rewritepolicylabel.

Properties

(click to see Operations)

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
priority	<Double>	Read-write	Specifies the priority of the policy.
policyname	<String>	Read-write	Name of the rewrite policy to bind to the policy label.

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
labelname	<String>	Read-write	Name of the rewrite policy label to which to bind the policy.
invoke_labelname	<String>	Read-write	* If labelType is policylabel, name of the policy label to invoke. * If labelType is reqvserver or resvserver, name of the virtual server to which to forward the request or response.
gotopriorityexpression	<String>	Read-write	Expression specifying the priority of the next policy which will get evaluated if the current policy rule evaluates to TRUE.
invoke	<Boolean>	Read-write	Suspend evaluation of policies bound to the current policy label, and then forward the request to the specified virtual server or evaluate the specified policy label.
labeltype	<String>	Read-write	Type of invocation. Available settings function as follows: * reqvserver - Forward the request to the specified request virtual server. * resvserver - Forward the response to the

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
			specified response virtual server. * policylabel - Invoke the specified policy label. Possible values = reqvserver, resvserver, policylabel
__count	<Double>	Read-write	count parameter

Operations

(click to see Properties)

ADD:| DELETE:| GET| GET (ALL)| COUNT

Some options that you can use for each operations:

- **Getting warnings in response:**NITRO allows you to get warnings in an operation by specifying the "warning" query parameter as "yes". For example, to get warnings while connecting to the Citrix ADC appliance, the URL is as follows:

  http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/login?warning=yes

  If any, the warnings are displayed in the response payload with the HTTP code "209 X-NITRO-WARNING".

- **Authenticated access for individual NITRO operations:**NITRO allows you to logon to the Citrix ADC appliance to perform individual operations. You can use this option instead of creating a NITRO session (using the login object) and then using that session to perform all operations,

  To do this, you must specify the username and password in the request header of the NITRO request as follows:

  X-NITRO-USER:<username>

  X-NITRO-PASS:<password>

  Note: In such cases, make sure that the request header DOES not include the following:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue>

`*Note:`*

Mandatory parameters are marked in redand placeholder content is marked in <green>.

add

URL:http://<Citrix-ADC-IP-address(NSIP)/nitro/v1/config/rewritepolicylabel_rewritepolicy_binding

HTTP Method:PUT

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Content-Type:application/json

Request Payload:

```
{

"rewritepolicylabel_rewritepolicy_binding":{

<b>"labelname":<String_value>,

</b><b>"policyname":<String_value>,

</b><b>"priority":<Double_value>,

</b>"gotopriorityexpression":<String_value>,

"invoke":<Boolean_value>,

"labeltype":<String_value>,

"invoke_labelname":<String_value>

}}
```

Response:

HTTP Status Code on Success: 201 Created HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error

delete

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/rewritepolicylabel_rewritepolicy_binding/labelname_value<String>

Query-parameters:

args

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/rewritepolicylabel_rewritepolicy_binding/labelname_value<String>?**args=policyname:<String_value>,priority:<Double_value>**

HTTP Method:DELETE

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue>

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error

get

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/rewritepolicylabel_rewritepolicy_binding/labelname_value<String>

Query-parameters:

filter

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/rewritepolicylabel_rewritepolicy_binding/labelname_value<String>?**filter=property-name1:property-value1,property-name2:property-value2**

Use this query-parameter to get the filtered set of rewritepolicylabel_rewritepolicy_binding resources configured on Citrix ADC. Filtering can be done on any of the properties of the resource.

pagination

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/rewritepolicylabel_rewritepolicy_binding/labelname_value<String>?**pagesize=#no;pageno=#no**

Use this query-parameter to get the rewritepolicylabel_rewritepolicy_binding resources in chunks.

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "rewritepolicylabel_rewritepolicy_binding": [ {

"priority":<Double_value>,

"policyname":<String_value>,

"labelname":<String_value>,

"invoke_labelname":<String_value>,

"gotopriorityexpression":<String_value>,

"invoke":<Boolean_value>,

"labeltype":<String_value>

}]}
```

get (all)

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/rewritepolicylabel_rewritepolicy_binding

Query-parameters:

bulkbindings

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/rewritepolicylabel_rewritepolicy_binding?**bulkbindings=yes**

NITRO allows you to fetch bindings in bulk.

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "rewritepolicylabel_rewritepolicy_binding": [ {

"priority":<Double_value>,

"policyname":<String_value>,

"labelname":<String_value>,

"invoke_labelname":<String_value>,

"gotopriorityexpression":<String_value>,

"invoke":<Boolean_value>,
```

```
"labeltype":<String_value>

}]}
```

count

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/rewritepolicylabel_rewritepolicy_binding/labelname_value<String>?**count=yes**

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{"rewritepolicylabel_rewritepolicy_binding": [ { "__count": "#no"} ] }
```

rewritepolicy_binding

Binding object which returns the resources bound to rewritepolicy.

Properties

(click to see Operations)

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
name	<String>	Read-write	Name of the rewrite policy.

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
rewritepolicy_csvserver_binding	<rewritepolicy_csvserver_binding[]>	Read-only	csvserver that can be bound to rewritepolicy.
rewritepolicy_rewriteglobal_binding	<rewritepolicy_rewriteglobal_binding[]>	Read-only	rewriteglobal that can be bound to rewritepolicy.
rewritepolicy_lbvserver_binding	<rewritepolicy_lbvserver_binding[]>	Read-only	lbvserver that can be bound to rewritepolicy.
rewritepolicy_rewritepolicylabel_binding	<rewritepolicy_rewritepolicylabel_binding[]>	Read-only	rewritepolicylabel that can be bound to rewritepolicy.

Operations

(click to see Properties)

GET| GET (ALL)

Some options that you can use for each operations:

- **Getting warnings in response:**NITRO allows you to get warnings in an operation by specifying the "warning" query parameter as "yes". For example, to get warnings while connecting to the Citrix ADC appliance, the URL is as follows:

  http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/login?warning=yes

  If any, the warnings are displayed in the response payload with the HTTP code "209 X-NITRO-WARNING".

- **Authenticated access for individual NITRO operations:**NITRO allows you to logon to the Citrix ADC appliance to perform individual operations. You can use this option instead of creating a NITRO session (using the login object) and then using that session to perform all operations,

  To do this, you must specify the username and password in the request header of the NITRO request as follows:

  X-NITRO-USER:<username>

X-NITRO-PASS:<password>

Note: In such cases, make sure that the request header DOES not include the following:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue>

`*Note:`*

Mandatory parameters are marked in redand placeholder content is marked in <green>.

get

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/rewritepolicy_binding/name_value<String>

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "rewritepolicy_binding": [ {

"name":<String_value>,

"rewritepolicy_lbvserver_binding":<rewritepolicy_lbvserver_binding[]_value
>,

"rewritepolicy_rewritepolicylabel_binding":<rewritepolicy_rewritepolicylab
el_binding[]_value>,

"rewritepolicy_rewriteglobal_binding":<rewritepolicy_rewriteglobal_binding
[]_value>,
```

```
"rewritepolicy_csvserver_binding":<rewritepolicy_csvserver_binding[]_value
>

}]}
```

get (all)

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/rewritepolicy_binding

Query-parameters:

bulkbindings

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/rewritepolicy_binding?**bulkbindings=yes**

NITRO allows you to fetch bindings in bulk.

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "rewritepolicy_binding": [ {

"name":<String_value>,

"rewritepolicy_lbvserver_binding":<rewritepolicy_lbvserver_binding[]_value
>,

"rewritepolicy_rewritepolicylabel_binding":<rewritepolicy_rewritepolicylab
```

```
el_binding[]_value>,

"rewritepolicy_rewriteglobal_binding":<rewritepolicy_rewriteglobal_binding
[]_value>,

"rewritepolicy_csvserver_binding":<rewritepolicy_csvserver_binding[]_value
>

}]}
```

rewritepolicy_csvserver_binding

Binding object showing the csvserver that can be bound to rewritepolicy.

Properties

(click to see Operations)

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
boundto	<String>	Read-write	Location where policy is bound.
name	<String>	Read-write	Name of the rewrite policy.
priority	<Double>	Read-only	Specifies the priority of the policy.
labelname	<String>	Read-only	Name of the label to invoke if the current policy rule evaluates to TRUE.
gotopriorityexpression	<String>	Read-only	Expression specifying the priority of the next policy which will get evaluated if the current policy rule evaluates to TRUE.

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
labeltype	<String>	Read-only	Type of policy label invocation. Possible values = reqvserver, resvserver, policylabel
activepolicy	<Integer>	Read-only	Indicates whether policy is bound or not.
__count	<Double>	Read-only	count parameter

Operations

(click to see Properties)

GET| GET (ALL)| COUNT

Some options that you can use for each operations:

- **Getting warnings in response:**NITRO allows you to get warnings in an operation by specifying the "warning" query parameter as "yes". For example, to get warnings while connecting to the Citrix ADC appliance, the URL is as follows:

  http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/login?warning=yes

  If any, the warnings are displayed in the response payload with the HTTP code "209 X-NITRO-WARNING".

- **Authenticated access for individual NITRO operations:**NITRO allows you to logon to the Citrix ADC appliance to perform individual operations. You can use this option instead of creating a NITRO session (using the login object) and then using that session to perform all operations,

  To do this, you must specify the username and password in the request header of the NITRO request as follows:

  X-NITRO-USER:<username>

  X-NITRO-PASS:<password>

Note: In such cases, make sure that the request header DOES not include the following:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue>

`*Note:`*

Mandatory parameters are marked in redand placeholder content is marked in <green>.

get

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/rewritepolicy_csvserver_binding/name_value<String>

Query-parameters:

filter

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/rewritepolicy_csvserver_binding/name_value<String>?**filter=property-name1:property-value1,property-name2:property-value2**

Use this query-parameter to get the filtered set of rewritepolicy_csvserver_binding resources configured on Citrix ADC. Filtering can be done on any of the properties of the resource.

pagination

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/rewritepolicy_csvserver_binding/name_value<String>?**pagesize=#no;pageno=#no**

Use this query-parameter to get the rewritepolicy_csvserver_binding resources in chunks.

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "rewritepolicy_csvserver_binding": [ {

"boundto":<String_value>,

"name":<String_value>,

"priority":<Double_value>,

"labelname":<String_value>,

"gotopriorityexpression":<String_value>,

"labeltype":<String_value>,

"activepolicy":<Integer_value>

}]}
```

get (all)

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/rewritepolicy_csvserver_binding

Query-parameters:

bulkbindings

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/rewritepolicy_csvserver_binding?**bulkbindings=yes**

NITRO allows you to fetch bindings in bulk.

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "rewritepolicy_csvserver_binding": [ {

"boundto":<String_value>,

"name":<String_value>,

"priority":<Double_value>,

"labelname":<String_value>,

"gotopriorityexpression":<String_value>,

"labeltype":<String_value>,

"activepolicy":<Integer_value>

}]}
```

count

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/rewritepolicy_csvserver_binding/name_value<String>?**count=yes**

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{"rewritepolicy_csvserver_binding": [ { "__count": "#no"} ] }
```

rewritepolicy_lbvserver_binding

Binding object showing the lbvserver that can be bound to rewritepolicy.

Properties

(click to see Operations)

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
boundto	<String>	Read-write	Location where policy is bound.
name	<String>	Read-write	Name of the rewrite policy.
priority	<Double>	Read-only	Specifies the priority of the policy.
labelname	<String>	Read-only	Name of the label to invoke if the current policy rule evaluates to TRUE.
gotopriorityexpression	<String>	Read-only	Expression specifying the priority of the next policy which will get evaluated if the

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
			current policy rule evaluates to TRUE.
labeltype	<String>	Read-only	Type of policy label invocation. Possible values = reqvserver, resvserver, policylabel
activepolicy	<Integer>	Read-only	Indicates whether policy is bound or not.
__count	<Double>	Read-only	count parameter

Operations

(click to see Properties)

GET| GET (ALL)| COUNT

Some options that you can use for each operations:

- **Getting warnings in response:**NITRO allows you to get warnings in an operation by specifying the "warning" query parameter as "yes". For example, to get warnings while connecting to the Citrix ADC appliance, the URL is as follows:

  http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/login?warning=yes

  If any, the warnings are displayed in the response payload with the HTTP code "209 X-NITRO-WARNING".

- **Authenticated access for individual NITRO operations:**NITRO allows you to logon to the Citrix ADC appliance to perform individual operations. You can use this option instead of creating a NITRO session (using the login object) and then using that session to perform all operations,

  To do this, you must specify the username and password in the request header of the NITRO request as follows:

  X-NITRO-USER:<username>

X-NITRO-PASS:<password>

Note: In such cases, make sure that the request header DOES not include the following:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue>

`*Note:`*

Mandatory parameters are marked in redand placeholder content is marked in <green>.

get

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/rewritepolicy_lbvserver_binding/name_value<String>

Query-parameters:

filter

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/rewritepolicy_lbvserver_binding/name_value<String>?**filter=property-name1:property-value1,property-name2:property-value2**

Use this query-parameter to get the filtered set of rewritepolicy_lbvserver_binding resources configured on Citrix ADC. Filtering can be done on any of the properties of the resource.

pagination

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/rewritepolicy_lbvserver_binding/name_value<String>?**pagesize=#no;pageno=#no**

Use this query-parameter to get the rewritepolicy_lbvserver_binding resources in chunks.

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "rewritepolicy_lbvserver_binding": [ {

"boundto":<String_value>,

"name":<String_value>,

"priority":<Double_value>,

"labelname":<String_value>,

"gotopriorityexpression":<String_value>,

"labeltype":<String_value>,

"activepolicy":<Integer_value>

}]}
```

get (all)

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/rewritepolicy_lbvserver_binding

Query-parameters:

bulkbindings

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/rewritepolicy_lbvserver_binding?**bulkbindings=yes**

NITRO allows you to fetch bindings in bulk.

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "rewritepolicy_lbvserver_binding": [ {

"boundto":<String_value>,

"name":<String_value>,

"priority":<Double_value>,

"labelname":<String_value>,

"gotopriorityexpression":<String_value>,

"labeltype":<String_value>,

"activepolicy":<Integer_value>

}]}
```

count

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/rewritepolicy_lbvserver_binding/name_value<String>?**count=yes**

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{"rewritepolicy_lbvserver_binding": [ { "__count": "#no"} ] }
```

rewritepolicy_rewriteglobal_binding

Binding object showing the rewriteglobal that can be bound to rewritepolicy.

Properties

(click to see Operations)

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
boundto	<String>	Read-write	Location where policy is bound.
name	<String>	Read-write	Name of the rewrite policy.
priority	<Double>	Read-only	Specifies the priority of the policy.
labelname	<String>	Read-only	Name of the label to invoke if the current policy rule evaluates to TRUE.
gotopriorityexpression	<String>	Read-only	Expression specifying the priority of the next policy which will get evaluated if the

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
			current policy rule evaluates to TRUE.
labeltype	<String>	Read-only	Type of policy label invocation. Possible values = reqvserver, resvserver, policylabel
activepolicy	<Integer>	Read-only	Indicates whether policy is bound or not.
__count	<Double>	Read-only	count parameter

Operations

(click to see Properties)

GET| GET (ALL)| COUNT

Some options that you can use for each operations:

- **Getting warnings in response:**NITRO allows you to get warnings in an operation by specifying the "warning" query parameter as "yes". For example, to get warnings while connecting to the Citrix ADC appliance, the URL is as follows:

  http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/login?warning=yes

  If any, the warnings are displayed in the response payload with the HTTP code "209 X-NITRO-WARNING".

- **Authenticated access for individual NITRO operations:**NITRO allows you to logon to the Citrix ADC appliance to perform individual operations. You can use this option instead of creating a NITRO session (using the login object) and then using that session to perform all operations,

  To do this, you must specify the username and password in the request header of the NITRO request as follows:

  X-NITRO-USER:<username>

X-NITRO-PASS:<password>

Note: In such cases, make sure that the request header DOES not include the following:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue>

`*Note:`*

Mandatory parameters are marked in redand placeholder content is marked in <green>.

get

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/rewritepolicy_rewriteglobal_binding/name_value<String>

Query-parameters:

filter

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/rewritepolicy_rewriteglobal_binding/name_value<String>?**filter=property-name1:property-value1,property-name2:property-value2**

Use this query-parameter to get the filtered set of rewritepolicy_rewriteglobal_binding resources configured on Citrix ADC. Filtering can be done on any of the properties of the resource.

pagination

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/rewritepolicy_rewriteglobal_binding/name_value<String>?**pagesize=#no;pageno=#no**

Use this query-parameter to get the rewritepolicy_rewriteglobal_binding resources in chunks.

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "rewritepolicy_rewriteglobal_binding": [ {

"boundto":<String_value>,

"name":<String_value>,

"priority":<Double_value>,

"labelname":<String_value>,

"gotopriorityexpression":<String_value>,

"labeltype":<String_value>,

"activepolicy":<Integer_value>

}]}
```

get (all)

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/rewritepolicy_rewriteglobal_binding

Query-parameters:

bulkbindings

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/rewritepolicy_rewriteglobal_binding?**bulkbindings=yes**

NITRO allows you to fetch bindings in bulk.

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "rewritepolicy_rewriteglobal_binding": [ {

"boundto":<String_value>,

"name":<String_value>,

"priority":<Double_value>,

"labelname":<String_value>,

"gotopriorityexpression":<String_value>,

"labeltype":<String_value>,

"activepolicy":<Integer_value>

}]}
```

count

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/rewritepolicy_rewriteglobal_binding/name_value<String>?**count=yes**

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{"rewritepolicy_rewriteglobal_binding": [ { "__count": "#no"} ] }
```

rewritepolicy_rewritepolicylabel_binding

Binding object showing the rewritepolicylabel that can be bound to rewritepolicy.

Properties

(click to see Operations)

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
boundto	<String>	Read-write	Location where policy is bound.
name	<String>	Read-write	Name of the rewrite policy.
priority	<Double>	Read-only	Specifies the priority of the policy.
labelname	<String>	Read-only	Name of the label to invoke if the current policy rule evaluates to TRUE.
gotopriorityexpression	<String>	Read-only	Expression specifying the priority of the next policy which will get evaluated if the

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
			current policy rule evaluates to TRUE.
labeltype	<String>	Read-only	Type of policy label invocation. Possible values = reqvserver, resvserver, policylabel
activepolicy	<Integer>	Read-only	Indicates whether policy is bound or not.
__count	<Double>	Read-only	count parameter

Operations

(click to see Properties)

GET| GET (ALL)| COUNT

Some options that you can use for each operations:

- **Getting warnings in response:**NITRO allows you to get warnings in an operation by specifying the "warning" query parameter as "yes". For example, to get warnings while connecting to the Citrix ADC appliance, the URL is as follows:

  http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/login?warning=yes

  If any, the warnings are displayed in the response payload with the HTTP code "209 X-NITRO-WARNING".

- **Authenticated access for individual NITRO operations:**NITRO allows you to logon to the Citrix ADC appliance to perform individual operations. You can use this option instead of creating a NITRO session (using the login object) and then using that session to perform all operations,

  To do this, you must specify the username and password in the request header of the NITRO request as follows:

  X-NITRO-USER:<username>

X-NITRO-PASS:<password>

Note: In such cases, make sure that the request header DOES not include the following:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue>

`*Note:`*

Mandatory parameters are marked in redand placeholder content is marked in <green>.

get

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/rewritepolicy_rewritepolicylabel_binding/name_value<String>

Query-parameters:

filter

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/rewritepolicy_rewritepolicylabel_binding/name_value<String>?**filter=property-name1:property-value1,property-name2:property-value2**

Use this query-parameter to get the filtered set of rewritepolicy_rewritepolicylabel_binding resources configured on Citrix ADC. Filtering can be done on any of the properties of the resource.

pagination

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/rewritepolicy_rewritepolicylabel_binding/name_value<String>?**pagesize=#no;pageno=#no**

Use this query-parameter to get the rewritepolicy_rewritepolicylabel_binding resources in chunks.

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "rewritepolicy_rewritepolicylabel_binding": [ {

"boundto":<String_value>,

"name":<String_value>,

"priority":<Double_value>,

"labelname":<String_value>,

"gotopriorityexpression":<String_value>,

"labeltype":<String_value>,

"activepolicy":<Integer_value>

}]}
```

get (all)

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/rewritepolicy_rewritepolicylabel_binding

Query-parameters:

bulkbindings

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/rewritepolicy_rewritepolicylabel_binding?**bulkbindings=yes**

NITRO allows you to fetch bindings in bulk.

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "rewritepolicy_rewritepolicylabel_binding": [ {

"boundto":<String_value>,

"name":<String_value>,

"priority":<Double_value>,

"labelname":<String_value>,

"gotopriorityexpression":<String_value>,

"labeltype":<String_value>,

"activepolicy":<Integer_value>

}]}
```

count

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/rewritepolicy_rewritepolicylabel_binding/name_value<String>?**count=yes**

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{"rewritepolicy_rewritepolicylabel_binding": [ { "__count": "#no"} ] }
```

Rise

Rise

RISE feature related configuration.

NAME	DESCRIPTION
riseapbrsvc	Configuration for RISE Apbr services
riseparam	Configuration for RISE Parameter
riseprofile	Configuration for RISE profile
riseprofile_binding	Binding object showing the resources that can be bound to riseprofile
riseprofile_interface_binding	Binding object showing the interface that can be bound to riseprofile
riserhi	Configuration for RISE RHI rules

riseapbrsvc

Configuration for RISE Apbr services resource.

Properties

(click to see Operations)

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
name	<String>	Read-only	Name for the APBR service.
risesvctype	<String>	Read-only	Service or Service Group. Possible values = Service, ServiceGroup
serverip	<String>	Read-only	Server IP for APBR service.
nexthopip	<String>	Read-only	Nexthop IP for APBR service.
vlan	<Double>	Read-only	Vlan id for APBR service.
protocol	<String>	Read-only	Protocol type for APBR service. Possible values = HTTP, FTP, TCP, UDP, SSL, SSL_BRIDGE, SSL_TCP, DTLS, NNTP, RPCSVR, DNS, ADNS, SNMP, RTSP, DHCPRA, ANY, SIP_UDP, SIP_TCP, SIP_SSL, DNS_TCP, ADNS_TCP, MYSQL, MSSQL, ORACLE, RADIUS, RADIUSListener, RDP, DIAMETER, SSL_DIAMETER, TFTP, SMPP, PPTP,

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
			GRE, SYSLOGTCP, SYSLOGUDP, FIX, SSL_FIX, USER_TCP, USER_SSL_TCP, QUIC, IPFIX, LOGSTREAM
serverport	<Integer>	Read-only	Server port for APBR service. Range 1 - 65535 * in CLI is represented as 65535 in NITRO API
__count	<Double>	Read-only	count parameter

Operations

(click to see Properties)

GET (ALL)| COUNT

Some options that you can use for each operations:

- **Getting warnings in response:**NITRO allows you to get warnings in an operation by specifying the "warning" query parameter as "yes". For example, to get warnings while connecting to the Citrix ADC appliance, the URL is as follows:

  http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/login?warning=yes

  If any, the warnings are displayed in the response payload with the HTTP code "209 X-NITRO-WARNING".

- **Authenticated access for individual NITRO operations:**NITRO allows you to logon to the Citrix ADC appliance to perform individual operations. You can use this option instead of creating a NITRO session (using the login object) and then using that session to perform all operations,

To do this, you must specify the username and password in the request header of the NITRO request as follows:

X-NITRO-USER:<username>

X-NITRO-PASS:<password>

Note: In such cases, make sure that the request header DOES not include the following:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue>

`*Note:`*

Mandatory parameters are marked in redand placeholder content is marked in <green>.

get (all)

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/riseapbrsvc

Query-parameters:

attrs

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/riseapbrsvc?**attrs=property-name1,property-name2**

Use this query parameter to specify the resource details that you want to retrieve.

filter

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/riseapbrsvc?**filter=property-name1:property-val1,property-name2:property-val2**

Use this query-parameter to get the filtered set of riseapbrsvc resources configured on Citrix ADC. Filtering can be done on any of the properties of the resource.

view

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/riseapbrsvc?**view=summary**

Note: By default, the retrieved results are displayed in detail view (?view=detail).

pagination

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/riseapbrsvc?**pagesize=#no;pageno=#no**

Use this query-parameter to get the riseapbrsvc resources in chunks.

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "riseapbrsvc": [ {

"name":<String_value>,

"risesvctype":<String_value>,

"serverip":<String_value>,

"nexthopip":<String_value>,

"vlan":<Double_value>,

"protocol":<String_value>,

"serverport":<Integer_value>

}]}
```

count

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/riseapbrsvc?**count=yes**

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "riseapbrsvc": [ { "__count": "#no"} ] }
```

riseparam

Configuration for RISE Parameter resource.

Properties

(click to see Operations)

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
directmode	<String>	Read-write	RISE Direct attach mode. Default value: ENABLED Possible values = ENABLED, DISABLED
indirectmode	<String>	Read-write	RISE Indirect attach mode. Default value: DISABLED

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
			Possible values = ENABLED, DISABLED

Operations

(click to see Properties)

UPDATE| UNSET| GET (ALL)

Some options that you can use for each operations:

- **Getting warnings in response:**NITRO allows you to get warnings in an operation by specifying the "warning" query parameter as "yes". For example, to get warnings while connecting to the Citrix ADC appliance, the URL is as follows:

  http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/login?warning=yes

  If any, the warnings are displayed in the response payload with the HTTP code "209 X-NITRO-WARNING".

- **Authenticated access for individual NITRO operations:**NITRO allows you to logon to the Citrix ADC appliance to perform individual operations. You can use this option instead of creating a NITRO session (using the login object) and then using that session to perform all operations,

  To do this, you must specify the username and password in the request header of the NITRO request as follows:

  X-NITRO-USER:<username>

  X-NITRO-PASS:<password>

  Note: In such cases, make sure that the request header DOES not include the following:

  Cookie:NITRO_AUTH_TOKEN=<tokenvalue>

`*Note:` *

Mandatory parameters are marked in redand placeholder content is marked in <green>.

update

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/riseparam

HTTP Method:PUT

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Content-Type:application/json

Request Payload:

```
{"riseparam":{

"directmode":<String_value>,

"indirectmode":<String_value>

}}
```

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error

unset

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/riseparam?**action=unset**

HTTP Method:POST

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Content-Type:application/json

Request Payload:

```
{"riseparam":{
```

```
"directmode":true,

"indirectmode":true

}}
```

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error

get (all)

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/riseparam

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "riseparam": [ {

"directmode":<String_value>,

"indirectmode":<String_value>

}]}
```

riseprofile

Configuration for RISE profile resource.

Properties

(click to see Operations)

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
profilename	<String>	Read-write	Name of the RISE profile. Minimum length = 1 Maximum length = 83
servicename	<String>	Read-only	RISE Service name. Minimum length = 1 Maximum length = 31
deviceid	<String>	Read-only	Device id. Minimum length = 1 Maximum length = 31
slotid	<Double>	Read-only	Slot number of the RISE profile.
slotno	<Double>	Read-only	Slot number of the RISE profile.
vdcid	<Double>	Read-only	RISE vdc id.
vpcid	<Double>	Read-only	RISE vpc id.
ipaddress	<String>	Read-only	RISE ip address.

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
mode	<String>	Read-only	RISE attach mode. Possible values = Direct, Indirect, vPC-Direct
status	<String>	Read-only	RISE status. Possible values = Active, Inactive
vlan	<Double>	Read-only	RISE Vlan id.
vlangroupid	<Double>	Read-only	RISE Vlan Group id.
vlancfgstatus	<String>	Read-only	RISE config status. Possible values = Ok, Invalid
ifnum	<String>	Read-only	RISE Interface number.
issu	<String>	Read-only	RISE issu status. Possible values = InProgress, None
__count	<Double>	Read-only	count parameter

Operations

(click to see Properties)

GET (ALL)| GET| COUNT

Some options that you can use for each operations:

- **Getting warnings in response:**NITRO allows you to get warnings in an operation by specifying the "warning" query parameter as "yes". For example, to get warnings while connecting to the Citrix ADC appliance, the URL is as follows:

  http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/login?warning=yes

If any, the warnings are displayed in the response payload with the HTTP code "209 X-NITRO-WARNING".

- **Authenticated access for individual NITRO operations:**NITRO allows you to logon to the Citrix ADC appliance to perform individual operations. You can use this option instead of creating a NITRO session (using the login object) and then using that session to perform all operations,

  To do this, you must specify the username and password in the request header of the NITRO request as follows:

  X-NITRO-USER:<username>

  X-NITRO-PASS:<password>

  Note: In such cases, make sure that the request header DOES not include the following:

  Cookie:NITRO_AUTH_TOKEN=<tokenvalue>

`*Note:`*

Mandatory parameters are marked in redand placeholder content is marked in <green>.

get (all)

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/riseprofile

Query-parameters:

attrs

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/riseprofile?**attrs=property-name1,property-name2**

Use this query parameter to specify the resource details that you want to retrieve.

filter

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/riseprofile?**filter=property-name1:property-val1,property-name2:property-val2**

Use this query-parameter to get the filtered set of riseprofile resources configured on Citrix ADC. Filtering can be done on any of the properties of the resource.

view

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/riseprofile?**view=summary**

Note: By default, the retrieved results are displayed in detail view (?view=detail).

pagination

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/riseprofile?**pagesize=#no;pageno=#no**

Use this query-parameter to get the riseprofile resources in chunks.

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "riseprofile": [ {

"profilename":<String_value>,

"servicename":<String_value>,

"deviceid":<String_value>,

"slotid":<Double_value>,

"slotno":<Double_value>,

"vdcid":<Double_value>,

"vpcid":<Double_value>,

"ipaddress":<String_value>,
```

```
"mode":<String_value>,

"status":<String_value>,

"vlan":<Double_value>,

"vlangroupid":<Double_value>,

"vlancfgstatus":<String_value>,

"ifnum":<String_value>,

"issu":<String_value>

}]}
```

get

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/riseprofile/profilename_value<String>

Query-parameters:

attrs

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/riseprofile/profilename_value<String>?**attrs=property-name1,property-name2**

Use this query parameter to specify the resource details that you want to retrieve.

view

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/riseprofile/profilename_value<String>?**view=summary**

Note: By default, the retrieved results are displayed in detail view (?view=detail).

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "riseprofile": [ {

"profilename":<String_value>,

"servicename":<String_value>,

"deviceid":<String_value>,

"slotid":<Double_value>,

"slotno":<Double_value>,

"vdcid":<Double_value>,

"vpcid":<Double_value>,

"ipaddress":<String_value>,

"mode":<String_value>,

"status":<String_value>,

"vlan":<Double_value>,

"vlangroupid":<Double_value>,

"vlancfgstatus":<String_value>,

"ifnum":<String_value>,

"issu":<String_value>

}]}
```

count

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/riseprofile?**count=yes**

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "riseprofile": [ { "__count": "#no"} ] }
```

riseprofile_binding

Binding object which returns the resources bound to riseprofile.

Properties

(click to see Operations)

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
profilename	<String>	Read-write	Name of the RISE profile. Minimum length = 1 Maximum length = 83

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
riseprofile_interface_binding	<riseprofile_interface_binding[]>	Read-only	interface that can be bound to riseprofile.

Operations

(click to see Properties)

GET| GET (ALL)

Some options that you can use for each operations:

- **Getting warnings in response:**NITRO allows you to get warnings in an operation by specifying the "warning" query parameter as "yes". For example, to get warnings while connecting to the Citrix ADC appliance, the URL is as follows:

  http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/login?warning=yes

  If any, the warnings are displayed in the response payload with the HTTP code "209 X-NITRO-WARNING".

- **Authenticated access for individual NITRO operations:**NITRO allows you to logon to the Citrix ADC appliance to perform individual operations. You can use this option instead of creating a NITRO session (using the login object) and then using that session to perform all operations,

  To do this, you must specify the username and password in the request header of the NITRO request as follows:

  X-NITRO-USER:<username>

  X-NITRO-PASS:<password>

  Note: In such cases, make sure that the request header DOES not include the following:

  Cookie:NITRO_AUTH_TOKEN=<tokenvalue>

`*Note:`*

Mandatory parameters are marked in redand placeholder content is marked in <green>.

get

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/riseprofile_binding/profilename_value<String>

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "riseprofile_binding": [ {

"profilename":<String_value>,

"riseprofile_interface_binding":<riseprofile_interface_binding[]_value>

}]}
```

get (all)

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/riseprofile_binding

Query-parameters:

bulkbindings

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/riseprofile_binding?**bulkbindings=yes**

NITRO allows you to fetch bindings in bulk.

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "riseprofile_binding": [ {

"profilename":<String_value>,

"riseprofile_interface_binding":<riseprofile_interface_binding[]_value>

}]}
```

riseprofile_interface_binding

Binding object showing the interface that can be bound to riseprofile.

Properties

(click to see Operations)

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
memberinterface	<String>	Read-write	RISE profile member interfaces.
profilename	<String>	Read-write	Name of the RISE profile. Minimum length = 1

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
			Maximum length = 83
__count	<Double>	Read-write	count parameter

Operations

(click to see Properties)

GET| GET (ALL)| COUNT

Some options that you can use for each operations:

- **Getting warnings in response:**NITRO allows you to get warnings in an operation by specifying the "warning" query parameter as "yes". For example, to get warnings while connecting to the Citrix ADC appliance, the URL is as follows:

  http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/login?warning=yes

  If any, the warnings are displayed in the response payload with the HTTP code "209 X-NITRO-WARNING".

- **Authenticated access for individual NITRO operations:**NITRO allows you to logon to the Citrix ADC appliance to perform individual operations. You can use this option instead of creating a NITRO session (using the login object) and then using that session to perform all operations,

  To do this, you must specify the username and password in the request header of the NITRO request as follows:

  X-NITRO-USER:<username>

  X-NITRO-PASS:<password>

  Note: In such cases, make sure that the request header DOES not include the following:

  Cookie:NITRO_AUTH_TOKEN=<tokenvalue>

`*Note:`*

Mandatory parameters are marked in redand placeholder content is marked in <green>.

get

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/riseprofile_interface_binding/profilename_value<String>

Query-parameters:

filter

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/riseprofile_interface_binding/profilename_value<String>?**filter=property-name1:property-value1,property-name2:property-value2**

Use this query-parameter to get the filtered set of riseprofile_interface_binding resources configured on Citrix ADC. Filtering can be done on any of the properties of the resource.

pagination

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/riseprofile_interface_binding/profilename_value<String>?**pagesize=#no;pageno=#no**

Use this query-parameter to get the riseprofile_interface_binding resources in chunks.

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "riseprofile_interface_binding": [ {

"memberinterface":<String_value>,
```

```
"profilename":<String_value>

}]}
```

get (all)

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/riseprofile_interface_binding

Query-parameters:

bulkbindings

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/riseprofile_interface_binding?**bulkbindings=yes**

NITRO allows you to fetch bindings in bulk.

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "riseprofile_interface_binding": [ {

"memberinterface":<String_value>,

"profilename":<String_value>

}]}
```

count

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/riseprofile_interface_binding/profilename_value<String>?**count=yes**

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{"riseprofile_interface_binding": [ { "__count": "#no"} ] }
```

riserhi

Configuration for RISE RHI rules resource.

Properties

(click to see Operations)

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
ipaddress	<String>	Read-only	(V)IP advertised.
prefixlen	<Double>	Read-only	Prefix Length.
hostrtgw	<String>	Read-only	Gateway for the advertised IP.

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
nexthopvlan	<Double>	Read-only	Vlan id on which the gateway is present.
weight	<Double>	Read-only	Cost of the route.
vserverrhilevel	<String>	Read-only	Advertisement level. Default value: ONE_VSERVER Possible values = ONE_VSERVER, ALL_VSERVERS, NONE, VSVR_CNTRLD
__count	<Double>	Read-only	count parameter

Operations

(click to see Properties)

GET (ALL)| COUNT

Some options that you can use for each operations:

- **Getting warnings in response:**NITRO allows you to get warnings in an operation by specifying the "warning" query parameter as "yes". For example, to get warnings while connecting to the Citrix ADC appliance, the URL is as follows:

  http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/login?warning=yes

  If any, the warnings are displayed in the response payload with the HTTP code "209 X-NITRO-WARNING".

- **Authenticated access for individual NITRO operations:**NITRO allows you to logon to the Citrix ADC appliance to perform individual operations. You can use this option instead of creating a NITRO session (using the login object) and then using that session to perform all operations,

  To do this, you must specify the username and password in the request header of the NITRO request as follows:

X-NITRO-USER:<username>

X-NITRO-PASS:<password>

Note: In such cases, make sure that the request header DOES not include the following:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue>

`*Note:`*

Mandatory parameters are marked in redand placeholder content is marked in <green>.

get (all)

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/riserhi

Query-parameters:

attrs

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/riserhi?**attrs=property-name1,property-name2**

Use this query parameter to specify the resource details that you want to retrieve.

filter

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/riserhi?**filter=property-name1:property-val1,property-name2:property-val2**

Use this query-parameter to get the filtered set of riserhi resources configured on Citrix ADC. Filtering can be done on any of the properties of the resource.

view

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/riserhi?**view=summary**

Note: By default, the retrieved results are displayed in detail view (?view=detail).

pagination

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/riserhi?**pagesize=#no;pageno=#no**

Use this query-parameter to get the riserhi resources in chunks.

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "riserhi": [ {

"ipaddress":<String_value>,

"prefixlen":<Double_value>,

"hostrtgw":<String_value>,

"nexthopvlan":<Double_value>,

"weight":<Double_value>,

"vserverrhilevel":<String_value>

}]}
```

count

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/riserhi?**count=yes**

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "riserhi": [ { "__count": "#no"} ] }
```

Router

Router

Router configuration.

NAME	DESCRIPTION
routerdynamicrouting	Configuration for dynamic routing config

routerdynamicrouting

Configuration for dynamic routing config resource.

Properties

(click to see Operations)

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
commandstring	<String>	Read-write	command to be executed.
nodeid	<Double>	Read-write	Unique number that identifies the cluster node. Minimum value = 0

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
			Maximum value = 31
output	<String>	Read-only	command output.
__count	<Double>	Read-only	count parameter

Operations

(click to see Properties)

ADD| DELETE| UPDATE| UNSET| APPLY| GET (ALL)| COUNT

Some options that you can use for each operations:

- **Getting warnings in response:**NITRO allows you to get warnings in an operation by specifying the "warning" query parameter as "yes". For example, to get warnings while connecting to the Citrix ADC appliance, the URL is as follows:

  http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/login?warning=yes

  If any, the warnings are displayed in the response payload with the HTTP code "209 X-NITRO-WARNING".

- **Authenticated access for individual NITRO operations:**NITRO allows you to logon to the Citrix ADC appliance to perform individual operations. You can use this option instead of creating a NITRO session (using the login object) and then using that session to perform all operations,

  To do this, you must specify the username and password in the request header of the NITRO request as follows:

  X-NITRO-USER:<username>

  X-NITRO-PASS:<password>

  Note: In such cases, make sure that the request header DOES not include the following:

  Cookie:NITRO_AUTH_TOKEN=<tokenvalue>

`*Note:`*

Mandatory parameters are marked in redand placeholder content is marked in <green>.

add

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/routerdynamicrouting

HTTP Method:POST

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Content-Type:application/json

Request Payload:

```
{"routerdynamicrouting":{

"commandstring":<String_value>

}}
```

Response:

HTTP Status Code on Success: 201 Created HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error

delete

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/routerdynamicrouting

HTTP Method:DELETE

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue>

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error

update

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/routerdynamicrouting

HTTP Method:PUT

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Content-Type:application/json

Request Payload:

```
{"routerdynamicrouting":{

"commandstring":<String_value>

}}
```

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error

unset

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/routerdynamicrouting?**action=unset**

HTTP Method:POST

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Content-Type:application/json

Request Payload:

```
{"routerdynamicrouting":{

"commandstring":true

}}
```

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error

apply

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/routerdynamicrouting?**action=apply**

HTTP Method:POST

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Content-Type:application/json

Request Payload:

```
{"routerdynamicrouting":{

"commandstring":<String_value>

}}
```

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error

get (all)

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/routerdynamicrouting

Query-parameters:

args

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/routerdynamicrouting?**args=commandstring:<String_value>,nodeid:<Double_value>**

Use this query-parameter to get routerdynamicrouting resources based on additional properties.

attrs

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/routerdynamicrouting?**attrs=property-name1,property-name2**

Use this query parameter to specify the resource details that you want to retrieve.

filter

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/routerdynamicrouting?**filter=property-name1:property-val1,property-name2:property-val2**

Use this query-parameter to get the filtered set of routerdynamicrouting resources configured on Citrix ADC. Filtering can be done on any of the properties of the resource.

view

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/routerdynamicrouting?**view=summary**

Note: By default, the retrieved results are displayed in detail view (?view=detail).

pagination

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/routerdynamicrouting?**pagesize=#no;pageno=#no**

Use this query-parameter to get the routerdynamicrouting resources in chunks.

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "routerdynamicrouting": [ {

commandstring:<String_value>,nodeid:<Double_value>"output":<String_value>

}]}
```

count

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/routerdynamicrouting?**count=yes**

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "routerdynamicrouting": [ { "__count": "#no"} ] }
```

SMPP

Smpp

All the commands associated with SMPP.

NAME	DESCRIPTION
smppparam	Configuration for SMPP configuration parameters
smppuser	Configuration for SMPP user

smppparam

Configuration for SMPP configuration parameters resource.

Properties

(click to see Operations)

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
clientmode	<String>	Read-write	Mode in which the client binds to the ADC. Applicable settings function as follows: * TRANSCEIVER - Client can send and receive messages to and from the message center. * TRANSMITTERONLY - Client can only send messages. * RECEIVERONLY - Client can only receive messages. Default value: TRANSCEIVER Possible values = TRANSCEIVER,

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
			TRANSMITTERONLY, RECEIVERONLY
msgqueue	<String>	Read-write	Queue SMPP messages if a client that is capable of receiving the destination address messages is not available. Default value: OFF Possible values = ON, OFF
msgqueuesize	<Double>	Read-write	Maximum number of SMPP messages that can be queued. After the limit is reached, the Citrix ADC sends a deliver_sm_resp PDU, with an appropriate error message, to the message center. Default value: 10000
addrton	<Double>	Read-write	Type of Number, such as an international number or a national number, used in the ESME address sent in the bind request. Default value: 0

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
			Minimum value = 0 Maximum value = 256
addrnpi	<Double>	Read-write	Numbering Plan Indicator, such as landline, data, or WAP client, used in the ESME address sent in the bind request. Default value: 0 Minimum value = 0 Maximum value = 256
addrrange	<String>	Read-write	Set of SME addresses, sent in the bind request, serviced by the ESME. Default value: "\\d*"

Operations

(click to see Properties)

UPDATE| UNSET| GET (ALL)

Some options that you can use for each operations:

- **Getting warnings in response:**NITRO allows you to get warnings in an operation by specifying the "warning" query parameter as "yes". For example, to get warnings while connecting to the Citrix ADC appliance, the URL is as follows:

  http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/login?warning=yes

  If any, the warnings are displayed in the response payload with the HTTP code "209 X-NITRO-WARNING".

- **Authenticated access for individual NITRO operations:**NITRO allows you to logon to the Citrix ADC appliance to perform individual operations. You can use this option instead of creating a NITRO session (using the login object) and then using that session to perform all operations,

  To do this, you must specify the username and password in the request header of the NITRO request as follows:

  X-NITRO-USER:<username>

  X-NITRO-PASS:<password>

  Note: In such cases, make sure that the request header DOES not include the following:

  Cookie:NITRO_AUTH_TOKEN=<tokenvalue>

`*Note:`*

Mandatory parameters are marked in redand placeholder content is marked in <green>.

update

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/smppparam

HTTP Method:PUT

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Content-Type:application/json

Request Payload:

```
{"smppparam":{

"clientmode":<String_value>,

"msgqueue":<String_value>,

"msgqueuesize":<Double_value>,

"addrton":<Double_value>,

"addrnpi":<Double_value>,
```

```
"addrrange":<String_value>

}}
```

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error

unset

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/smppparam?**action=unset**

HTTP Method:POST

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Content-Type:application/json

Request Payload:

```
{"smppparam":{

"clientmode":true,

"msgqueue":true,

"msgqueuesize":true,

"addrton":true,

"addrnpi":true,

"addrrange":true

}}
```

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error

get (all)

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/smppparam

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "smppparam": [ {

"clientmode":<String_value>,

"msgqueue":<String_value>,

"msgqueuesize":<Double_value>,

"addrton":<Double_value>,

"addrnpi":<Double_value>,

"addrrange":<String_value>

}]}
```

smppuser

Configuration for SMPP user resource.

Properties

(click to see Operations)

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
username	<String>	Read-write	Name of the SMPP user. Must be the same as the user name specified in the SMPP server. Minimum length = 1
password	<String>	Read-write	Password for binding to the SMPP server. Must be the same as the password specified in the SMPP server. Minimum length = 1
__count	<Double>	Read-only	count parameter

Operations

(click to see Properties)

ADD| DELETE| UPDATE| GET (ALL)| GET| COUNT

Some options that you can use for each operations:

- **Getting warnings in response:**NITRO allows you to get warnings in an operation by specifying the "warning" query parameter as "yes". For example, to get warnings while connecting to the Citrix ADC appliance, the URL is as follows:

  http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/login?warning=yes

  If any, the warnings are displayed in the response payload with the HTTP code "209 X-NITRO-WARNING".

- **Authenticated access for individual NITRO operations:**NITRO allows you to logon to the Citrix ADC appliance to perform individual operations. You can use this option instead of creating a NITRO session (using the login object) and then using that session to perform all operations,

  To do this, you must specify the username and password in the request header of the NITRO request as follows:

  X-NITRO-USER:<username>

  X-NITRO-PASS:<password>

  Note: In such cases, make sure that the request header DOES not include the following:

  Cookie:NITRO_AUTH_TOKEN=<tokenvalue>

`*Note:`*

Mandatory parameters are marked in redand placeholder content is marked in <green>.

add

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/smppuser

HTTP Method:POST

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Content-Type:application/json

Request Payload:

```
{"smppuser":{

<b>"username":<String_value>,

</b>"password":<String_value>

}}
```

Response:

HTTP Status Code on Success: 201 Created HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error

delete

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/smppuser/username_value<String>

HTTP Method:DELETE

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue>

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error

update

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/smppuser

HTTP Method:PUT

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Content-Type:application/json

Request Payload:

```
{"smppuser":{

<b>"username":<String_value>,

</b>"password":<String_value>

}}
```

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error

get (all)

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/smppuser

Query-parameters:

attrs

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/smppuser?**attrs=property-name1,property-name2**

Use this query parameter to specify the resource details that you want to retrieve.

filter

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/smppuser?**filter=property-name1:property-val1,property-name2:property-val2**

Use this query-parameter to get the filtered set of smppuser resources configured on Citrix ADC. Filtering can be done on any of the properties of the resource.

view

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/smppuser?**view=summary**

Note: By default, the retrieved results are displayed in detail view (?view=detail).

pagination

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/smppuser?**pagesize=#no;pageno=#no**

Use this query-parameter to get the smppuser resources in chunks.

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "smppuser": [ {

"username":<String_value>,



}]}
```

get

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/smppuser/username_value<String>

Query-parameters:

attrs

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/smppuser/username_value<String>?**attrs=property-name1,property-name2**

Use this query parameter to specify the resource details that you want to retrieve.

view

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/smppuser/username_value<String>?**view=summary**

Note: By default, the retrieved results are displayed in detail view (?view=detail).

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "smppuser": [ {

"username":<String_value>,



}]}
```

count

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/smppuser?**count=yes**

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "smppuser": [ { "__count": "#no"} ] }
```

SNMP

SNMP

SNMP(Simple Network Management Protocol) configuration.

NAME	DESCRIPTION
snmpalarm	Configuration for alarm
snmpcommunity	Configuration for community
snmpengineid	Configuration for SNMP engine id
snmpgroup	Configuration for SNMP group
snmpmanager	Configuration for manager
snmpmib	Configuration for SNMP mib
snmpoid	Configuration for SNMP Object Identifier
snmpoption	Configuration for SNMP option
snmptrap	Configuration for snmp trap
snmptrap_binding	Binding object showing the resources that can be bound to snmptrap
snmptrap_snmpuser_binding	Binding object showing the snmpuser that can be bound to snmptrap
snmpuser	Configuration for SNMP user
snmpview	Configuration for view

snmpalarm

Configuration for alarm resource.

Properties

(click to see Operations)

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
trapname	<String>	Read-write	Name of the SNMP alarm. This parameter is required for identifying the SNMP alarm and cannot be modified. Possible values = CPU-USAGE, AVERAGE-CPU, MEMORY, MGMT-CPU-USAGE, SYNFLOOD, TCP-SYNFLOOD, VSERVER-REQRATE, SERVICE-REQRATE, ENTITY-RXRATE, ENTITY-TXRATE, ENTITY-SYNFLOOD, SERVICE-MAXCLIENTS, HA-STATE-CHANGE, ENTITY-STATE, CONFIG-CHANGE, CONFIG-SAVE, SERVICEGROUP-MEMBER-REQRATE, SERVICEGROUP-MEMBER-

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
			MAXCLIENTS, MONITOR-RTO-THRESHOLD, LOGIN-FAILURE, SSL-CERT-EXPIRY, FAN-SPEED-LOW, VOLTAGE-LOW, VOLTAGE-HIGH, TEMPERATURE-HIGH, CPU-TEMPERATURE-HIGH, POWER-SUPPLY-FAILURE, DISK-USAGE-HIGH, INTERFACE-THROUGHPUT-LOW, MON_PROBE_FAILED, HA-VERSION-MISMATCH, HA-SYNC-FAILURE, HA-NO-HEARTBEATS, HA-BAD-SECONDARY-STATE, INTERFACE-BW-USAGE, RATE-LIMIT-THRESHOLD-EXCEEDED, ENTITY-NAME-CHANGE, HA-PROP-FAILURE, IP-CONFLICT, PF-RL-RATE-THRESHOLD, PF-

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
			RL-PPS-THRESHOLD, PF-RL-RATE-PKTS-DROPPED, PF-RL-PPS-PKTS-DROPPED, APPFW-START-URL, APPFW-DENY-URL, APPFW-VIOLATIONS-TYPE, APPFW-REFERER-HEADER, APPFW-CSRF-TAG, APPFW-COOKIE, APPFW-FIELD-CONSISTENCY, APPFW-BUFFER-OVERFLOW, APPFW-FIELD-FORMAT, APPFW-SAFE-COMMERCE, APPFW-SAFE-OBJECT, APPFW-SESSION-LIMIT, APPFW-POLICY-HIT, APPFW-XSS, APPFW-XML-XSS, APPFW-SQL, APPFW-XML-SQL, APPFW-XML-ATTACHMENT, APPFW-XML-DOS, APPFW-XML-

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
			VALIDATION, APPFW-XML-WSI, APPFW-XML-SCHEMA-COMPILE, APPFW-XML-SOAP-FAULT, APPFW-NEW-SIGNATURE-ADDED, DNSKEY-EXPIRY, HA-LICENSE-MISMATCH, SSL-CARD-FAILED, SSL-CARD-NORMAL, WARM-RESTART-EVENT, HARD-DISK-DRIVE-ERRORS, COMPACT-FLASH-ERRORS, CALLHOME-UPLOAD-EVENT, 1024KEY-EXCHANGE-RATE, 2048KEY-EXCHANGE-RATE, 4096KEY-EXCHANGE-RATE, SSL-CUR-SESSION-INUSE, CLUSTER-NODE-HEALTH, CLUSTER-NODE-QUORUM, CLUSTER-VERSION-

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
			MISMATCH, CLUSTER-CCO-CHANGE, CLUSTER-OVS-CHANGE, CLUSTER-SYNC-FAILURE, CLUSTER-PROP-FAILURE, HA-STICKY-PRIMARY, INBAND-PROTOCOL-VERSION-MISMATCH, SSL-CHIP-REINIT, VRID-STATE-CHANGE, PORT-ALLOC-FAILED, LLDP-REMOTE-CHANGE, DUPLICATE-IPV6, PARTITION-CONFIG-EVENT, PARTITION-SWITCHED, LSN-PORTALLOC-FAILED, LSN-PORTQUOTA-EXCEED, LSN-SESSIONQUOTA-EXCEED, LSN-MEM-RECOVERY-KICKEDIN, VSERVER-SPILLOVER, PARTITION-RATE-

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
			LIMIT, POOLED-LICENSE-ONGRACE, POOLED-LICENSE-PARTIAL, CLUSTER-BACKPLANE-HB-MISSING, GSLB-SITE-MEP-FLAP, DNS-MAXNEGCACHE-USAGE, DNS-MAXCACHE-USAGE, NS-LICENSE-EXPIRY, PKT-RATELIMITING-ATTACK, GSLB-SYNC-STATUS-FLIP, URLFILT-DB-UPDATE-STATUS, URLFILT-INIT-SDK
thresholdvalue	<Double>	Read-write	Value for the high threshold. The Citrix ADC generates an SNMP trap message when the value of the attribute associated with the alarm is greater than or equal to the specified high threshold value. Minimum value = 1

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
normalvalue	<Double>	Read-write	Value for the normal threshold. A trap message is generated if the value of the respective attribute falls to or below this value after exceeding the high threshold. Minimum value = 1
time	<Double>	Read-write	Interval, in seconds, at which the Citrix ADC generates SNMP trap messages when the conditions specified in the SNMP alarm are met.Can be specified for the following alarms: SYNFLOOD, HA-VERSION-MISMATCH, HA-SYNC-FAILURE, HA-NO-HEARTBEATS,HA-BAD-SECONDARY-STATE, CLUSTER-NODE-HEALTH, CLUSTER-NODE-QUORUM, CLUSTER-VERSION-MISMATCH, CLUSTER-BKHB-FAILED, PORT-ALLOC-FAILED, COMPACT-FLASH-ERRORS, HARD-

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
			DISK-DRIVE-ERRORS and APPFW traps. Default trap time intervals: SYNFLOOD and APPFW traps = 1sec, PORT-ALLOC-FAILED = 3600sec(1 hour), Other Traps = 86400sec(1 day). Default value: 1
state	<String>	Read-write	Current state of the SNMP alarm. The Citrix ADC generates trap messages only for SNMP alarms that are enabled. Some alarms are enabled by default, but you can disable them. Default value: ENABLED Possible values = ENABLED, DISABLED
severity	<String>	Read-write	Severity level assigned to trap messages generated by this alarm. The severity levels are, in increasing order of severity, Informational, Warning, Minor, Major, and Critical.

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
			This parameter is useful when you want the Citrix ADC to send trap messages to a trap listener on the basis of severity level. Trap messages with a severity level lower than the specified level (in the trap listener entry) are not sent. Default value: Unknown Possible values = Critical, Major, Minor, Warning, Informational
logging	<String>	Read-write	Logging status of the alarm. When logging is enabled, the Citrix ADC logs every trap message that is generated for this alarm. Default value: ENABLED Possible values = ENABLED, DISABLED
timeout	<Double>	Read-only	If DB is enabled and clear config is fired, then to reset

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
			timeinterval of alarm, corresponding default time value is needed. This hidden argument holds the default time value for the corresponding alarm.
__count	<Double>	Read-only	count parameter

Operations

(click to see Properties)

UPDATE| UNSET| ENABLE| DISABLE| GET (ALL)| GET| COUNT

Some options that you can use for each operations:

- **Getting warnings in response:**NITRO allows you to get warnings in an operation by specifying the "warning" query parameter as "yes". For example, to get warnings while connecting to the Citrix ADC appliance, the URL is as follows:

  http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/login?warning=yes

  If any, the warnings are displayed in the response payload with the HTTP code "209 X-NITRO-WARNING".

- **Authenticated access for individual NITRO operations:**NITRO allows you to logon to the Citrix ADC appliance to perform individual operations. You can use this option instead of creating a NITRO session (using the login object) and then using that session to perform all operations,

  To do this, you must specify the username and password in the request header of the NITRO request as follows:

  X-NITRO-USER:<username>

  X-NITRO-PASS:<password>

  Note: In such cases, make sure that the request header DOES not include the following:

  Cookie:NITRO_AUTH_TOKEN=<tokenvalue>

`*Note:` *

Mandatory parameters are marked in redand placeholder content is marked in <green>.

update

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/snmpalarm

HTTP Method:PUT

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Content-Type:application/json

Request Payload:

```
{"snmpalarm":{

<b>"trapname":<String_value>,

</b>"thresholdvalue":<Double_value>,

"normalvalue":<Double_value>,

"time":<Double_value>,

"state":<String_value>,

"severity":<String_value>,

"logging":<String_value>

}}
```

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error

unset

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/snmpalarm?**action=unset**

HTTP Method:POST

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Content-Type:application/json

Request Payload:

```
{"snmpalarm":{

<b>"trapname":<String_value>,

</b>"thresholdvalue":true,

"normalvalue":true,

"time":true,

"state":true,

"severity":true,

"logging":true

}}
```

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error

enable

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/snmpalarm?**action=enable**

HTTP Method:POST

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Content-Type:application/json

Request Payload:

```
{"snmpalarm":{

<b>"trapname":<String_value>

</b>}}
```

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error

disable

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/snmpalarm?**action=disable**

HTTP Method:POST

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Content-Type:application/json

Request Payload:

```
{"snmpalarm":{

<b>"trapname":<String_value>

</b>}}
```

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error

get (all)

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/snmpalarm

Query-parameters:

attrs

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/snmpalarm?**attrs=property-name1,property-name2**

Use this query parameter to specify the resource details that you want to retrieve.

filter

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/snmpalarm?**filter=property-name1:property-val1,property-name2:property-val2**

Use this query-parameter to get the filtered set of snmpalarm resources configured on Citrix ADC. Filtering can be done on any of the properties of the resource.

view

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/snmpalarm?**view=summary**

Note: By default, the retrieved results are displayed in detail view (?view=detail).

pagination

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/snmpalarm?**pagesize=#no;pageno=#no**

Use this query-parameter to get the snmpalarm resources in chunks.

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "snmpalarm": [ {

"trapname":<String_value>,

"thresholdvalue":<Double_value>,

"normalvalue":<Double_value>,

"time":<Double_value>,

"state":<String_value>,

"severity":<String_value>,

"logging":<String_value>,

"timeout":<Double_value>

}]}
```

get

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/snmpalarm/trapname_value<String>

Query-parameters:

attrs

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/snmpalarm/trapname_value<String>?**attrs=property-name1,property-name2**

Use this query parameter to specify the resource details that you want to retrieve.

view

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/snmpalarm/trapname_value<String>?**view=summary**

Note: By default, the retrieved results are displayed in detail view (?view=detail).

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "snmpalarm": [ {

"trapname":<String_value>,

"thresholdvalue":<Double_value>,

"normalvalue":<Double_value>,

"time":<Double_value>,

"state":<String_value>,

"severity":<String_value>,

"logging":<String_value>,

"timeout":<Double_value>

}]}
```

count

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/snmpalarm?**count=yes**

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "snmpalarm": [ { "__count": "#no"} ] }
```

snmpcommunity

Configuration for community resource.

Properties

(click to see Operations)

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
communityname	<String>	Read-write	The SNMP community string. Can consist of 1 to 31 characters that include uppercase and lowercase

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
			letters,numbers and special characters. The following requirement applies only to the Citrix ADC CLI: If the string includes one or more spaces, enclose the name in double or single quotation marks (for example, "my string" or 'my string'). Minimum length = 1
permissions	<String>	Read-write	The SNMP V1 or V2 query-type privilege that you want to associate with this SNMP community. Possible values = GET, GET_NEXT, GET_BULK, SET, ALL
__count	<Double>	Read-only	count parameter

Operations

(click to see Properties)

ADD| DELETE| GET (ALL)| GET| COUNT

Some options that you can use for each operations:

- **Getting warnings in response:**NITRO allows you to get warnings in an operation by specifying the "warning" query parameter as "yes". For example, to get warnings while connecting to the Citrix ADC appliance, the URL is as follows:

  http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/login?warning=yes

  If any, the warnings are displayed in the response payload with the HTTP code "209 X-NITRO-WARNING".

- **Authenticated access for individual NITRO operations:**NITRO allows you to logon to the Citrix ADC appliance to perform individual operations. You can use this option instead of creating a NITRO session (using the login object) and then using that session to perform all operations,

  To do this, you must specify the username and password in the request header of the NITRO request as follows:

  X-NITRO-USER:<username>

  X-NITRO-PASS:<password>

  Note: In such cases, make sure that the request header DOES not include the following:

  Cookie:NITRO_AUTH_TOKEN=<tokenvalue>

`*Note:`*

Mandatory parameters are marked in redand placeholder content is marked in <green>.

add

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/snmpcommunity

HTTP Method:POST

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Content-Type:application/json

Request Payload:

```
{"snmpcommunity":{
```

```
<b>"communityname":<String_value>,

</b><b>"permissions":<String_value>

</b>}}
```

Response:

HTTP Status Code on Success: 201 Created HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error

delete

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/snmpcommunity/communityname_value<String>

HTTP Method:DELETE

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue>

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error

get (all)

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/snmpcommunity

Query-parameters:

attrs

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/snmpcommunity?**attrs=property-name1,property-name2**

Use this query parameter to specify the resource details that you want to retrieve.

filter

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/snmpcommunity?**filter=property-name1:property-val1,property-name2:property-val2**

Use this query-parameter to get the filtered set of snmpcommunity resources configured on Citrix ADC. Filtering can be done on any of the properties of the resource.

view

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/snmpcommunity?**view=summary**

Note: By default, the retrieved results are displayed in detail view (?view=detail).

pagination

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/snmpcommunity?**pagesize=#no;pageno=#no**

Use this query-parameter to get the snmpcommunity resources in chunks.

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "snmpcommunity": [ {

"communityname":<String_value>,

"permissions":<String_value>

}]}
```

get

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/snmpcommunity/communityname_value<String>

Query-parameters:

attrs

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/snmpcommunity/communityname_value<String>?**attrs=property-name1,property-name2**

Use this query parameter to specify the resource details that you want to retrieve.

view

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/snmpcommunity/communityname_value<String>?**view=summary**

Note: By default, the retrieved results are displayed in detail view (?view=detail).

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "snmpcommunity": [ {

"communityname":<String_value>,
```

```
"permissions":<String_value>

}]}
```

count

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/snmpcommunity?**count=yes**

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "snmpcommunity": [ { "__count": "#no"} ] }
```

snmpengineid

Configuration for SNMP engine id resource.

Properties

(click to see Operations)

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
engineid	<String>	Read-write	A hexadecimal value of at least 10 characters, uniquely

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
			identifying the engineid. Minimum length = 10 Maximum length = 31
ownernode	<Double>	Read-write	ID of the cluster node for which you are setting the engineid. Default value: -1 Minimum value = 0 Maximum value = 31
defaultengineid	<String>	Read-only	Unique identifier to assign to the SNMPv3 engine. Should be a hexadecimal value with a minimum length of 10 hex characters.
__count	<Double>	Read-only	count parameter

Operations

(click to see Properties)

UPDATE| UNSET| GET (ALL)| GET| COUNT

Some options that you can use for each operations:

- **Getting warnings in response:**NITRO allows you to get warnings in an operation by specifying the "warning" query parameter as "yes". For example, to get warnings while connecting to the Citrix ADC appliance, the URL is as follows:

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/login?warning=yes

If any, the warnings are displayed in the response payload with the HTTP code "209 X-NITRO-WARNING".

- **Authenticated access for individual NITRO operations:**NITRO allows you to logon to the Citrix ADC appliance to perform individual operations. You can use this option instead of creating a NITRO session (using the login object) and then using that session to perform all operations,

  To do this, you must specify the username and password in the request header of the NITRO request as follows:

  X-NITRO-USER:<username>

  X-NITRO-PASS:<password>

  Note: In such cases, make sure that the request header DOES not include the following:

  Cookie:NITRO_AUTH_TOKEN=<tokenvalue>

`*Note:`*

Mandatory parameters are marked in redand placeholder content is marked in <green>.

update

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/snmpengineid

HTTP Method:PUT

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Content-Type:application/json

Request Payload:

```
{"snmpengineid":{

<b>"engineid":<String_value>,

</b>"ownernode":<Double_value>

}}
```

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error

unset

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/snmpengineid?**action=unset**

HTTP Method:POST

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Content-Type:application/json

Request Payload:

```
{"snmpengineid":{

"ownernode":<Double_value>

}}
```

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error

get (all)

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/snmpengineid

Query-parameters:

attrs

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/snmpengineid?**attrs=property-name1,property-name2**

Use this query parameter to specify the resource details that you want to retrieve.

filter

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/snmpengineid?**filter=property-name1:property-val1,property-name2:property-val2**

Use this query-parameter to get the filtered set of snmpengineid resources configured on Citrix ADC. Filtering can be done on any of the properties of the resource.

view

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/snmpengineid?**view=summary**

Note: By default, the retrieved results are displayed in detail view (?view=detail).

pagination

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/snmpengineid?**pagesize=#no;pageno=#no**

Use this query-parameter to get the snmpengineid resources in chunks.

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "snmpengineid": [ {

"ownernode":<Double_value>,

"engineid":<String_value>,
```

```
"defaultengineid":<String_value>

}]}
```

get

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/snmpengineid/ownernode_value<Double>

Query-parameters:

attrs

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/snmpengineid/ownernode_value<Double>?**attrs=property-name1,property-name2**

Use this query parameter to specify the resource details that you want to retrieve.

view

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/snmpengineid/ownernode_value<Double>?**view=summary**

Note: By default, the retrieved results are displayed in detail view (?view=detail).

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "snmpengineid": [ {

"ownernode":<Double_value>,

"engineid":<String_value>,

"defaultengineid":<String_value>

}]}
```

count

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/snmpengineid?**count=yes**

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "snmpengineid": [ { "__count": "#no"} ] }
```

snmpgroup

Configuration for SNMP group resource.

Properties

(click to see Operations)

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
name	<String>	Read-write	Name for the SNMPv3 group. Can consist of 1 to 31 characters that include uppercase and lowercase letters, numbers, and the hyphen (-), period (.) pound (#), space ( ), at sign (@), equals (=), colon (:), and underscore (_) characters. You should choose a name that helps identify the SNMPv3 group. The following requirement applies only to the Citrix ADC CLI: If the name includes one or more spaces, enclose it in double or single quotation marks (for example, "my name" or 'my name'). Minimum length = 1
securitylevel	<String>	Read-write	Security level required for

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
			communication between the Citrix ADC and the SNMPv3 users who belong to the group. Specify one of the following options: noAuthNoPriv. Require neither authentication nor encryption. authNoPriv. Require authentication but no encryption. authPriv. Require authentication and encryption. Note: If you specify authentication, you must specify an encryption algorithm when you assign an SNMPv3 user to the group. If you also specify encryption, you must assign both an authentication and an encryption algorithm for each group member. Possible values = noAuthNoPriv,

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
			authNoPriv, authPriv
readviewname	<String>	Read-write	Name of the configured SNMPv3 view that you want to bind to this SNMPv3 group. An SNMPv3 user bound to this group can access the subtrees that are bound to this SNMPv3 view as type INCLUDED, but cannot access the ones that are type EXCLUDED. If the Citrix ADC has multiple SNMPv3 view entries with the same name, all such entries are associated with the SNMPv3 group. Minimum length = 1
storagetype	<String>	Read-only	The storage type for this group. Possible values = volatile, nonVolatile
status	<String>	Read-only	The status of this group. Possible values = active
__count	<Double>	Read-only	count parameter

Operations

(click to see Properties)

ADD| DELETE| UPDATE| GET (ALL)| COUNT

Some options that you can use for each operations:

- **Getting warnings in response:**NITRO allows you to get warnings in an operation by specifying the "warning" query parameter as "yes". For example, to get warnings while connecting to the Citrix ADC appliance, the URL is as follows:

  http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/login?warning=yes

  If any, the warnings are displayed in the response payload with the HTTP code "209 X-NITRO-WARNING".

- **Authenticated access for individual NITRO operations:**NITRO allows you to logon to the Citrix ADC appliance to perform individual operations. You can use this option instead of creating a NITRO session (using the login object) and then using that session to perform all operations,

  To do this, you must specify the username and password in the request header of the NITRO request as follows:

  X-NITRO-USER:<username>

  X-NITRO-PASS:<password>

  Note: In such cases, make sure that the request header DOES not include the following:

  Cookie:NITRO_AUTH_TOKEN=<tokenvalue>

`*Note:`*

Mandatory parameters are marked in redand placeholder content is marked in <green>.

add

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/snmpgroup

HTTP Method:POST

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Content-Type:application/json

Request Payload:

```
{"snmpgroup":{

<b>"name":<String_value>,

</b><b>"securitylevel":<String_value>,

</b><b>"readviewname":<String_value>

</b>}}
```

Response:

HTTP Status Code on Success: 201 Created HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error

delete

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/snmpgroup/name_value<String>

HTTP Method:DELETE

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue>

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error

update

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/snmpgroup

HTTP Method:PUT

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Content-Type:application/json

Request Payload:

```
{"snmpgroup":{

<b>"name":<String_value>,

</b><b>"securitylevel":<String_value>,

</b><b>"readviewname":<String_value>

</b>}}
```

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error

get (all)

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/snmpgroup

Query-parameters:

attrs

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/snmpgroup?**attrs=property-name1,property-name2**

Use this query parameter to specify the resource details that you want to retrieve.

filter

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/snmpgroup?**filter=property-name1:property-val1,property-name2:property-val2**

Use this query-parameter to get the filtered set of snmpgroup resources configured on Citrix ADC. Filtering can be done on any of the properties of the resource.

view

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/snmpgroup?**view=summary**

Note: By default, the retrieved results are displayed in detail view (?view=detail).

pagination

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/snmpgroup?**pagesize=#no;pageno=#no**

Use this query-parameter to get the snmpgroup resources in chunks.

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "snmpgroup": [ {

"name":<String_value>,

"securitylevel":<String_value>,

"readviewname":<String_value>,

"storagetype":<String_value>,

"status":<String_value>

}]}
```

count

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/snmpgroup?**count=yes**

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "snmpgroup": [ { "__count": "#no"} ] }
```

snmpmanager

Configuration for manager resource.

Properties

(click to see Operations)

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
ipaddress	<String>	Read-write	IP address of the SNMP manager. Can be an IPv4 or IPv6 address. You can instead specify an IPv4 network address or IPv6 network prefix if you want the Citrix ADC

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
			to respond to SNMP queries from any device on the specified network. Alternatively, instead of an IPv4 address, you can specify a host name that has been assigned to an SNMP manager. If you do so, you must add a DNS name server that resolves the host name of the SNMP manager to its IP address. Note: The Citrix ADC does not support host names for SNMP managers that have IPv6 addresses. Minimum length = 1 Maximum length = 255
netmask	<String>	Read-write	Subnet mask associated with an IPv4 network address. If the IP address specifies the address or host name of a specific host, accept the default value of 255.255.255.255.

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
domainresolveretry	<Integer>	Read-write	Amount of time, in seconds, for which the Citrix ADC waits before sending another DNS query to resolve the host name of the SNMP manager if the last query failed. This parameter is valid for host-name based SNMP managers only. After a query succeeds, the TTL determines the wait time. Default value: 5 Minimum value = 5 Maximum value = 20939
ip	<String>	Read-only	The resolved IP address of the hostname manager.
domain	<String>	Read-only	IP address of manager. It will be zero for hostname manager.
__count	<Double>	Read-only	count parameter

Operations

(click to see Properties)

ADD| DELETE| UPDATE| UNSET| GET (ALL)| COUNT

Some options that you can use for each operations:

- **Getting warnings in response:**NITRO allows you to get warnings in an operation by specifying the "warning" query parameter as "yes". For example, to get warnings while connecting to the Citrix ADC appliance, the URL is as follows:

  http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/login?warning=yes

  If any, the warnings are displayed in the response payload with the HTTP code "209 X-NITRO-WARNING".

- **Authenticated access for individual NITRO operations:**NITRO allows you to logon to the Citrix ADC appliance to perform individual operations. You can use this option instead of creating a NITRO session (using the login object) and then using that session to perform all operations,

  To do this, you must specify the username and password in the request header of the NITRO request as follows:

  X-NITRO-USER:<username>

  X-NITRO-PASS:<password>

  Note: In such cases, make sure that the request header DOES not include the following:

  Cookie:NITRO_AUTH_TOKEN=<tokenvalue>

`*Note:`*

Mandatory parameters are marked in redand placeholder content is marked in <green>.

add

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/snmpmanager

HTTP Method:POST

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Content-Type:application/json

Request Payload:

```
{"snmpmanager":{
```

```
<b>"ipaddress":<String_value>,

</b>"netmask":<String_value>,

"domainresolveretry":<Integer_value>

}}
```

Response:

HTTP Status Code on Success: 201 Created HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error

delete

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/snmpmanager/ipaddress_value<String>

HTTP Method:DELETE

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue>

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error

update

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/snmpmanager

HTTP Method:PUT

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Content-Type:application/json

Request Payload:

```
{"snmpmanager":{

<b>"ipaddress":<String_value>,

</b>"netmask":<String_value>,

"domainresolveretry":<Integer_value>

}}
```

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error

unset

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/snmpmanager?**action=unset**

HTTP Method:POST

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Content-Type:application/json

Request Payload:

```
{"snmpmanager":{

<b>"ipaddress":<String_value>,

</b>"netmask":<String_value>,

"domainresolveretry":true

}}
```

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error

get (all)

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/snmpmanager

Query-parameters:

attrs

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/snmpmanager?**attrs=property-name1,property-name2**

Use this query parameter to specify the resource details that you want to retrieve.

filter

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/snmpmanager?**filter=property-name1:property-val1,property-name2:property-val2**

Use this query-parameter to get the filtered set of snmpmanager resources configured on Citrix ADC. Filtering can be done on any of the properties of the resource.

view

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/snmpmanager?**view=summary**

Note: By default, the retrieved results are displayed in detail view (?view=detail).

pagination

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/snmpmanager?**pagesize=#no;pageno=#no**

Use this query-parameter to get the snmpmanager resources in chunks.

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "snmpmanager": [ {

"ipaddress":<String_value>,

"netmask":<String_value>,

"ip":<String_value>,

"domain":<String_value>,

"domainresolveretry":<Integer_value>

}]}
```

count

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/snmpmanager?**count=yes**

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "snmpmanager": [ { "__count": "#no"} ] }
```

snmpmib

Configuration for SNMP mib resource.

Properties

(click to see Operations)

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
contact	<String>	Read-write	Name of the administrator for this Citrix ADC. Along with the name, you can include information on how to contact this person, such as a phone number or an email address. Can consist of 1 to 127 characters that include uppercase and lowercase letters, numbers, and the hyphen (-), period (.) pound (#), space ( ), at sign (@), equals (=), colon (:), and underscore (_) characters. The following requirement applies

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
			only to the Citrix ADC CLI: If the information includes one or more spaces, enclose it in double or single quotation marks (for example, "my contact" or 'my contact'). Default value: "WebMaster (default)" Minimum length = 1
name	<String>	Read-write	Name for this Citrix ADC. Can consist of 1 to 127 characters that include uppercase and lowercase letters, numbers, and the hyphen (-), period (.) pound (#), space ( ), at sign (@), equals (=), colon (:), and underscore (_) characters. You should choose a name that helps identify the Citrix ADC appliance. The following requirement applies

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
			only to the Citrix ADC CLI: If the name includes one or more spaces, enclose it in double or single quotation marks (for example, "my name" or 'my name'). Default value: "Citrix ADC" Minimum length = 1
location	<String>	Read-write	Physical location of the Citrix ADC. For example, you can specify building name, lab number, and rack number. Can consist of 1 to 127 characters that include uppercase and lowercase letters, numbers, and the hyphen (-), period (.) pound (#), space ( ), at sign (@), equals (=), colon (:), and underscore (_) characters. The following requirement applies only to the Citrix ADC CLI:

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
			If the location includes one or more spaces, enclose it in double or single quotation marks (for example, "my location" or 'my location'). Default value: "POP (default)" Minimum length = 1
customid	<String>	Read-write	Custom identification number for the Citrix ADC. Can consist of 1 to 127 characters that include uppercase and lowercase letters, numbers, and the hyphen (-), period (.) pound (#), space ( ), at sign (@), equals (=), colon (:), and underscore (_) characters. You should choose a custom identification that helps identify the Citrix ADC appliance. The following requirement applies only to the Citrix ADC CLI: If the ID includes one or more spaces, enclose it in

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
			double or single quotation marks (for example, "my ID" or 'my ID'). Default value: "Default" Minimum length = 1
ownernode	<Double>	Read-write	ID of the cluster node for which we are setting the mib. This is a mandatory argument to set snmp mib on CLIP. Default value: -1 Minimum value = 0 Maximum value = 31
sysdesc	<String>	Read-only	The description of the system.
sysuptime	<Double>	Read-only	The UP time of the system in 100th of a second.
sysservices	<Double>	Read-only	The services offered by the system.
sysoid	<String>	Read-only	The OID of the system's management system.
__count	<Double>	Read-only	count parameter

Operations

(click to see Properties)

UPDATE| UNSET| GET (ALL)| GET| COUNT

Some options that you can use for each operations:

- **Getting warnings in response:**NITRO allows you to get warnings in an operation by specifying the "warning" query parameter as "yes". For example, to get warnings while connecting to the Citrix ADC appliance, the URL is as follows:

  http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/login?warning=yes

  If any, the warnings are displayed in the response payload with the HTTP code "209 X-NITRO-WARNING".

- **Authenticated access for individual NITRO operations:**NITRO allows you to logon to the Citrix ADC appliance to perform individual operations. You can use this option instead of creating a NITRO session (using the login object) and then using that session to perform all operations,

  To do this, you must specify the username and password in the request header of the NITRO request as follows:

  X-NITRO-USER:<username>

  X-NITRO-PASS:<password>

  Note: In such cases, make sure that the request header DOES not include the following:

  Cookie:NITRO_AUTH_TOKEN=<tokenvalue>

`*Note:`*

Mandatory parameters are marked in redand placeholder content is marked in <green>.

update

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/snmpmib

HTTP Method:PUT

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Content-Type:application/json

Request Payload:

```
{"snmpmib":{

"contact":<String_value>,

"name":<String_value>,

"location":<String_value>,

"customid":<String_value>,

"ownernode":<Double_value>

}}
```

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error

unset

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/snmpmib?**action=unset**

HTTP Method:POST

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Content-Type:application/json

Request Payload:

```
{"snmpmib":{

"contact":true,

"name":true,
```

```
"location":true,

"customid":true,

"ownernode":<Double_value>

}}
```

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error

get (all)

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/snmpmib

Query-parameters:

attrs

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/snmpmib?**attrs=property-name1,property-name2**

Use this query parameter to specify the resource details that you want to retrieve.

filter

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/snmpmib?**filter=property-name1:property-val1,property-name2:property-val2**

Use this query-parameter to get the filtered set of snmpmib resources configured on Citrix ADC. Filtering can be done on any of the properties of the resource.

view

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/snmpmib?**view=summary**

Note: By default, the retrieved results are displayed in detail view (?view=detail).

pagination

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/snmpmib?**pagesize=#no;pageno=#no**

Use this query-parameter to get the snmpmib resources in chunks.

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "snmpmib": [ {

"ownernode":<Double_value>,

"contact":<String_value>,

"name":<String_value>,

"location":<String_value>,

"sysdesc":<String_value>,

"sysuptime":<Double_value>,

"sysservices":<Double_value>,

"sysoid":<String_value>,

"customid":<String_value>

}]}
```

get

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/snmpmib/ownernode_value<Double>

Query-parameters:

attrs

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/snmpmib/ownernode_value<Double>?**attrs=property-name1,property-name2**

Use this query parameter to specify the resource details that you want to retrieve.

view

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/snmpmib/ownernode_value<Double>?**view=summary**

Note: By default, the retrieved results are displayed in detail view (?view=detail).

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "snmpmib": [ {

"ownernode":<Double_value>,

"contact":<String_value>,
```

```
"name":<String_value>,

"location":<String_value>,

"sysdesc":<String_value>,

"sysuptime":<Double_value>,

"sysservices":<Double_value>,

"sysoid":<String_value>,

"customid":<String_value>

}]}
```

count

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/snmpmib?**count=yes**

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "snmpmib": [ { "__count": "#no"} ] }
```

snmpoid

Configuration for SNMP Object Identifier resource.

Properties

(click to see Operations)

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
entitytype	<String>	Read-write	The type of entity whose SNMP OIDs you want to displayType of entity whose SNMP OIDs you want the Citrix ADC to display. Possible values = VSERVER, SERVICE, SERVICEGROUP
name	<String>	Read-write	Name of the entity whose SNMP OID you want the Citrix ADC to display. Minimum length = 1
Snmpoid	<String>	Read-only	The snmp oid.
__count	<Double>	Read-only	count parameter

Operations

(click to see Properties)

GET (ALL)| COUNT

Some options that you can use for each operations:

- **Getting warnings in response:**NITRO allows you to get warnings in an operation by specifying the "warning" query parameter as "yes". For example, to get warnings while connecting to the Citrix ADC appliance, the URL is as follows:

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/login?warning=yes

If any, the warnings are displayed in the response payload with the HTTP code "209 X-NITRO-WARNING".

- **Authenticated access for individual NITRO operations:**NITRO allows you to logon to the Citrix ADC appliance to perform individual operations. You can use this option instead of creating a NITRO session (using the login object) and then using that session to perform all operations,

  To do this, you must specify the username and password in the request header of the NITRO request as follows:

  X-NITRO-USER:<username>

  X-NITRO-PASS:<password>

  Note: In such cases, make sure that the request header DOES not include the following:

  Cookie:NITRO_AUTH_TOKEN=<tokenvalue>

`*Note:`*

Mandatory parameters are marked in redand placeholder content is marked in <green>.

get (all)

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/snmpoid

Query-parameters:

args

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/snmpoid?**args=entitytype:<String_value>,name:<String_value>**

Use this query-parameter to get snmpoid resources based on additional properties.

attrs

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/snmpoid?**attrs=property-name1,property-name2**

Use this query parameter to specify the resource details that you want to retrieve.

filter

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/snmpoid?**filter=property-name1:property-val1,property-name2:property-val2**

Use this query-parameter to get the filtered set of snmpoid resources configured on Citrix ADC. Filtering can be done on any of the properties of the resource.

view

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/snmpoid?**view=summary**

Note: By default, the retrieved results are displayed in detail view (?view=detail).

pagination

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/snmpoid?**pagesize=#no;pageno=#no**

Use this query-parameter to get the snmpoid resources in chunks.

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "snmpoid": [ {

<b>entitytype:<String_value>,</
b>name:<String_value>"Snmpoid":<String_value>

}]}
```

count

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/snmpoid?**args=entitytype:<String_value>,name:<String_value>;count=yes**

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "snmpoid": [ { "__count": "#no"} ] }
```

snmpoption

Configuration for SNMP option resource.

Properties

(click to see Operations)

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
snmpset	<String>	Read-write	Accept SNMP SET requests sent to the Citrix ADC, and allow SNMP managers to write values to MIB objects that are configured for write access.

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
			Default value: DISABLED Possible values = ENABLED, DISABLED
snmptraplogging	<String>	Read-write	Log any SNMP trap events (for SNMP alarms in which logging is enabled) even if no trap listeners are configured. With the default setting, SNMP trap events are logged if at least one trap listener is configured on the appliance. Default value: DISABLED Possible values = ENABLED, DISABLED
partitionnameintrap	<String>	Read-write	Send partition name as a varbind in traps. By default the partition names are not sent as a varbind. Default value: DISABLED Possible values = ENABLED, DISABLED

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
snmptraploggingl evel	<String>	Read-write	Audit log level of SNMP trap logs. The default value is INFORMATIONAL. Default value: INFORMATIONAL Possible values = EMERGENCY, ALERT, CRITICAL, ERROR, WARNING, NOTICE, INFORMATIONAL, DEBUG

Operations

(click to see Properties)

UPDATE| UNSET| GET (ALL)

Some options that you can use for each operations:

- **Getting warnings in response:**NITRO allows you to get warnings in an operation by specifying the "warning" query parameter as "yes". For example, to get warnings while connecting to the Citrix ADC appliance, the URL is as follows:

  http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/login?warning=yes

  If any, the warnings are displayed in the response payload with the HTTP code "209 X-NITRO-WARNING".

- **Authenticated access for individual NITRO operations:**NITRO allows you to logon to the Citrix ADC appliance to perform individual operations. You can use this option instead of creating a NITRO session (using the login object) and then using that session to perform all operations,

  To do this, you must specify the username and password in the request header of the NITRO request as follows:

X-NITRO-USER:<username>

X-NITRO-PASS:<password>

Note: In such cases, make sure that the request header DOES not include the following:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue>

`*Note:`*

Mandatory parameters are marked in redand placeholder content is marked in <green>.

update

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/snmpoption

HTTP Method:PUT

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Content-Type:application/json

Request Payload:

```
{"snmpoption":{

"snmpset":<String_value>,

"snmptraplogging":<String_value>,

"partitionnameintrap":<String_value>,

"snmptraploggginglevel":<String_value>

}}
```

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error

unset

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/snmpoption?**action=unset**

HTTP Method:POST

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Content-Type:application/json

Request Payload:

```
{"snmpoption":{

"snmpset":true,

"snmptraplogging":true,

"partitionnameintrap":true,

"snmptraploggginglevel":true

}}
```

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error

get (all)

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/snmpoption

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "snmpoption": [ {

"snmpset":<String_value>,

"snmptraplogging":<String_value>,

"partitionnameintrap":<String_value>,

"snmptraploggginglevel":<String_value>

}]}
```

snmptrap

Configuration for snmp trap resource.

Properties

(click to see Operations)

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
trapclass	<String>	Read-write	Type of trap messages that the Citrix ADC sends to the trap listener: Generic or the enterprise-specific messages defined in the MIB file.

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
			Possible values = generic, specific
trapdestination	<String>	Read-write	IPv4 or the IPv6 address of the trap listener to which the Citrix ADC is to send SNMP trap messages. Minimum length = 1
version	<String>	Read-write	SNMP version, which determines the format of trap messages sent to the trap listener. This setting must match the setting on the trap listener. Otherwise, the listener drops the trap messages. Default value: V2 Possible values = V1, V2, V3
td	<Double>	Read-write	Integer value that uniquely identifies the traffic domain in which you want to configure the entity. If you do not specify an ID, the entity becomes part of the default traffic domain, which has an ID of 0.

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
			Minimum value = 0 Maximum value = 4094
destport	<Integer>	Read-write	UDP port at which the trap listener listens for trap messages. This setting must match the setting on the trap listener. Otherwise, the listener drops the trap messages. Default value: 162 Minimum value = 1 Maximum value = 65534
communityname	<String>	Read-write	Password (string) sent with the trap messages, so that the trap listener can authenticate them. Can include 1 to 31 uppercase or lowercase letters, numbers, and hyphen (-), period (.) pound (#), space ( ), at (@), equals (=), colon (:), and underscore (_) characters. You must specify the same community string on the trap listener

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
			device. Otherwise, the trap listener drops the trap messages. The following requirement applies only to the Citrix ADC CLI: If the string includes one or more spaces, enclose the name in double or single quotation marks (for example, "my string" or 'my string').
srcip	<String>	Read-write	IPv4 or IPv6 address that the Citrix ADC inserts as the source IP address in all SNMP trap messages that it sends to this trap listener. By default this is the appliance's NSIP or NSIP6 address, but you can specify an IPv4 MIP or SNIP/ SNIP6 address. In cluster setup, the default value is the individual node's NSIP, but it can be set to CLIP or Striped SNIP address. In non

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
			default partition, this parameter must be set to the SNIP/ SNIP6 address. Minimum length = 1
severity	<String>	Read-write	Severity level at or above which the Citrix ADC sends trap messages to this trap listener. The severity levels, in increasing order of severity, are Informational, Warning, Minor, Major, Critical. This parameter can be set for trap listeners of type SPECIFIC only. The default is to send all levels of trap messages. Important: Trap messages are not assigned severity levels unless you specify severity levels when configuring SNMP alarms. Default value: Unknown Possible values = Critical, Major, Minor, Warning, Informational

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
allpartitions	<String>	Read-write	Send traps of all partitions to this destination. Default value: DISABLED Possible values = ENABLED, DISABLED
__count	<Double>	Read-only	count parameter

Operations

(click to see Properties)

ADD| DELETE| UPDATE| UNSET| GET (ALL)| COUNT

Some options that you can use for each operations:

- **Getting warnings in response:**NITRO allows you to get warnings in an operation by specifying the "warning" query parameter as "yes". For example, to get warnings while connecting to the Citrix ADC appliance, the URL is as follows:

  http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/login?warning=yes

  If any, the warnings are displayed in the response payload with the HTTP code "209 X-NITRO-WARNING".

- **Authenticated access for individual NITRO operations:**NITRO allows you to logon to the Citrix ADC appliance to perform individual operations. You can use this option instead of creating a NITRO session (using the login object) and then using that session to perform all operations,

  To do this, you must specify the username and password in the request header of the NITRO request as follows:

  X-NITRO-USER:<username>

  X-NITRO-PASS:<password>

  Note: In such cases, make sure that the request header DOES not include the following:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue>

`*Note:` *

Mandatory parameters are marked in redand placeholder content is marked in <green>.

add

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/snmptrap

HTTP Method:POST

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Content-Type:application/json

Request Payload:

```
{"snmptrap":{

<b>"trapclass":<String_value>,

</b><b>"trapdestination":<String_value>,

</b>"version":<String_value>,

"td":<Double_value>,

"destport":<Integer_value>,

"communityname":<String_value>,

"srcip":<String_value>,

"severity":<String_value>,

"allpartitions":<String_value>

}}
```

Response:

HTTP Status Code on Success: 201 Created HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error

delete

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/snmptrap/trapclass_value<String>

HTTP Method:DELETE

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue>

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error

update

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/snmptrap

HTTP Method:PUT

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Content-Type:application/json

Request Payload:

```
{"snmptrap":{

<b>"trapclass":<String_value>,

</b><b>"trapdestination":<String_value>,

</b>"version":<String_value>,

"td":<Double_value>,

"destport":<Integer_value>,
```

```
"communityname":<String_value>,

"srcip":<String_value>,

"severity":<String_value>,

"allpartitions":<String_value>

}}
```

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error

unset

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/snmptrap?**action=unset**

HTTP Method:POST

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Content-Type:application/json

Request Payload:

```
{"snmptrap":{

<b>"trapclass":<String_value>,

</b><b>"trapdestination":<String_value>,

</b>"version":<String_value>,

"td":<Double_value>,

"destport":true,
```

```
"communityname":true,

"srcip":true,

"severity":true,

"allpartitions":true

}}
```

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error

get (all)

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/snmptrap

Query-parameters:

attrs

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/snmptrap?**attrs=property-name1,property-name2**

Use this query parameter to specify the resource details that you want to retrieve.

filter

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/snmptrap?**filter=property-name1:property-val1,property-name2:property-val2**

Use this query-parameter to get the filtered set of snmptrap resources configured on Citrix ADC. Filtering can be done on any of the properties of the resource.

view

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/snmptrap?**view=summary**

Note: By default, the retrieved results are displayed in detail view (?view=detail).

pagination

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/snmptrap?**pagesize=#no;pageno=#no**

Use this query-parameter to get the snmptrap resources in chunks.

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "snmptrap": [ {

"trapclass":<String_value>,

"trapdestination":<String_value>,

"version":<String_value>,

"td":<Double_value>,

"destport":<Integer_value>,

"communityname":<String_value>,

"srcip":<String_value>,

"severity":<String_value>,

"allpartitions":<String_value>
```

```
}]}
```

count

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/snmptrap?**count=yes**

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "snmptrap": [ { "__count": "#no"} ] }
```

snmptrap_binding

Binding object which returns the resources bound to snmptrap.

Properties

(click to see Operations)

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
trapclass	<String>	Read-write	Trap type specified in the trap listener entry.

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
			Possible values = generic, specific
trapdestination	<String>	Read-write	IP address specified in the trap listener entry. Minimum length = 1
version	<String>	Read-write	The SNMP version of the trap specified in the trap listener entry. Possible values = V1, V2, V3
td	<Double>	Read-write	Integer value that uniquely identifies the traffic domain in which you want to configure the entity. If you do not specify an ID, the entity becomes part of the default traffic domain, which has an ID of 0. Minimum value = 0 Maximum value = 4094
snmptrap_snmpuser_binding	<snmptrap_snmpuser_binding[]>	Read-only	snmpuser that can be bound to snmptrap.

Operations

(click to see Properties)

GET| GET (ALL)

Some options that you can use for each operations:

- **Getting warnings in response:**NITRO allows you to get warnings in an operation by specifying the "warning" query parameter as "yes". For example, to get warnings while connecting to the Citrix ADC appliance, the URL is as follows:

  http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/login?warning=yes

  If any, the warnings are displayed in the response payload with the HTTP code "209 X-NITRO-WARNING".

- **Authenticated access for individual NITRO operations:**NITRO allows you to logon to the Citrix ADC appliance to perform individual operations. You can use this option instead of creating a NITRO session (using the login object) and then using that session to perform all operations,

  To do this, you must specify the username and password in the request header of the NITRO request as follows:

  X-NITRO-USER:<username>

  X-NITRO-PASS:<password>

  Note: In such cases, make sure that the request header DOES not include the following:

  Cookie:NITRO_AUTH_TOKEN=<tokenvalue>

`*Note:`*

Mandatory parameters are marked in redand placeholder content is marked in <green>.

get

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/snmptrap_binding

Query-parameters:

args

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/snmptrap_binding?
args=trapdestination:<String_value>,version:<String_value>,td:<Double_value>,trapclass:<String_valu

Use this query-parameter to get snmptrap_binding resources based on additional properties.

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "snmptrap_binding": [ {

"trapdestination":<String_value>,

"version":<String_value>,

"td":<Double_value>,

"trapclass":<String_value>,

"snmptrap_snmpuser_binding":<snmptrap_snmpuser_binding[]_value>

}]}
```

get (all)

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/snmptrap_binding

Query-parameters:

bulkbindings

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/snmptrap_binding?**bulkbindings=yes**

NITRO allows you to fetch bindings in bulk.

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "snmptrap_binding": [ {

"trapdestination":<String_value>,

"version":<String_value>,

"td":<Double_value>,

"trapclass":<String_value>,

"snmptrap_snmpuser_binding":<snmptrap_snmpuser_binding[]_value>

}]}
```

snmptrap_snmpuser_binding

Binding object showing the snmpuser that can be bound to snmptrap.

Properties

(click to see Operations)

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
trapclass	<String>	Read-write	Type of trap messages that the Citrix ADC sends to

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
			the trap listener: Generic or the enterprise-specific messages defined in the MIB file. Possible values = generic, specific
securitylevel	<String>	Read-write	Security level of the SNMPv3 trap. Default value: authNoPriv, Possible values = noAuthNoPriv, authNoPriv, authPriv
username	<String>	Read-write	Name of the SNMP user that will send the SNMPv3 traps.
td	<Double>	Read-write	Integer value that uniquely identifies the traffic domain in which you want to configure the entity. If you do not specify an ID, the entity becomes part of the default traffic domain, which has an ID of 0. Minimum value = 0 Maximum value = 4094
trapdestination	<String>	Read-write	IPv4 or the IPv6 address of the trap listener to which the

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
			Citrix ADC is to send SNMP trap messages. Minimum length = 1
version	<String>	Read-write	SNMP version, which determines the format of trap messages sent to the trap listener. This setting must match the setting on the trap listener. Otherwise, the listener drops the trap messages. Default value: V3 Possible values = V1, V2, V3
__count	<Double>	Read-write	count parameter

Operations

(click to see Properties)

ADD:| DELETE:| GET| GET (ALL)| COUNT

Some options that you can use for each operations:

- **Getting warnings in response:**NITRO allows you to get warnings in an operation by specifying the "warning" query parameter as "yes". For example, to get warnings while connecting to the Citrix ADC appliance, the URL is as follows:

  http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/login?warning=yes

  If any, the warnings are displayed in the response payload with the HTTP code "209 X-NITRO-WARNING".

- **Authenticated access for individual NITRO operations:**NITRO allows you to logon to the Citrix ADC appliance to perform individual operations. You can use this option instead of creating a NITRO session (using the login object) and then using that session to perform all operations,

  To do this, you must specify the username and password in the request header of the NITRO request as follows:

  X-NITRO-USER:<username>

  X-NITRO-PASS:<password>

  Note: In such cases, make sure that the request header DOES not include the following:

  Cookie:NITRO_AUTH_TOKEN=<tokenvalue>

`*Note:`*

Mandatory parameters are marked in redand placeholder content is marked in <green>.

add

URL:http://<Citrix-ADC-IP-address(NSIP)/nitro/v1/config/snmptrap_snmpuser_binding

HTTP Method:PUT

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Content-Type:application/json

Request Payload:

```
{

"snmptrap_snmpuser_binding":{

<b>"trapclass":<String_value>,

</b><b>"trapdestination":<String_value>,

</b>"td":<Double_value>,

"version":<String_value>,
```

```
<b>"username":<String_value>,

</b>"securitylevel":<String_value>

}}
```

Response:

HTTP Status Code on Success: 201 Created HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error

delete

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/snmptrap_snmpuser_binding/trapclass_value<String>

Query-parameters:

args

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/snmptrap_snmpuser_binding/trapclass_value<String>?**args=trapdestination:<String_value>,td:<Double_value>,version:<String_value>,username:<String_valu**

HTTP Method:DELETE

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue>

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error

get

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/snmptrap_snmpuser_binding

Query-parameters:

args

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/snmptrap_snmpuser_binding?**args=trapdestination:<String_value>,version:<String_value>,td:<Double_value>,trapclass:<String_valu**

Use this query-parameter to get snmptrap_snmpuser_binding resources based on additional properties.

filter

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/snmptrap_snmpuser_binding?**filter=property-name1:property-value1,property-name2:property-value2**

Use this query-parameter to get the filtered set of snmptrap_snmpuser_binding resources configured on Citrix ADC. Filtering can be done on any of the properties of the resource.

pagination

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/snmptrap_snmpuser_binding?**pagesize=#no;pageno=#no**

Use this query-parameter to get the snmptrap_snmpuser_binding resources in chunks.

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "snmptrap_snmpuser_binding": [ {

"trapclass":<String_value>,
```

```
"securitylevel":<String_value>,

"username":<String_value>,

"td":<Double_value>,

"trapdestination":<String_value>,

"version":<String_value>

}]}
```

get (all)

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/snmptrap_snmpuser_binding

Query-parameters:

bulkbindings

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/snmptrap_snmpuser_binding?**bulkbindings=yes**

NITRO allows you to fetch bindings in bulk.

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "snmptrap_snmpuser_binding": [ {

"trapclass":<String_value>,

"securitylevel":<String_value>,

"username":<String_value>,

"td":<Double_value>,

"trapdestination":<String_value>,

"version":<String_value>

}]}
```

count

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/snmptrap_snmpuser_binding?**args=trapdestination:<String_value>,version:<String_value>,td:<Double_value>,trapclass:<String_valu**

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "snmptrap_snmpuser_binding": [ { "__count": "#no"} ] }
```

snmpuser

Configuration for SNMP user resource.

Properties

(click to see Operations)

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
name	<String>	Read-write	Name for the SNMPv3 user. Can consist of 1 to 31 characters that include uppercase and lowercase letters, numbers, and the hyphen (-), period (.) pound (#), space ( ), at sign (@), equals (=), colon (:), and underscore (_) characters. The following requirement applies only to the Citrix ADC CLI: If the name includes one or more spaces, enclose it in double or single quotation marks (for example, "my user" or 'my user'). Minimum length = 1

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
group	<String>	Read-write	Name of the configured SNMPv3 group to which to bind this SNMPv3 user. The access rights (bound SNMPv3 views) and security level set for this group are assigned to this user. Minimum length = 1
authtype	<String>	Read-write	Authentication algorithm used by the Citrix ADC and the SNMPv3 user for authenticating the communication between them. You must specify the same authentication algorithm when you configure the SNMPv3 user in the SNMP manager. Possible values = MD5, SHA
authpasswd	<String>	Read-write	Plain-text pass phrase to be used by the authentication algorithm specified by the authType (Authentication Type) parameter. Can consist of 1 to 31 characters that include uppercase and lowercase letters, numbers, and

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
			the hyphen (-), period (.) pound (#), space ( ), at sign (@), equals (=), colon (:), and underscore (_) characters. The following requirement applies only to the Citrix ADC CLI: If the pass phrase includes one or more spaces, enclose it in double or single quotation marks (for example, "my phrase" or 'my phrase'). Minimum length = 8
privtype	<String>	Read-write	Encryption algorithm used by the Citrix ADC and the SNMPv3 user for encrypting the communication between them. You must specify the same encryption algorithm when you configure the SNMPv3 user in the SNMP manager. Possible values = DES, AES

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
privpasswd	<String>	Read-write	Encryption key to be used by the encryption algorithm specified by the privType (Encryption Type) parameter. Can consist of 1 to 31 characters that include uppercase and lowercase letters, numbers, and the hyphen (-), period (.) pound (#), space ( ), at sign (@), equals (=), colon (:), and underscore (_) characters. The following requirement applies only to the Citrix ADC CLI: If the key includes one or more spaces, enclose it in double or single quotation marks (for example, "my key" or 'my key'). Minimum length = 8
engineid	<String>	Read-only	The context engine ID of the user.
storagetype	<String>	Read-only	The storage type for this user.

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
			Possible values = volatile, nonVolatile
status	<String>	Read-only	The status of this user. Possible values = active
__count	<Double>	Read-only	count parameter

Operations

(click to see Properties)

ADD| DELETE| UPDATE| UNSET| GET (ALL)| GET| COUNT

Some options that you can use for each operations:

- **Getting warnings in response:**NITRO allows you to get warnings in an operation by specifying the "warning" query parameter as "yes". For example, to get warnings while connecting to the Citrix ADC appliance, the URL is as follows:

  http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/login?warning=yes

  If any, the warnings are displayed in the response payload with the HTTP code "209 X-NITRO-WARNING".

- **Authenticated access for individual NITRO operations:**NITRO allows you to logon to the Citrix ADC appliance to perform individual operations. You can use this option instead of creating a NITRO session (using the login object) and then using that session to perform all operations,

  To do this, you must specify the username and password in the request header of the NITRO request as follows:

  X-NITRO-USER:<username>

  X-NITRO-PASS:<password>

  Note: In such cases, make sure that the request header DOES not include the following:

  Cookie:NITRO_AUTH_TOKEN=<tokenvalue>

`*Note:` *

Mandatory parameters are marked in redand placeholder content is marked in <green>.

add

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/snmpuser

HTTP Method:POST

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Content-Type:application/json

Request Payload:

```
{"snmpuser":{

<b>"name":<String_value>,

</b><b>"group":<String_value>,

</b>"authtype":<String_value>,

"authpasswd":<String_value>,

"privtype":<String_value>,

"privpasswd":<String_value>

}}
```

Response:

HTTP Status Code on Success: 201 Created HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error

delete

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/snmpuser/name_value<String>

HTTP Method:DELETE

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue>

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error

update

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/snmpuser

HTTP Method:PUT

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Content-Type:application/json

Request Payload:

```
{"snmpuser":{

<b>"name":<String_value>,

</b>"group":<String_value>,

"authtype":<String_value>,

"authpasswd":<String_value>,

"privtype":<String_value>,

"privpasswd":<String_value>

}}
```

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error

unset

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/snmpuser?**action=unset**

HTTP Method:POST

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Content-Type:application/json

Request Payload:

```
{"snmpuser":{

<b>"name":<String_value>,

</b>"authtype":true,

"privtype":true,

"authpasswd":true,

"privpasswd":true

}}
```

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error

get (all)

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/snmpuser

Query-parameters:

attrs

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/snmpuser?**attrs=property-name1,property-name2**

Use this query parameter to specify the resource details that you want to retrieve.

filter

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/snmpuser?**filter=property-name1:property-val1,property-name2:property-val2**

Use this query-parameter to get the filtered set of snmpuser resources configured on Citrix ADC. Filtering can be done on any of the properties of the resource.

view

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/snmpuser?**view=summary**

Note: By default, the retrieved results are displayed in detail view (?view=detail).

pagination

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/snmpuser?**pagesize=#no;pageno=#no**

Use this query-parameter to get the snmpuser resources in chunks.

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "snmpuser": [ {

"name":<String_value>,

"group":<String_value>,

"authtype":<String_value>,

"privtype":<String_value>,

"engineid":<String_value>,

"storagetype":<String_value>,

"status":<String_value>

}]}
```

get

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/snmpuser/name_value<String>

Query-parameters:

attrs

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/snmpuser/name_value<String>?**attrs=property-name1,property-name2**

Use this query parameter to specify the resource details that you want to retrieve.

view

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/snmpuser/name_value<String>?**view=summary**

Note: By default, the retrieved results are displayed in detail view (?view=detail).

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "snmpuser": [ {

"name":<String_value>,

"group":<String_value>,

"authtype":<String_value>,

"privtype":<String_value>,

"engineid":<String_value>,

"storagetype":<String_value>,

"status":<String_value>

}]}
```

count

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/snmpuser?**count=yes**

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "snmpuser": [ { "__count": "#no"} ] }
```

snmpview

Configuration for view resource.

Properties

(click to see Operations)

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
name	<String>	Read-write	Name for the SNMPv3 view. Can consist of 1 to 31 characters that include uppercase and lowercase letters, numbers, and the hyphen (-), period (.) pound (#), space ( ), at sign (@), equals (=), colon (:), and underscore (_) characters. You should choose a name that helps identify the SNMPv3 view.

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
			The following requirement applies only to the Citrix ADC CLI: If the name includes one or more spaces, enclose it in double or single quotation marks (for example, "my view" or 'my view'). Minimum length = 1
subtree	<String>	Read-write	A particular branch (subtree) of the MIB tree that you want to associate with this SNMPv3 view. You must specify the subtree as an SNMP OID. Minimum length = 1
type	<String>	Read-write	Include or exclude the subtree, specified by the subtree parameter, in or from this view. This setting can be useful when you have included a subtree, such as A, in an SNMPv3 view and you want to exclude a specific subtree of A, such as B, from the SNMPv3 view.

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
			Possible values = included, excluded
storagetype	<String>	Read-only	The storage type for this view. Possible values = volatile, nonVolatile
status	<String>	Read-only	The status of this view. Possible values = active
__count	<Double>	Read-only	count parameter

Operations

(click to see Properties)

ADD| DELETE| UPDATE| GET (ALL)| COUNT

Some options that you can use for each operations:

- **Getting warnings in response:**NITRO allows you to get warnings in an operation by specifying the "warning" query parameter as "yes". For example, to get warnings while connecting to the Citrix ADC appliance, the URL is as follows:

  http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/login?warning=yes

  If any, the warnings are displayed in the response payload with the HTTP code "209 X-NITRO-WARNING".

- **Authenticated access for individual NITRO operations:**NITRO allows you to logon to the Citrix ADC appliance to perform individual operations. You can use this option instead of creating a NITRO session (using the login object) and then using that session to perform all operations,

  To do this, you must specify the username and password in the request header of the NITRO request as follows:

  X-NITRO-USER:<username>

X-NITRO-PASS:<password>

Note: In such cases, make sure that the request header DOES not include the following:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue>

`*Note:`*

Mandatory parameters are marked in redand placeholder content is marked in <green>.

add

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/snmpview

HTTP Method:POST

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Content-Type:application/json

Request Payload:

```
{"snmpview":{

<b>"name":<String_value>,

</b><b>"subtree":<String_value>,

</b><b>"type":<String_value>

</b>}}
```

Response:

HTTP Status Code on Success: 201 Created HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error

delete

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/snmpview/name_value<String>

HTTP Method:DELETE

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue>

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error

update

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/snmpview

HTTP Method:PUT

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Content-Type:application/json

Request Payload:

```
{"snmpview":{

<b>"name":<String_value>,

</b><b>"subtree":<String_value>,

</b><b>"type":<String_value>

</b>}}
```

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error

get (all)

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/snmpview

Query-parameters:

attrs

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/snmpview?**attrs=property-name1,property-name2**

Use this query parameter to specify the resource details that you want to retrieve.

filter

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/snmpview?**filter=property-name1:property-val1,property-name2:property-val2**

Use this query-parameter to get the filtered set of snmpview resources configured on Citrix ADC. Filtering can be done on any of the properties of the resource.

view

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/snmpview?**view=summary**

Note: By default, the retrieved results are displayed in detail view (?view=detail).

pagination

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/snmpview?**pagesize=#no;pageno=#no**

Use this query-parameter to get the snmpview resources in chunks.

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "snmpview": [ {

"name":<String_value>,

"subtree":<String_value>,

"type":<String_value>,

"storagetype":<String_value>,

"status":<String_value>

}]}
```

count

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/snmpview?**count=yes**

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "snmpview": [ { "__count": "#no"} ] }
```

Spillover

Spillover

Spillover policies and actions.

NAME	DESCRIPTION
spilloveraction	Configuration for Spillover action
spilloverpolicy	Configuration for Spillover policy
spilloverpolicy_binding	Binding object showing the resources that can be bound to spilloverpolicy
spilloverpolicy_csvserver_binding	Binding object showing the csvserver that can be bound to spilloverpolicy
spilloverpolicy_gslbvserver_binding	Binding object showing the gslbvserver that can be bound to spilloverpolicy
spilloverpolicy_lbvserver_binding	Binding object showing the lbvserver that can be bound to spilloverpolicy

spilloveraction

Configuration for Spillover action resource.

Properties

(click to see Operations)

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
name	<String>	Read-write	Name of the spillover action.
action	<String>	Read-write	Spillover action. Currently only type SPILLOVER is supported. Possible values = SPILLOVER

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
newname	<String>	Read-write	New name for the spillover action. Must begin with an ASCII alphabetic or underscore (_) character, and must contain only ASCII alphanumeric, underscore, hash (#), period (.), space, colon (:), at (@), equals (=), and hyphen (-) characters. Choose a name that can be correlated with the function that the action performs. The following requirement applies only to the Citrix ADC CLI: If the name includes one or more spaces, enclose the name in double or single quotation marks (for example, "my action" or 'my action'). Minimum length = 1
__count	<Double>	Read-only	count parameter

Operations

(click to see Properties)

ADD| DELETE| RENAME| GET (ALL)| GET| COUNT

Some options that you can use for each operations:

- **Getting warnings in response:**NITRO allows you to get warnings in an operation by specifying the "warning" query parameter as "yes". For example, to get warnings while connecting to the Citrix ADC appliance, the URL is as follows:

  http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/login?warning=yes

  If any, the warnings are displayed in the response payload with the HTTP code "209 X-NITRO-WARNING".

- **Authenticated access for individual NITRO operations:**NITRO allows you to logon to the Citrix ADC appliance to perform individual operations. You can use this option instead of creating a NITRO session (using the login object) and then using that session to perform all operations,

  To do this, you must specify the username and password in the request header of the NITRO request as follows:

  X-NITRO-USER:<username>

  X-NITRO-PASS:<password>

  Note: In such cases, make sure that the request header DOES not include the following:

  Cookie:NITRO_AUTH_TOKEN=<tokenvalue>

`*Note:`*

Mandatory parameters are marked in redand placeholder content is marked in <green>.

add

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/spilloveraction

HTTP Method:POST

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Content-Type:application/json

Request Payload:

```
{"spilloveraction":{

<b>"name":<String_value>,

</b><b>"action":<String_value>

</b>}}
```

Response:

HTTP Status Code on Success: 201 Created HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error

delete

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/spilloveraction/name_value<String>

HTTP Method:DELETE

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue>

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error

rename

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/spilloveraction?**action=rename**

HTTP Method:POST

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Content-Type:application/json

Request Payload:

```
{"spilloveraction":{

<b>"name":<String_value>,

</b><b>"newname":<String_value>

</b>}}
```

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error

get (all)

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/spilloveraction

Query-parameters:

attrs

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/spilloveraction?**attrs=property-name1,property-name2**

Use this query parameter to specify the resource details that you want to retrieve.

filter

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/spilloveraction?**filter=property-name1:property-val1,property-name2:property-val2**

Use this query-parameter to get the filtered set of spilloveraction resources configured on Citrix ADC. Filtering can be done on any of the properties of the resource.

view

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/spilloveraction?**view=summary**

Note: By default, the retrieved results are displayed in detail view (?view=detail).

pagination

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/spilloveraction?**pagesize=#no;pageno=#no**

Use this query-parameter to get the spilloveraction resources in chunks.

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "spilloveraction": [ {

"name":<String_value>,

"action":<String_value>

}]}
```

get

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/spilloveraction/name_value<String>

Query-parameters:

attrs

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/spilloveraction/name_value<String>?**attrs=property-name1,property-name2**

Use this query parameter to specify the resource details that you want to retrieve.

view

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/spilloveraction/name_value<String>?**view=summary**

Note: By default, the retrieved results are displayed in detail view (?view=detail).

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "spilloveraction": [ {

"name":<String_value>,

"action":<String_value>

}]}
```

count

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/spilloveraction?**count=yes**

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "spilloveraction": [ { "__count": "#no"} ] }
```

spilloverpolicy

Configuration for Spillover policy resource.

Properties

(click to see Operations)

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
name	<String>	Read-write	Name of the spillover policy.
rule	<String>	Read-write	Expression to be used by the spillover policy.
action	<String>	Read-write	Action for the spillover policy. Action is created using add spillover action command.
comment	<String>	Read-write	Any comments that you might want to associate with the spillover policy.

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
newname	<String>	Read-write	New name for the spillover policy. Must begin with an ASCII alphabetic or underscore (_) character, and must contain only ASCII alphanumeric, underscore, hash (#), period (.), space, colon (:), at (@), equals (=), and hyphen (-) characters. Choose a name that reflects the function that the policy performs. The following requirement applies only to the Citrix ADC CLI: If the name includes one or more spaces, enclose the name in double or single quotation marks (for example, "my policy" or 'my policy'). Minimum length = 1
hits	<Double>	Read-only	The number of times the policy has been hit.

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
undefhits	<Double>	Read-only	Number of policy UNDEF hits.
__count	<Double>	Read-only	count parameter

Operations

(click to see Properties)

ADD| DELETE| UPDATE| UNSET| RENAME| GET (ALL)| GET| COUNT

Some options that you can use for each operations:

- **Getting warnings in response:**NITRO allows you to get warnings in an operation by specifying the "warning" query parameter as "yes". For example, to get warnings while connecting to the Citrix ADC appliance, the URL is as follows:

  http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/login?warning=yes

  If any, the warnings are displayed in the response payload with the HTTP code "209 X-NITRO-WARNING".

- **Authenticated access for individual NITRO operations:**NITRO allows you to logon to the Citrix ADC appliance to perform individual operations. You can use this option instead of creating a NITRO session (using the login object) and then using that session to perform all operations,

  To do this, you must specify the username and password in the request header of the NITRO request as follows:

  X-NITRO-USER:<username>

  X-NITRO-PASS:<password>

  Note: In such cases, make sure that the request header DOES not include the following:

  Cookie:NITRO_AUTH_TOKEN=<tokenvalue>

`*Note:`*

Mandatory parameters are marked in redand placeholder content is marked in <green>.

add

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/spilloverpolicy

HTTP Method:POST

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Content-Type:application/json

Request Payload:

```
{"spilloverpolicy":{

<b>"name":<String_value>,

</b><b>"rule":<String_value>,

</b><b>"action":<String_value>,

</b>"comment":<String_value>

}}
```

Response:

HTTP Status Code on Success: 201 Created HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error

delete

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/spilloverpolicy/name_value<String>

HTTP Method:DELETE

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue>

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error

update

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/spilloverpolicy

HTTP Method:PUT

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Content-Type:application/json

Request Payload:

```
{"spilloverpolicy":{

<b>"name":<String_value>,

</b>"rule":<String_value>,

"action":<String_value>,

"comment":<String_value>

}}
```

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error

unset

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/spilloverpolicy?**action=unset**

HTTP Method:POST

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Content-Type:application/json

Request Payload:

```
{"spilloverpolicy":{

<b>"name":<String_value>,

</b>"comment":true

}}
```

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error

rename

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/spilloverpolicy?**action=rename**

HTTP Method:POST

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Content-Type:application/json

Request Payload:

```
{"spilloverpolicy":{

<b>"name":<String_value>,

</b><b>"newname":<String_value>

</b>}}
```

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error

get (all)

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/spilloverpolicy

Query-parameters:

attrs

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/spilloverpolicy?**attrs=property-name1,property-name2**

Use this query parameter to specify the resource details that you want to retrieve.

filter

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/spilloverpolicy?**filter=property-name1:property-val1,property-name2:property-val2**

Use this query-parameter to get the filtered set of spilloverpolicy resources configured on Citrix ADC. Filtering can be done on any of the properties of the resource.

view

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/spilloverpolicy?**view=summary**

Note: By default, the retrieved results are displayed in detail view (?view=detail).

pagination

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/spilloverpolicy?**pagesize=#no;pageno=#no**

Use this query-parameter to get the spilloverpolicy resources in chunks.

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "spilloverpolicy": [ {

"name":<String_value>,

"rule":<String_value>,

"action":<String_value>,

"hits":<Double_value>,

"undefhits":<Double_value>,

"comment":<String_value>

}]}
```

get

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/spilloverpolicy/name_value<String>

Query-parameters:

attrs

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/spilloverpolicy/name_value<String>?**attrs=property-name1,property-name2**

Use this query parameter to specify the resource details that you want to retrieve.

view

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/spilloverpolicy/name_value<String>?**view=summary**

Note: By default, the retrieved results are displayed in detail view (?view=detail).

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "spilloverpolicy": [ {

"name":<String_value>,

"rule":<String_value>,

"action":<String_value>,

"hits":<Double_value>,

"undefhits":<Double_value>,

"comment":<String_value>

}]}
```

count

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/spilloverpolicy?**count=yes**

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "spilloverpolicy": [ { "__count": "#no"} ] }
```

spilloverpolicy_binding

Binding object which returns the resources bound to spilloverpolicy.

Properties

(click to see Operations)

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
name	<String>	Read-write	Name of the spillover policy.
spilloverpolicy_gslbvserver_binding	<spilloverpolicy_gslbvserver_binding[]>	Read-only	gslbvserver that can be bound to spilloverpolicy.
spilloverpolicy_csvserver_binding	<spilloverpolicy_csvserver_binding[]>	Read-only	csvserver that can be bound to spilloverpolicy.
spilloverpolicy_lbvserver_binding	<spilloverpolicy_lbvserver_binding[]>	Read-only	lbvserver that can be bound to spilloverpolicy.

Operations

(click to see Properties)

GET| GET (ALL)

Some options that you can use for each operations:

- **Getting warnings in response:**NITRO allows you to get warnings in an operation by specifying the "warning" query parameter as "yes". For example, to get warnings while connecting to the Citrix ADC appliance, the URL is as follows:

  http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/login?warning=yes

  If any, the warnings are displayed in the response payload with the HTTP code "209 X-NITRO-WARNING".

- **Authenticated access for individual NITRO operations:**NITRO allows you to logon to the Citrix ADC appliance to perform individual operations. You can use this option instead of creating a NITRO session (using the login object) and then using that session to perform all operations,

  To do this, you must specify the username and password in the request header of the NITRO request as follows:

  X-NITRO-USER:<username>

  X-NITRO-PASS:<password>

  Note: In such cases, make sure that the request header DOES not include the following:

  Cookie:NITRO_AUTH_TOKEN=<tokenvalue>

`*Note:`*

Mandatory parameters are marked in redand placeholder content is marked in <green>.

get

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/spilloverpolicy_binding/name_value<String>

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "spilloverpolicy_binding": [ {

"name":<String_value>,

"spilloverpolicy_csvserver_binding":<spilloverpolicy_csvserver_binding[]_v
alue>,

"spilloverpolicy_lbvserver_binding":<spilloverpolicy_lbvserver_binding[]_v
alue>,

"spilloverpolicy_gslbvserver_binding":<spilloverpolicy_gslbvserver_binding
[]_value>

}]}
```

get (all)

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/spilloverpolicy_binding

Query-parameters:

bulkbindings

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/spilloverpolicy_binding?**bulkbindings=yes**

NITRO allows you to fetch bindings in bulk.

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "spilloverpolicy_binding": [ {

"name":<String_value>,

"spilloverpolicy_csvserver_binding":<spilloverpolicy_csvserver_binding[]_value>,

"spilloverpolicy_lbvserver_binding":<spilloverpolicy_lbvserver_binding[]_value>,

"spilloverpolicy_gslbvserver_binding":<spilloverpolicy_gslbvserver_binding[]_value>

}]}
```

spilloverpolicy_csvserver_binding

Binding object showing the csvserver that can be bound to spilloverpolicy.

Properties

(click to see Operations)

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
boundto	<String>	Read-write	The name of the entity to which the policy is bound.

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
name	<String>	Read-write	Name of the spillover policy.
priority	<Double>	Read-only	Specifies the priority of the policy.
labelname	<String>	Read-only	Name of the label to invoke if the current policy rule evaluates to TRUE.
gotopriorityexpression	<String>	Read-only	Expression specifying the priority of the next policy which will get evaluated if the current policy rule evaluates to TRUE.
labeltype	<String>	Read-only	Type of policy label invocation. Possible values = reqvserver, resvserver, policylabel
activepolicy	<Double>	Read-only	Indicates whether policy is bound or not.
__count	<Double>	Read-only	count parameter

Operations

(click to see Properties)

GET| GET (ALL)| COUNT

Some options that you can use for each operations:

- **Getting warnings in response:**NITRO allows you to get warnings in an operation by specifying the "warning" query parameter as "yes". For example, to get warnings while connecting to the Citrix ADC appliance, the URL is as follows:

  http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/login?warning=yes

  If any, the warnings are displayed in the response payload with the HTTP code "209 X-NITRO-WARNING".

- **Authenticated access for individual NITRO operations:**NITRO allows you to logon to the Citrix ADC appliance to perform individual operations. You can use this option instead of creating a NITRO session (using the login object) and then using that session to perform all operations,

  To do this, you must specify the username and password in the request header of the NITRO request as follows:

  X-NITRO-USER:<username>

  X-NITRO-PASS:<password>

  Note: In such cases, make sure that the request header DOES not include the following:

  Cookie:NITRO_AUTH_TOKEN=<tokenvalue>

`*Note:`*

Mandatory parameters are marked in redand placeholder content is marked in <green>.

get

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/spilloverpolicy_csvserver_binding/name_value<String>

Query-parameters:

filter

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/spilloverpolicy_csvserver_binding/name_value<String>?**filter=property-name1:property-value1,property-name2:property-value2**

Use this query-parameter to get the filtered set of spilloverpolicy_csvserver_binding resources configured on Citrix ADC. Filtering can be done on any of the properties of the resource.

pagination

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/spilloverpolicy_csvserver_binding/name_value<String>?**pagesize=#no;pageno=#no**

Use this query-parameter to get the spilloverpolicy_csvserver_binding resources in chunks.

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "spilloverpolicy_csvserver_binding": [ {

"boundto":<String_value>,

"name":<String_value>,

"priority":<Double_value>,

"labelname":<String_value>,

"gotopriorityexpression":<String_value>,

"labeltype":<String_value>,

"activepolicy":<Double_value>

}]}
```

get (all)

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/spilloverpolicy_csvserver_binding

Query-parameters:

bulkbindings

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/spilloverpolicy_csvserver_binding?**bulkbindings=yes**

NITRO allows you to fetch bindings in bulk.

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "spilloverpolicy_csvserver_binding": [ {

"boundto":<String_value>,

"name":<String_value>,

"priority":<Double_value>,

"labelname":<String_value>,

"gotopriorityexpression":<String_value>,

"labeltype":<String_value>,
```

```
"activepolicy":<Double_value>

}]}
```

count

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/spilloverpolicy_csvserver_binding/name_value<String>?**count=yes**

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{"spilloverpolicy_csvserver_binding": [ { "__count": "#no"} ] }
```

spilloverpolicy_gslbvserver_binding

Binding object showing the gslbvserver that can be bound to spilloverpolicy.

Properties

(click to see Operations)

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
boundto	<String>	Read-write	The name of the entity to which the policy is bound.
name	<String>	Read-write	Name of the spillover policy.
priority	<Double>	Read-only	Specifies the priority of the policy.
labelname	<String>	Read-only	Name of the label to invoke if the current policy rule evaluates to TRUE.
gotopriorityexpression	<String>	Read-only	Expression specifying the priority of the next policy which will get evaluated if the current policy rule evaluates to TRUE.
labeltype	<String>	Read-only	Type of policy label invocation. Possible values = reqvserver, resvserver, policylabel
activepolicy	<Double>	Read-only	Indicates whether policy is bound or not.
__count	<Double>	Read-only	count parameter

Operations

(click to see Properties)

GET| GET (ALL)| COUNT

Some options that you can use for each operations:

- **Getting warnings in response:**NITRO allows you to get warnings in an operation by specifying the "warning" query parameter as "yes". For example, to get warnings while connecting to the Citrix ADC appliance, the URL is as follows:

  http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/login?warning=yes

  If any, the warnings are displayed in the response payload with the HTTP code "209 X-NITRO-WARNING".

- **Authenticated access for individual NITRO operations:**NITRO allows you to logon to the Citrix ADC appliance to perform individual operations. You can use this option instead of creating a NITRO session (using the login object) and then using that session to perform all operations,

  To do this, you must specify the username and password in the request header of the NITRO request as follows:

  X-NITRO-USER:<username>

  X-NITRO-PASS:<password>

  Note: In such cases, make sure that the request header DOES not include the following:

  Cookie:NITRO_AUTH_TOKEN=<tokenvalue>

`*Note:`*

Mandatory parameters are marked in redand placeholder content is marked in <green>.

get

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/spilloverpolicy_gslbvserver_binding/name_value<String>

Query-parameters:

filter

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/spilloverpolicy_gslbvserver_binding/name_value<String>?**filter=property-name1:property-value1,property-name2:property-value2**

Use this query-parameter to get the filtered set of spilloverpolicy_gslbvserver_binding resources configured on Citrix ADC. Filtering can be done on any of the properties of the resource.

pagination

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/spilloverpolicy_gslbvserver_binding/name_value<String>?**pagesize=#no;pageno=#no**

Use this query-parameter to get the spilloverpolicy_gslbvserver_binding resources in chunks.

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "spilloverpolicy_gslbvserver_binding": [ {

"boundto":<String_value>,

"name":<String_value>,

"priority":<Double_value>,

"labelname":<String_value>,

"gotopriorityexpression":<String_value>,

"labeltype":<String_value>,

"activepolicy":<Double_value>

}]}
```

get (all)

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/spilloverpolicy_gslbvserver_binding

Query-parameters:

bulkbindings

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/spilloverpolicy_gslbvserver_binding?**bulkbindings=yes**

NITRO allows you to fetch bindings in bulk.

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "spilloverpolicy_gslbvserver_binding": [ {

"boundto":<String_value>,

"name":<String_value>,

"priority":<Double_value>,

"labelname":<String_value>,

"gotopriorityexpression":<String_value>,

"labeltype":<String_value>,
```

```
"activepolicy":<Double_value>

}]}
```

count

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/spilloverpolicy_gslbvserver_binding/name_value<String>?**count=yes**

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{"spilloverpolicy_gslbvserver_binding": [ { "__count": "#no"} ] }
```

spilloverpolicy_lbvserver_binding

Binding object showing the lbvserver that can be bound to spilloverpolicy.

Properties

(click to see Operations)

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
boundto	<String>	Read-write	The name of the entity to which the policy is bound.
name	<String>	Read-write	Name of the spillover policy.
priority	<Double>	Read-only	Specifies the priority of the policy.
labelname	<String>	Read-only	Name of the label to invoke if the current policy rule evaluates to TRUE.
gotopriorityexpression	<String>	Read-only	Expression specifying the priority of the next policy which will get evaluated if the current policy rule evaluates to TRUE.
labeltype	<String>	Read-only	Type of policy label invocation. Possible values = reqvserver, resvserver, policylabel
activepolicy	<Double>	Read-only	Indicates whether policy is bound or not.
__count	<Double>	Read-only	count parameter

Operations

(click to see Properties)

GET| GET (ALL)| COUNT

Some options that you can use for each operations:

- **Getting warnings in response:**NITRO allows you to get warnings in an operation by specifying the "warning" query parameter as "yes". For example, to get warnings while connecting to the Citrix ADC appliance, the URL is as follows:

  http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/login?warning=yes

  If any, the warnings are displayed in the response payload with the HTTP code "209 X-NITRO-WARNING".

- **Authenticated access for individual NITRO operations:**NITRO allows you to logon to the Citrix ADC appliance to perform individual operations. You can use this option instead of creating a NITRO session (using the login object) and then using that session to perform all operations,

  To do this, you must specify the username and password in the request header of the NITRO request as follows:

  X-NITRO-USER:<username>

  X-NITRO-PASS:<password>

  Note: In such cases, make sure that the request header DOES not include the following:

  Cookie:NITRO_AUTH_TOKEN=<tokenvalue>

`*Note:`*

Mandatory parameters are marked in redand placeholder content is marked in <green>.

get

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/spilloverpolicy_lbvserver_binding/name_value<String>

Query-parameters:

filter

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/spilloverpolicy_lbvserver_binding/name_value<String>?**filter=property-name1:property-value1,property-name2:property-value2**

Use this query-parameter to get the filtered set of spilloverpolicy_lbvserver_binding resources configured on Citrix ADC. Filtering can be done on any of the properties of the resource.

pagination

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/spilloverpolicy_lbvserver_binding/name_value<String>?**pagesize=#no;pageno=#no**

Use this query-parameter to get the spilloverpolicy_lbvserver_binding resources in chunks.

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "spilloverpolicy_lbvserver_binding": [ {

"boundto":<String_value>,

"name":<String_value>,

"priority":<Double_value>,

"labelname":<String_value>,

"gotopriorityexpression":<String_value>,

"labeltype":<String_value>,

"activepolicy":<Double_value>

}]}
```

get (all)

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/spilloverpolicy_lbvserver_binding

Query-parameters:

bulkbindings

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/spilloverpolicy_lbvserver_binding?**bulkbindings=yes**

NITRO allows you to fetch bindings in bulk.

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "spilloverpolicy_lbvserver_binding": [ {

"boundto":<String_value>,

"name":<String_value>,

"priority":<Double_value>,

"labelname":<String_value>,

"gotopriorityexpression":<String_value>,

"labeltype":<String_value>,
```

```
"activepolicy":<Double_value>

}]}
```

count

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/spilloverpolicy_lbvserver_binding/name_value<String>?**count=yes**

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{"spilloverpolicy_lbvserver_binding": [ { "__count": "#no"} ] }
```

SSL

SSL

SSL Configuration

NAME	DESCRIPTION
sslaction	Configuration for SSL action

NAME	DESCRIPTION
sslcacertgroup	Configuration for Group of CA certificate-key pairs
sslcacertgroup_binding	Binding object showing the resources that can be bound to sslcacertgroup
sslcacertgroup_sslcertkey_binding	Binding object showing the sslcertkey that can be bound to sslcacertgroup
sslcert	Configuration for cerificate
sslcertbundle	Configuration for Imported Certbundle
sslcertchain	Configuration for CERT Chain
sslcertchain_binding	Binding object showing the resources that can be bound to sslcertchain
sslcertchain_sslcertkey_binding	Binding object showing the sslcertkey that can be bound to sslcertchain
sslcertfile	Configuration for Imported Certfile
sslcertkey	Configuration for certificate key
sslcertkey_binding	Binding object showing the resources that can be bound to sslcertkey
sslcertkey_crldistribution_binding	Binding object showing the crldistribution that can be bound to sslcertkey
sslcertkey_service_binding	Binding object showing the service that can be bound to sslcertkey
sslcertkey_sslocspresponder_binding	Binding object showing the sslocspresponder that can be bound to sslcertkey
sslcertkey_sslprofile_binding	Binding object showing the sslprofile that can be bound to sslcertkey
sslcertkey_sslvserver_binding	Binding object showing the sslvserver that can be bound to sslcertkey

NAME	DESCRIPTION
sslcertlink	Configuration for linked certificate
sslcertreq	Configuration for certificate request
sslcipher	Configuration for cipher
sslcipher_binding	Binding object showing the resources that can be bound to sslcipher
sslcipher_individualcipher_binding	Binding object showing the individualcipher that can be bound to sslcipher
sslcipher_service_binding	Binding object showing the service that can be bound to sslcipher
sslcipher_servicegroup_binding	Binding object showing the servicegroup that can be bound to sslcipher
sslcipher_sslciphersuite_binding	Binding object showing the sslciphersuite that can be bound to sslcipher
sslcipher_sslprofile_binding	Binding object showing the sslprofile that can be bound to sslcipher
sslcipher_sslvserver_binding	Binding object showing the sslvserver that can be bound to sslcipher
sslciphersuite	Configuration for ciphersuite
sslcrl	Configuration for Certificate Revocation List
sslcrl_binding	Binding object showing the resources that can be bound to sslcrl
sslcrl_serialnumber_binding	Binding object showing the serialnumber that can be bound to sslcrl
sslcrlfile	Configuration for Imported crl files
ssldhfile	Configuration for dh imported file
ssldhparam	Configuration for dh Parameter

NAME	DESCRIPTION
ssldsakey	Configuration for dsa key
ssldtlsprofile	Configuration for DTLS profile
sslecdsakey	Configuration for ecdsa key
sslfips	Configuration for fips
sslfipskey	Configuration for FIPS key
sslfipssimsource	Configuration for FIPsSIM source
sslfipssimtarget	Configuration for FIPS SIM Target
sslglobal_binding	Binding object showing the resources that can be bound to sslglobal
sslglobal_sslpolicy_binding	Binding object showing the sslpolicy that can be bound to sslglobal
sslhsmkey	Configuration for HSM key
sslkeyfile	Configuration for Imported ssl key files
ssllogprofile	Configuration for SSL logging Profile
sslocspresponder	Configuration for OCSP responser
sslparameter	Configuration for SSL parameter
sslpkcs12	Configuration for pkcs12
sslpkcs8	Configuration for pkcs8
sslpolicy	Configuration for SSL policy
sslpolicy_binding	Binding object showing the resources that can be bound to sslpolicy
sslpolicy_csvserver_binding	Binding object showing the csvserver that can be bound to sslpolicy

NAME	DESCRIPTION
sslpolicy_lbvserver_binding	Binding object showing the lbvserver that can be bound to sslpolicy
sslpolicy_sslglobal_binding	Binding object showing the sslglobal that can be bound to sslpolicy
sslpolicy_sslpolicylabel_binding	Binding object showing the sslpolicylabel that can be bound to sslpolicy
sslpolicy_sslservice_binding	Binding object showing the sslservice that can be bound to sslpolicy
sslpolicy_sslvserver_binding	Binding object showing the sslvserver that can be bound to sslpolicy
sslpolicylabel	Configuration for SSL policy label
sslpolicylabel_binding	Binding object showing the resources that can be bound to sslpolicylabel
sslpolicylabel_sslpolicy_binding	Binding object showing the sslpolicy that can be bound to sslpolicylabel
sslprofile	Configuration for SSL profile
sslprofile_binding	Binding object showing the resources that can be bound to sslprofile
sslprofile_ecccurve_binding	Binding object showing the ecccurve that can be bound to sslprofile
sslprofile_sslcertkey_binding	Binding object showing the sslcertkey that can be bound to sslprofile
sslprofile_sslcipher_binding	Binding object showing the sslcipher that can be bound to sslprofile
sslprofile_sslciphersuite_binding	Binding object showing the sslciphersuite that can be bound to sslprofile
sslprofile_sslvserver_binding	Binding object showing the sslvserver that can be bound to sslprofile

NAME	DESCRIPTION
sslrsakey	Configuration for RSA key
sslservice	Configuration for SSL service
sslservice_binding	Binding object showing the resources that can be bound to sslservice
sslservice_ecccurve_binding	Binding object showing the ecccurve that can be bound to sslservice
sslservice_sslcertkey_binding	Binding object showing the sslcertkey that can be bound to sslservice
sslservice_sslcipher_binding	Binding object showing the sslcipher that can be bound to sslservice
sslservice_sslciphersuite_binding	Binding object showing the sslciphersuite that can be bound to sslservice
sslservice_sslpolicy_binding	Binding object showing the sslpolicy that can be bound to sslservice
sslservicegroup	Configuration for SSL service group
sslservicegroup_binding	Binding object showing the resources that can be bound to sslservicegroup
sslservicegroup_ecccurve_binding	Binding object showing the ecccurve that can be bound to sslservicegroup
sslservicegroup_sslcertkey_binding	Binding object showing the sslcertkey that can be bound to sslservicegroup
sslservicegroup_sslcipher_binding	Binding object showing the sslcipher that can be bound to sslservicegroup
sslservicegroup_sslciphersuite_binding	Binding object showing the sslciphersuite that can be bound to sslservicegroup
sslvserver	Configuration for SSL virtual server

NAME	DESCRIPTION
sslvserver_binding	Binding object showing the resources that can be bound to sslvserver
sslvserver_ecccurve_binding	Binding object showing the ecccurve that can be bound to sslvserver
sslvserver_sslcertkey_binding	Binding object showing the sslcertkey that can be bound to sslvserver
sslvserver_sslcipher_binding	Binding object showing the sslcipher that can be bound to sslvserver
sslvserver_sslciphersuite_binding	Binding object showing the sslciphersuite that can be bound to sslvserver
sslvserver_sslpolicy_binding	Binding object showing the sslpolicy that can be bound to sslvserver
sslwrapkey	Configuration for WRAP key

sslaction

Configuration for SSL action resource.

Properties

(click to see Operations)

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
name	<String>	Read-write	Name for the SSL action. Must begin with an ASCII alphanumeric or underscore (_) character, and must contain only ASCII alphanumeric, underscore, hash (#), period (.), space, colon (:), at (@), equals (=), and

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
			hyphen (-) characters. Cannot be changed after the action is created. The following requirement applies only to the Citrix ADC CLI: If the name includes one or more spaces, enclose the name in double or single quotation marks (for example, "my action" or 'my action'). Minimum length = 1
clientauth	<String>	Read-write	Perform client certificate authentication. Possible values = DOCLIENTAUTH, NOCLIENTAUTH
clientcertverification	<String>	Read-write	Client certificate verification is mandatory or optional. Default value: Mandatory Possible values = Mandatory, Optional
ssllogprofile	<String>	Read-write	The name of the ssllogprofile.

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
			Minimum length = 1 Maximum length = 127
clientcert	<String>	Read-write	Insert the entire client certificate into the HTTP header of the request being sent to the web server. The certificate is inserted in ASCII (PEM) format. Possible values = ENABLED, DISABLED
certheader	<String>	Read-write	Name of the header into which to insert the client certificate.
clientcertserialnumber	<String>	Read-write	Insert the entire client serial number into the HTTP header of the request being sent to the web server. Possible values = ENABLED, DISABLED
certserialheader	<String>	Read-write	Name of the header into which to insert the client serial number.
clientcertsubject	<String>	Read-write	Insert the client certificate subject, also known as the distinguished name

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
			(DN), into the HTTP header of the request being sent to the web server. Possible values = ENABLED, DISABLED
certsubjectheader	<String>	Read-write	Name of the header into which to insert the client certificate subject.
clientcerthash	<String>	Read-write	Insert the certificate's signature into the HTTP header of the request being sent to the web server. The signature is the value extracted directly from the X.509 certificate signature field. All X.509 certificates contain a signature field. Possible values = ENABLED, DISABLED
certhashheader	<String>	Read-write	Name of the header into which to insert the client certificate signature (hash).
clientcertfingerprint	<String>	Read-write	Insert the certificate's fingerprint into the HTTP header of the request being sent to the web server. The fingerprint is derived by computing the

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
			specified hash value (SHA256, for example) of the DER-encoding of the client certificate. Possible values = ENABLED, DISABLED
certfingerprintheader	<String>	Read-write	Name of the header into which to insert the client certificate fingerprint.
certfingerprintdigest	<String>	Read-write	Digest algorithm used to compute the fingerprint of the client certificate. Possible values = SHA1, SHA224, SHA256, SHA384, SHA512
clientcertissuer	<String>	Read-write	Insert the certificate issuer details into the HTTP header of the request being sent to the web server. Possible values = ENABLED, DISABLED
certissuerheader	<String>	Read-write	Name of the header into which to insert the client certificate issuer details.
sessionid	<String>	Read-write	Insert the SSL session ID into the HTTP header of the

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
			request being sent to the web server. Every SSL connection that the client and the Citrix ADC share has a unique ID that identifies the specific connection. Possible values = ENABLED, DISABLED
sessionidheader	<String>	Read-write	Name of the header into which to insert the Session ID.
cipher	<String>	Read-write	Insert the cipher suite that the client and the Citrix ADC negotiated for the SSL session into the HTTP header of the request being sent to the web server. The appliance inserts the cipher-suite name, SSL protocol, export or non-export string, and cipher strength bit, depending on the type of browser connecting to the SSL virtual server or service (for example, Cipher-Suite: RC4-MD5 SSLv3 Non-Export 128-bit).

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
			Possible values = ENABLED, DISABLED
cipherheader	<String>	Read-write	Name of the header into which to insert the name of the cipher suite.
clientcertnotbefore	<String>	Read-write	Insert the date from which the certificate is valid into the HTTP header of the request being sent to the web server. Every certificate is configured with the date and time from which it is valid. Possible values = ENABLED, DISABLED
certnotbeforeheader	<String>	Read-write	Name of the header into which to insert the date and time from which the certificate is valid.
clientcertnotafter	<String>	Read-write	Insert the date of expiry of the certificate into the HTTP header of the request being sent to the web server. Every certificate is configured with the date and time at which the certificate expires.

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
			Possible values = ENABLED, DISABLED
certnotafterheader	<String>	Read-write	Name of the header into which to insert the certificate's expiry date.
owasupport	<String>	Read-write	If the appliance is in front of an Outlook Web Access (OWA) server, insert a special header field, FRONT-END-HTTPS: ON, into the HTTP requests going to the OWA server. This header communicates to the server that the transaction is HTTPS and not HTTP. Possible values = ENABLED, DISABLED
forward	<String>	Read-write	This action takes an argument a vserver name, to this vserver one will be able to forward all the packets. Minimum length = 1 Maximum length = 127
cacertgrpname	<String>	Read-write	This action will allow to pick CA(s) from

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
			the specific CA group, to verify the client certificate. Minimum length = 1 Maximum length = 31
hits	<Double>	Read-only	The number of times the action has been taken.
undefhits	<Double>	Read-only	The number of times the action resulted in UNDEF.
referencecount	<Double>	Read-only	The number of references to the action.
description	<String>	Read-only	Description of the action.
__count	<Double>	Read-only	count parameter

Operations

(click to see Properties)

ADD| DELETE| GET (ALL)| GET| COUNT

Some options that you can use for each operations:

- **Getting warnings in response:**NITRO allows you to get warnings in an operation by specifying the "warning" query parameter as "yes". For example, to get warnings while connecting to the Citrix ADC appliance, the URL is as follows:

  http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/login?warning=yes

  If any, the warnings are displayed in the response payload with the HTTP code "209 X-NITRO-WARNING".

- **Authenticated access for individual NITRO operations:**NITRO allows you to logon to the Citrix ADC appliance to perform individual operations. You can use this option instead of creating a NITRO session (using the login object) and then using that session to perform all operations,

  To do this, you must specify the username and password in the request header of the NITRO request as follows:

  X-NITRO-USER:<username>

  X-NITRO-PASS:<password>

  Note: In such cases, make sure that the request header DOES not include the following:

  Cookie:NITRO_AUTH_TOKEN=<tokenvalue>

`*Note:`*

Mandatory parameters are marked in redand placeholder content is marked in <green>.

add

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/sslaction

HTTP Method:POST

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Content-Type:application/json

Request Payload:

```
{"sslaction":{

<b>"name":<String_value>,

</b>"clientauth":<String_value>,

"clientcertverification":<String_value>,

"ssllogprofile":<String_value>,

"clientcert":<String_value>,
```

```
"certheader":<String_value>,

"clientcertserialnumber":<String_value>,

"certserialheader":<String_value>,

"clientcertsubject":<String_value>,

"certsubjectheader":<String_value>,

"clientcerthash":<String_value>,

"certhashheader":<String_value>,

"clientcertfingerprint":<String_value>,

"certfingerprintheader":<String_value>,

"certfingerprintdigest":<String_value>,

"clientcertissuer":<String_value>,

"certissuerheader":<String_value>,

"sessionid":<String_value>,

"sessionidheader":<String_value>,

"cipher":<String_value>,

"cipherheader":<String_value>,

"clientcertnotbefore":<String_value>,

"certnotbeforeheader":<String_value>,

"clientcertnotafter":<String_value>,

"certnotafterheader":<String_value>,

"owasupport":<String_value>,

"forward":<String_value>,
```

```
"cacertgrpname":<String_value>

}}
```

Response:

HTTP Status Code on Success: 201 Created HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error

delete

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/sslaction/name_value<String>

HTTP Method:DELETE

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue>

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error

get (all)

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/sslaction

Query-parameters:

attrs

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/sslaction?**attrs=property-name1,property-name2**

Use this query parameter to specify the resource details that you want to retrieve.

filter

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/sslaction?**filter=property-name1:property-val1,property-name2:property-val2**

Use this query-parameter to get the filtered set of sslaction resources configured on Citrix ADC. Filtering can be done on any of the properties of the resource.

view

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/sslaction?**view=summary**

Note: By default, the retrieved results are displayed in detail view (?view=detail).

pagination

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/sslaction?**pagesize=#no;pageno=#no**

Use this query-parameter to get the sslaction resources in chunks.

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "sslaction": [ {

"name":<String_value>,

"clientauth":<String_value>,

"clientcertverification":<String_value>,

"clientcert":<String_value>,

"certheader":<String_value>,
```

```
"clientcertserialnumber":<String_value>,

"certserialheader":<String_value>,

"clientcertsubject":<String_value>,

"certsubjectheader":<String_value>,

"clientcerthash":<String_value>,

"certhashheader":<String_value>,

"clientcertfingerprint":<String_value>,

"certfingerprintheader":<String_value>,

"certfingerprintdigest":<String_value>,

"clientcertissuer":<String_value>,

"certissuerheader":<String_value>,

"sessionid":<String_value>,

"sessionidheader":<String_value>,

"cipher":<String_value>,

"cipherheader":<String_value>,

"owasupport":<String_value>,

"forward":<String_value>,

"cacertgrpname":<String_value>,

"clientcertnotbefore":<String_value>,

"certnotbeforeheader":<String_value>,

"clientcertnotafter":<String_value>,

"certnotafterheader":<String_value>,
```

```
"hits":<Double_value>,

"undefhits":<Double_value>,

"referencecount":<Double_value>,

"description":<String_value>,

"ssllogprofile":<String_value>

}]}
```

get

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/sslaction/name_value<String>

Query-parameters:

attrs

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/sslaction/name_value<String>?**attrs=property-name1,property-name2**

Use this query parameter to specify the resource details that you want to retrieve.

view

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/sslaction/name_value<String>?**view=summary**

Note: By default, the retrieved results are displayed in detail view (?view=detail).

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "sslaction": [ {

"name":<String_value>,

"clientauth":<String_value>,

"clientcertverification":<String_value>,

"clientcert":<String_value>,

"certheader":<String_value>,

"clientcertserialnumber":<String_value>,

"certserialheader":<String_value>,

"clientcertsubject":<String_value>,

"certsubjectheader":<String_value>,

"clientcerthash":<String_value>,

"certhashheader":<String_value>,

"clientcertfingerprint":<String_value>,

"certfingerprintheader":<String_value>,

"certfingerprintdigest":<String_value>,

"clientcertissuer":<String_value>,

"certissuerheader":<String_value>,

"sessionid":<String_value>,

"sessionidheader":<String_value>,
```

```
"cipher":<String_value>,

"cipherheader":<String_value>,

"owasupport":<String_value>,

"forward":<String_value>,

"cacertgrpname":<String_value>,

"clientcertnotbefore":<String_value>,

"certnotbeforeheader":<String_value>,

"clientcertnotafter":<String_value>,

"certnotafterheader":<String_value>,

"hits":<Double_value>,

"undefhits":<Double_value>,

"referencecount":<Double_value>,

"description":<String_value>,

"ssllogprofile":<String_value>

}]}
```

count

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/sslaction?**count=yes**

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "sslaction": [ { "__count": "#no"} ] }
```

sslcert

Configuration for cerificate resource.

Properties

(click to see Operations)

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
certfile	<String>	Read-write	Name for and, optionally, path to the generated certificate file. /nsconfig/ssl/ is the default path. Maximum length = 63
reqfile	<String>	Read-write	Name for and, optionally, path to the certificate-signing request (CSR). /nsconfig/ssl/ is the default path. Maximum length = 63

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
certtype	<String>	Read-write	Type of certificate to generate. Specify one of the following: * ROOT_CERT - Self-signed Root-CA certificate. You must specify the key file name. The generated Root-CA certificate can be used for signing end-user client or server certificates or to create Intermediate-CA certificates. * INTM_CERT - Intermediate-CA certificate. * CLNT_CERT - End-user client certificate used for client authentication. * SRVR_CERT - SSL server certificate used on SSL servers for end-to-end encryption. Possible values = ROOT_CERT, INTM_CERT,

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
			CLNT_CERT, SRVR_CERT
keyfile	<String>	Read-write	Name for and, optionally, path to the private key. You can either use an existing RSA or DSA key that you own or create a new private key on the Citrix ADC. This file is required only when creating a self-signed Root-CA certificate. The key file is stored in the /nsconfig/ssl directory by default. If the input key specified is an encrypted key, you are prompted to enter the PEM pass phrase that was used for encrypting the key. Maximum length = 63
keyform	<String>	Read-write	Format in which the key is stored on the appliance. Default value: PEM Possible values = DER, PEM
pempassphrase	<String>	Read-write	.

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
			Minimum length = 1 Maximum length = 31
days	<Double>	Read-write	Number of days for which the certificate will be valid, beginning with the time and day (system time) of creation. Default value: 365 Minimum value = 1 Maximum value = 3650
subjectaltname	<String>	Read-write	Subject Alternative Name (SAN) is an extension to X.509 that allows various values to be associated with a security certificate using a subjectAltName field. These values are called "Subject Alternative Names" (SAN). Names include: 1. Email addresses 2. IP addresses 3. URIs

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
			4. DNS names (This is usually also provided as the Common Name RDN within the Subject field of the main certificate.) 5. directory names (alternative Distinguished Names to that given in the Subject). Minimum length = 1
certform	<String>	Read-write	Format in which the certificate is stored on the appliance. Default value: PEM Possible values = DER, PEM
cacert	<String>	Read-write	Name of the CA certificate file that issues and signs the Intermediate-CA certificate or the end-user client and server certificates. Maximum length = 63
cacertform	<String>	Read-write	Format of the CA certificate. Default value: PEM

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
			Possible values = DER, PEM
cakey	<String>	Read-write	Private key, associated with the CA certificate that is used to sign the Intermediate-CA certificate or the end-user client and server certificate. If the CA key file is password protected, the user is prompted to enter the pass phrase that was used to encrypt the key. Maximum length = 63
cakeyform	<String>	Read-write	Format for the CA certificate. Default value: PEM Possible values = DER, PEM
caserial	<String>	Read-write	Serial number file maintained for the CA certificate. This file contains the serial number of the next certificate to be issued or signed by the CA. If the specified file does not exist, a new file is created, with /nsconfig/ssl/ as the default path. If you

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
			do not specify a proper path for the existing serial file, a new serial file is created. This might change the certificate serial numbers assigned by the CA certificate to each of the certificates it signs. Maximum length = 63

Operations

(click to see Properties)

CREATE

Some options that you can use for each operations:

- **Getting warnings in response:**NITRO allows you to get warnings in an operation by specifying the "warning" query parameter as "yes". For example, to get warnings while connecting to the Citrix ADC appliance, the URL is as follows:

  http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/login?warning=yes

  If any, the warnings are displayed in the response payload with the HTTP code "209 X-NITRO-WARNING".

- **Authenticated access for individual NITRO operations:**NITRO allows you to logon to the Citrix ADC appliance to perform individual operations. You can use this option instead of creating a NITRO session (using the login object) and then using that session to perform all operations,

  To do this, you must specify the username and password in the request header of the NITRO request as follows:

  X-NITRO-USER:<username>

  X-NITRO-PASS:<password>

Note: In such cases, make sure that the request header DOES not include the following:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue>

`*Note:` *

Mandatory parameters are marked in redand placeholder content is marked in <green>.

create

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/sslcert?**action=create**

HTTP Method:POST

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Content-Type:application/json

Request Payload:

```
{"sslcert":{

<b>"certfile":<String_value>,

</b><b>"reqfile":<String_value>,

</b><b>"certtype":<String_value>,

</b>"keyfile":<String_value>,

"keyform":<String_value>,

"pempassphrase":<String_value>,

"days":<Double_value>,

"subjectaltname":<String_value>,

"certform":<String_value>,

"cacert":<String_value>,

"cacertform":<String_value>,
```

```
"cakey":<String_value>,

"cakeyform":<String_value>,

"caserial":<String_value>

}}
```

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error

sslcertbundle

Configuration for Imported Certbundle resource.

Properties

(click to see Operations)

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
name	<String>	Read-write	Name to assign to the imported certificate bundle. Must begin with an ASCII alphanumeric or underscore (_) character, and must contain only ASCII alphanumeric, underscore, hash (#), period (.), space, colon (:), at (@), equals (=), and hyphen (-) characters. The following requirement applies

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
			only to the Citrix ADC CLI: If the name includes one or more spaces, enclose the name in double or single quotation marks (for example, "my file" or 'my file'). Minimum length = 1 Maximum length = 31
src	<String>	Read-write	URL specifying the protocol, host, and path, including file name, to the certificate bundle to be imported or exported. For example, http://www.example.com/cert_bundle_file. NOTE: The import fails if the object to be imported is on an HTTPS server that requires client certificate authentication for access. Minimum length = 1 Maximum length = 2047
inuse	<String>	Read-only	.

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
			Possible values = YES, NO
__count	<Double>	Read-only	count parameter

Operations

(click to see Properties)

IMPORT| DELETE| APPLY| EXPORT| GET (ALL)| COUNT

Some options that you can use for each operations:

- **Getting warnings in response:**NITRO allows you to get warnings in an operation by specifying the "warning" query parameter as "yes". For example, to get warnings while connecting to the Citrix ADC appliance, the URL is as follows:

  http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/login?warning=yes

  If any, the warnings are displayed in the response payload with the HTTP code "209 X-NITRO-WARNING".

- **Authenticated access for individual NITRO operations:**NITRO allows you to logon to the Citrix ADC appliance to perform individual operations. You can use this option instead of creating a NITRO session (using the login object) and then using that session to perform all operations,

  To do this, you must specify the username and password in the request header of the NITRO request as follows:

  X-NITRO-USER:<username>

  X-NITRO-PASS:<password>

  Note: In such cases, make sure that the request header DOES not include the following:

  Cookie:NITRO_AUTH_TOKEN=<tokenvalue>

`*Note:`*

Mandatory parameters are marked in redand placeholder content is marked in <green>.

Import

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/sslcertbundle?**action=Import**

HTTP Method:POST

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Content-Type:application/json

Request Payload:

```
{"sslcertbundle":{

<b>"name":<String_value>,

</b><b>"src":<String_value>

</b>}}
```

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error

delete

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/sslcertbundle

HTTP Method:DELETE

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue>

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error

apply

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/sslcertbundle?**action=apply**

HTTP Method:POST

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Content-Type:application/json

Request Payload:

```
{"sslcertbundle":{

<b>"name":<String_value>

</b>}}
```

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error

export

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/sslcertbundle?**action=export**

HTTP Method:POST

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Content-Type:application/json

Request Payload:

```
{"sslcertbundle":{

<b>"name":<String_value>,
```

```
</b><b>"src":<String_value>

</b>}}
```

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error

get (all)

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/sslcertbundle

Query-parameters:

attrs

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/sslcertbundle?**attrs=property-name1,property-name2**

Use this query parameter to specify the resource details that you want to retrieve.

filter

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/sslcertbundle?**filter=property-name1:property-val1,property-name2:property-val2**

Use this query-parameter to get the filtered set of sslcertbundle resources configured on Citrix ADC. Filtering can be done on any of the properties of the resource.

view

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/sslcertbundle?**view=summary**

Note: By default, the retrieved results are displayed in detail view (?view=detail).

pagination

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/sslcertbundle?**pagesize=#no;pageno=#no**

Use this query-parameter to get the sslcertbundle resources in chunks.

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "sslcertbundle": [ {

"name":<String_value>,

"inuse":<String_value>,

"src":<String_value>

}]}
```

count

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/sslcertbundle?**count=yes**

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "sslcertbundle": [ { "__count": "#no"} ] }
```

sslcertchain

Configuration for CERT Chain resource.

Properties

(click to see Operations)

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
certkeyname	<String>	Read-write	Name of the Certificate. Minimum length = 1
__count	<Double>	Read-only	count parameter

Operations

(click to see Properties)

GET (ALL)| GET| COUNT

Some options that you can use for each operations:

- **Getting warnings in response:**NITRO allows you to get warnings in an operation by specifying the "warning" query parameter as "yes". For example, to get warnings while connecting to the Citrix ADC appliance, the URL is as follows:

  http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/login?warning=yes

  If any, the warnings are displayed in the response payload with the HTTP code "209 X-NITRO-WARNING".

- **Authenticated access for individual NITRO operations:**NITRO allows you to logon to the Citrix ADC appliance to perform individual operations. You can use this option instead of

creating a NITRO session (using the login object) and then using that session to perform all operations,

To do this, you must specify the username and password in the request header of the NITRO request as follows:

X-NITRO-USER:<username>

X-NITRO-PASS:<password>

Note: In such cases, make sure that the request header DOES not include the following:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue>

`*Note:`*

Mandatory parameters are marked in redand placeholder content is marked in <green>.

get (all)

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/sslcertchain

Query-parameters:

attrs

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/sslcertchain?**attrs=property-name1,property-name2**

Use this query parameter to specify the resource details that you want to retrieve.

filter

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/sslcertchain?**filter=property-name1:property-val1,property-name2:property-val2**

Use this query-parameter to get the filtered set of sslcertchain resources configured on Citrix ADC. Filtering can be done on any of the properties of the resource.

view

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/sslcertchain?**view=summary**

Note: By default, the retrieved results are displayed in detail view (?view=detail).

pagination

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/sslcertchain?**pagesize=#no;pageno=#no**

Use this query-parameter to get the sslcertchain resources in chunks.

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "sslcertchain": [ {

"certkeyname":<String_value>,



}]}
```

get

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/sslcertchain/certkeyname_value<String>

Query-parameters:

attrs

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/sslcertchain/certkeyname_value<String>?**attrs=property-name1,property-name2**

Use this query parameter to specify the resource details that you want to retrieve.

view

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/sslcertchain/certkeyname_value<String>?**view=summary**

Note: By default, the retrieved results are displayed in detail view (?view=detail).

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "sslcertchain": [ {

"certkeyname":<String_value>,



}]}
```

count

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/sslcertchain?**count=yes**

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "sslcertchain": [ { "__count": "#no"} ] }
```

sslcertchain_binding

Binding object which returns the resources bound to sslcertchain.

Properties

(click to see Operations)

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
certkeyname	<String>	Read-write	Name of the Certificate. Minimum length = 1
sslcertchain_sslcertkey_binding	<sslcertchain_sslcertkey_binding[]>	Read-only	sslcertkey that can be bound to sslcertchain.

Operations

(click to see Properties)

GET| GET (ALL)

Some options that you can use for each operations:

- **Getting warnings in response:**NITRO allows you to get warnings in an operation by specifying the "warning" query parameter as "yes". For example, to get warnings while connecting to the Citrix ADC appliance, the URL is as follows:

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/login?warning=yes

If any, the warnings are displayed in the response payload with the HTTP code "209 X-NITRO-WARNING".

- **Authenticated access for individual NITRO operations:**NITRO allows you to logon to the Citrix ADC appliance to perform individual operations. You can use this option instead of creating a NITRO session (using the login object) and then using that session to perform all operations,

  To do this, you must specify the username and password in the request header of the NITRO request as follows:

  X-NITRO-USER:<username>

  X-NITRO-PASS:<password>

  Note: In such cases, make sure that the request header DOES not include the following:

  Cookie:NITRO_AUTH_TOKEN=<tokenvalue>

`*Note:`*

Mandatory parameters are marked in redand placeholder content is marked in <green>.

get

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/sslcertchain_binding/certkeyname_value<String>

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "sslcertchain_binding": [ {

"certkeyname":<String_value>,

"sslcertchain_sslcertkey_binding":<sslcertchain_sslcertkey_binding[]_value
>

}]}
```

get (all)

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/sslcertchain_binding

Query-parameters:

bulkbindings

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/sslcertchain_binding?**bulkbindings=yes**

NITRO allows you to fetch bindings in bulk.

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "sslcertchain_binding": [ {
```

```
"certkeyname":<String_value>,

"sslcertchain_sslcertkey_binding":<sslcertchain_sslcertkey_binding[]_value
>

}]}
```

sslcertchain_sslcertkey_binding

Binding object showing the sslcertkey that can be bound to sslcertchain.

Properties

(click to see Operations)

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
linkcertkeyname	<String>	Read-write	Name of the Linked Certificate. Minimum length = 1
certkeyname	<String>	Read-write	Name of the Certificate. Minimum length = 1
isca	<Boolean>	Read-only	Used to find if certificate is a CA.
islinked	<Boolean>	Read-only	Used to find if certificate is linked.
addsubject	<Boolean>	Read-only	Used to find if certificate is linked.
__count	<Double>	Read-only	count parameter

Operations

(click to see Properties)

GET| GET (ALL)| COUNT

Some options that you can use for each operations:

- **Getting warnings in response:**NITRO allows you to get warnings in an operation by specifying the "warning" query parameter as "yes". For example, to get warnings while connecting to the Citrix ADC appliance, the URL is as follows:

  http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/login?warning=yes

  If any, the warnings are displayed in the response payload with the HTTP code "209 X-NITRO-WARNING".

- **Authenticated access for individual NITRO operations:**NITRO allows you to logon to the Citrix ADC appliance to perform individual operations. You can use this option instead of creating a NITRO session (using the login object) and then using that session to perform all operations,

  To do this, you must specify the username and password in the request header of the NITRO request as follows:

  X-NITRO-USER:<username>

  X-NITRO-PASS:<password>

  Note: In such cases, make sure that the request header DOES not include the following:

  Cookie:NITRO_AUTH_TOKEN=<tokenvalue>

`*Note:`*

Mandatory parameters are marked in redand placeholder content is marked in <green>.

get

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/sslcertchain_sslcertkey_binding/certkeyname_value<String>

Query-parameters:

filter

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/sslcertchain_sslcertkey_binding/certkeyname_value<String>?**filter=property-name1:property-value1,property-name2:property-value2**

Use this query-parameter to get the filtered set of sslcertchain_sslcertkey_binding resources configured on Citrix ADC. Filtering can be done on any of the properties of the resource.

pagination

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/sslcertchain_sslcertkey_binding/certkeyname_value<String>?**pagesize=#no;pageno=#no**

Use this query-parameter to get the sslcertchain_sslcertkey_binding resources in chunks.

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "sslcertchain_sslcertkey_binding": [ {

"linkcertkeyname":<String_value>,

"certkeyname":<String_value>,

"isca":<Boolean_value>,

"islinked":<Boolean_value>,

"addsubject":<Boolean_value>

}]}
```

get (all)

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/sslcertchain_sslcertkey_binding

Query-parameters:

bulkbindings

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/sslcertchain_sslcertkey_binding?**bulkbindings=yes**

NITRO allows you to fetch bindings in bulk.

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "sslcertchain_sslcertkey_binding": [ {

"linkcertkeyname":<String_value>,

"certkeyname":<String_value>,

"isca":<Boolean_value>,

"islinked":<Boolean_value>,

"addsubject":<Boolean_value>

}]}
```

count

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/sslcertchain_sslcertkey_binding/certkeyname_value<String>?**count=yes**

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{"sslcertchain_sslcertkey_binding": [ { "__count": "#no"} ] }
```

sslcertfile

Configuration for Imported Certfile resource.

Properties

(click to see Operations)

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
name	<String>	Read-write	Name to assign to the imported certificate file. Must begin with an ASCII alphanumeric or underscore (_)

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
			character, and must contain only ASCII alphanumeric, underscore, hash (#), period (.), space, colon (:), at (@), equals (=), and hyphen (-) characters. The following requirement applies only to the Citrix ADC CLI: If the name includes one or more spaces, enclose the name in double or single quotation marks (for example, "my file" or 'my file'). Minimum length = 1 Maximum length = 31
src	<String>	Read-write	URL specifying the protocol, host, and path, including file name, to the certificate file to be imported. For example, http://www.example.com/cert_file. NOTE: The import fails if the object to be imported is on an HTTPS server that requires client certificate

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
			authentication for access. Minimum length = 1 Maximum length = 2047
__count	<Double>	Read-only	count parameter

Operations

(click to see Properties)

IMPORT| DELETE| GET (ALL)| COUNT

Some options that you can use for each operations:

- **Getting warnings in response:**NITRO allows you to get warnings in an operation by specifying the "warning" query parameter as "yes". For example, to get warnings while connecting to the Citrix ADC appliance, the URL is as follows:

  http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/login?warning=yes

  If any, the warnings are displayed in the response payload with the HTTP code "209 X-NITRO-WARNING".
- **Authenticated access for individual NITRO operations:**NITRO allows you to logon to the Citrix ADC appliance to perform individual operations. You can use this option instead of creating a NITRO session (using the login object) and then using that session to perform all operations,

  To do this, you must specify the username and password in the request header of the NITRO request as follows:

  X-NITRO-USER:<username>

  X-NITRO-PASS:<password>

  Note: In such cases, make sure that the request header DOES not include the following:

  Cookie:NITRO_AUTH_TOKEN=<tokenvalue>

`*Note:` *

Mandatory parameters are marked in redand placeholder content is marked in <green>.

Import

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/sslcertfile?**action=Import**

HTTP Method:POST

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Content-Type:application/json

Request Payload:

```
{"sslcertfile":{

<b>"name":<String_value>,

</b><b>"src":<String_value>

</b>}}
```

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error

delete

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/sslcertfile

HTTP Method:DELETE

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue>

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error

get (all)

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/sslcertfile

Query-parameters:

attrs

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/sslcertfile?**attrs=property-name1,property-name2**

Use this query parameter to specify the resource details that you want to retrieve.

filter

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/sslcertfile?**filter=property-name1:property-val1,property-name2:property-val2**

Use this query-parameter to get the filtered set of sslcertfile resources configured on Citrix ADC. Filtering can be done on any of the properties of the resource.

view

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/sslcertfile?**view=summary**

Note: By default, the retrieved results are displayed in detail view (?view=detail).

pagination

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/sslcertfile?**pagesize=#no;pageno=#no**

Use this query-parameter to get the sslcertfile resources in chunks.

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "sslcertfile": [ {

"name":<String_value>,

"src":<String_value>

}]}
```

count

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/sslcertfile?**count=yes**

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "sslcertfile": [ { "__count": "#no"} ] }
```

sslcertkey

Configuration for certificate key resource.

Properties

(click to see Operations)

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
certkey	<String>	Read-write	Name for the certificate and private-key pair. Must begin with an ASCII alphanumeric or underscore (_) character, and must contain only ASCII alphanumeric, underscore, hash (#), period (.), space, colon (:), at (@), equals (=), and hyphen (-) characters. Cannot be changed after the certificate-key pair is created. The following requirement applies only to the Citrix ADC CLI: If the name includes one or more spaces, enclose the name in double or single quotation marks (for example,

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
			"my cert" or 'my cert'). Minimum length = 1
cert	<String>	Read-write	Name of and, optionally, path to the X509 certificate file that is used to form the certificate-key pair. The certificate file should be present on the appliance's hard-disk drive or solid-state drive. Storing a certificate in any location other than the default might cause inconsistency in a high availability setup. /nsconfig/ssl/ is the default path. Minimum length = 1
key	<String>	Read-write	Name of and, optionally, path to the private-key file that is used to form the certificate-key pair. The certificate file should be present on the appliance's hard-disk drive or solid-state drive. Storing a certificate in any location other than the default might cause inconsistency in a high availability

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
			setup. /nsconfig/ssl/ is the default path. Minimum length = 1
password	<Boolean>	Read-write	Passphrase that was used to encrypt the private-key. Use this option to load encrypted private-keys in PEM format.
fipskey	<String>	Read-write	Name of the FIPS key that was created inside the Hardware Security Module (HSM) of a FIPS appliance, or a key that was imported into the HSM. Minimum length = 1
hsmkey	<String>	Read-write	Name of the HSM key that was created in the External Hardware Security Module (HSM) of a FIPS appliance. Minimum length = 1
inform	<String>	Read-write	Input format of the certificate and the private-key files. The three formats supported by the appliance are: PEM - Privacy Enhanced Mail DER - Distinguished Encoding Rule

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
			PFX - Personal Information Exchange. Default value: PEM Possible values = DER, PEM, PFX
passplain	<String>	Read-write	Pass phrase used to encrypt the private-key. Required when adding an encrypted private-key in PEM format. Minimum length = 1
expirymonitor	<String>	Read-write	Issue an alert when the certificate is about to expire. Possible values = ENABLED, DISABLED
notificationperiod	<Double>	Read-write	Time, in number of days, before certificate expiration, at which to generate an alert that the certificate is about to expire. Minimum value = 10 Maximum value = 100
bundle	<String>	Read-write	Parse the certificate chain as a single file after linking the server certificate to

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
			its issuer's certificate within the file. Default value: NO Possible values = YES, NO
deletefromdevice	<Boolean>	Read-write	Delete cert/key file from file system.
linkcertkeyname	<String>	Read-write	Name of the Certificate Authority certificate-key pair to which to link a certificate-key pair. Minimum length = 1
nodomaincheck	<Boolean>	Read-write	Override the check for matching domain names during a certificate update operation.
ocspstaplingcache	<Boolean>	Read-write	Clear cached ocspStapling response in certkey.
signaturealg	<String>	Read-only	Signature algorithm.
certificatetype	<String[]>	Read-only	Specifies whether the certificate is of type root-CA, intermediate-CA, server, client, or client and server. Possible values = ROOT_CERT, INTM_CERT, CLNT_CERT,

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
			SRVR_CERT, UNKNOWN_CERT
serial	<String>	Read-only	Serial number.
issuer	<String>	Read-only	Issuer name.
clientcertnotbefore	<String>	Read-only	Not-Before date.
clientcertnotafter	<String>	Read-only	Not-After date.
daystoexpiration	<Integer>	Read-only	Days remaining for the certificate to expire.
subject	<String>	Read-only	Subject name.
publickey	<String>	Read-only	Public key algorithm.
publickeysize	<Integer>	Read-only	Size of the public key.
version	<Integer>	Read-only	Version.
priority	<Double>	Read-only	ocsp priority.
status	<String>	Read-only	Status of the certificate. Possible values = Valid, Not yet valid, Expired
passcrypt	<String>	Read-only	Passcrypt. Minimum length = 1
data	<Double>	Read-only	Vserver Id.
servicename	<String>	Read-only	Service name to which the certificate key pair is bound.

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
ocspresponsestatus	<String>	Read-only	Ocsp response status of the certificate. Possible values = NONE, EXPIRED, VALID
__count	<Double>	Read-only	count parameter

Operations

(click to see Properties)

ADD| DELETE| UPDATE| UNSET| LINK| UNLINK| CHANGE| CLEAR| GET (ALL)| GET| COUNT

Some options that you can use for each operations:

- **Getting warnings in response:**NITRO allows you to get warnings in an operation by specifying the "warning" query parameter as "yes". For example, to get warnings while connecting to the Citrix ADC appliance, the URL is as follows:

  http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/login?warning=yes

  If any, the warnings are displayed in the response payload with the HTTP code "209 X-NITRO-WARNING".

- **Authenticated access for individual NITRO operations:**NITRO allows you to logon to the Citrix ADC appliance to perform individual operations. You can use this option instead of creating a NITRO session (using the login object) and then using that session to perform all operations,

  To do this, you must specify the username and password in the request header of the NITRO request as follows:

  X-NITRO-USER:<username>

  X-NITRO-PASS:<password>

  Note: In such cases, make sure that the request header DOES not include the following:

  Cookie:NITRO_AUTH_TOKEN=<tokenvalue>

`*Note:` *

Mandatory parameters are marked in redand placeholder content is marked in <green>.

add

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/sslcertkey

HTTP Method:POST

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Content-Type:application/json

Request Payload:

```
{"sslcertkey":{

<b>"certkey":<String_value>,

</b><b>"cert":<String_value>,

</b>"key":<String_value>,

"password":<Boolean_value>,

"fipskey":<String_value>,

"hsmkey":<String_value>,

"inform":<String_value>,

"passplain":<String_value>,

"expirymonitor":<String_value>,

"notificationperiod":<Double_value>,

"bundle":<String_value>

}}
```

Response:

HTTP Status Code on Success: 201 Created HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error

delete

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/sslcertkey/certkey_value<String>

HTTP Method:DELETE

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue>

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error

update

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/sslcertkey

HTTP Method:PUT

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Content-Type:application/json

Request Payload:

```
{"sslcertkey":{

<b>"certkey":<String_value>,

</b>"expirymonitor":<String_value>,

"notificationperiod":<Double_value>
```

```
}}
```

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error

unset

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/sslcertkey?**action=unset**

HTTP Method:POST

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Content-Type:application/json

Request Payload:

```
{"sslcertkey":{

<b>"certkey":<String_value>,

</b>"expirymonitor":true,

"notificationperiod":true

}}
```

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error

link

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/sslcertkey?**action=link**

HTTP Method:POST

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Content-Type:application/json

Request Payload:

```
{"sslcertkey":{

<b>"certkey":<String_value>,

</b><b>"linkcertkeyname":<String_value>

</b>}}
```

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error

unlink

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/sslcertkey?**action=unlink**

HTTP Method:POST

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Content-Type:application/json

Request Payload:

```
{"sslcertkey":{

<b>"certkey":<String_value>

</b>}}
```

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error

change

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/sslcertkey?**action=update**

HTTP Method:POST

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Content-Type:application/json

Request Payload:

```
{"sslcertkey":{

<b>"certkey":<String_value>,

</b>"cert":<String_value>,

"key":<String_value>,

"password":<Boolean_value>,

"fipskey":<String_value>,

"inform":<String_value>,

"passplain":<String_value>,

"nodomaincheck":<Boolean_value>

}}
```

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error

clear

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/sslcertkey?**action=clear**

HTTP Method:POST

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Content-Type:application/json

Request Payload:

```
{"sslcertkey":{

<b>"certkey":<String_value>,

</b>"ocspstaplingcache":<Boolean_value>

}}
```

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error

get (all)

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/sslcertkey

Query-parameters:

attrs

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/sslcertkey?**attrs=property-name1,property-name2**

Use this query parameter to specify the resource details that you want to retrieve.

filter

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/sslcertkey?**filter=property-name1:property-val1,property-name2:property-val2**

Use this query-parameter to get the filtered set of sslcertkey resources configured on Citrix ADC. Filtering can be done on any of the properties of the resource.

view

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/sslcertkey?**view=summary**

Note: By default, the retrieved results are displayed in detail view (?view=detail).

pagination

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/sslcertkey?**pagesize=#no;pageno=#no**

Use this query-parameter to get the sslcertkey resources in chunks.

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "sslcertkey": [ {

"certkey":<String_value>,

"cert":<String_value>,

"key":<String_value>,
```

```
"inform":<String_value>,

"signaturealg":<String_value>,

"certificatetype":<String[]_value>,

"serial":<String_value>,

"issuer":<String_value>,

"clientcertnotbefore":<String_value>,

"clientcertnotafter":<String_value>,

"daystoexpiration":<Integer_value>,

"subject":<String_value>,

"publickey":<String_value>,

"publickeysize":<Integer_value>,

"version":<Integer_value>,

"priority":<Double_value>,

"status":<String_value>,

"fipskey":<String_value>,

"hsmkey":<String_value>,

"passcrypt":<String_value>,

"data":<Double_value>,

"servicename":<String_value>,

"expirymonitor":<String_value>,

"notificationperiod":<Double_value>,

"linkcertkeyname":<String_value>,
```

```
"ocspresponsestatus":<String_value>

}]}
```

get

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/sslcertkey/certkey_value<String>

Query-parameters:

attrs

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/sslcertkey/certkey_value<String>?**attrs=property-name1,property-name2**

Use this query parameter to specify the resource details that you want to retrieve.

view

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/sslcertkey/certkey_value<String>?**view=summary**

Note: By default, the retrieved results are displayed in detail view (?view=detail).

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "sslcertkey": [ {
```

```
"certkey":<String_value>,

"cert":<String_value>,

"key":<String_value>,

"inform":<String_value>,

"signaturealg":<String_value>,

"certificatetype":<String[]_value>,

"serial":<String_value>,

"issuer":<String_value>,

"clientcertnotbefore":<String_value>,

"clientcertnotafter":<String_value>,

"daystoexpiration":<Integer_value>,

"subject":<String_value>,

"publickey":<String_value>,

"publickeysize":<Integer_value>,

"version":<Integer_value>,

"priority":<Double_value>,

"status":<String_value>,

"fipskey":<String_value>,

"hsmkey":<String_value>,

"passcrypt":<String_value>,

"data":<Double_value>,

"servicename":<String_value>,
```

```
"expirymonitor":<String_value>,

"notificationperiod":<Double_value>,

"linkcertkeyname":<String_value>,

"ocspresponsestatus":<String_value>

}]}
```

count

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/sslcertkey?**count=yes**

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "sslcertkey": [ { "__count": "#no"} ] }
```

sslcertkey_binding

Binding object which returns the resources bound to sslcertkey.

Properties

(click to see Operations)

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
certkey	<String>	Read-write	Name of the certificate-key pair for which to show detailed information. Minimum length = 1
sslcertkey_crldistribution_binding	<sslcertkey_crldistribution_binding[]>	Read-only	crldistribution that can be bound to sslcertkey.
sslcertkey_sslvserver_binding	<sslcertkey_sslvserver_binding[]>	Read-only	sslvserver that can be bound to sslcertkey.
sslcertkey_sslocspresponder_binding	<sslcertkey_sslocspresponder_binding[]>	Read-only	sslocspresponder that can be bound to sslcertkey.
sslcertkey_service_binding	<sslcertkey_service_binding[]>	Read-only	service that can be bound to sslcertkey.
sslcertkey_sslprofile_binding	<sslcertkey_sslprofile_binding[]>	Read-only	sslprofile that can be bound to sslcertkey.

Operations

(click to see Properties)

GET| GET (ALL)

Some options that you can use for each operations:

- **Getting warnings in response:**NITRO allows you to get warnings in an operation by specifying the "warning" query parameter as "yes". For example, to get warnings while connecting to the Citrix ADC appliance, the URL is as follows:

  http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/login?warning=yes

  If any, the warnings are displayed in the response payload with the HTTP code "209 X-NITRO-WARNING".

- **Authenticated access for individual NITRO operations:**NITRO allows you to logon to the Citrix ADC appliance to perform individual operations. You can use this option instead of creating a NITRO session (using the login object) and then using that session to perform all operations,

  To do this, you must specify the username and password in the request header of the NITRO request as follows:

  X-NITRO-USER:<username>

  X-NITRO-PASS:<password>

  Note: In such cases, make sure that the request header DOES not include the following:

  Cookie:NITRO_AUTH_TOKEN=<tokenvalue>

`*Note:`*

Mandatory parameters are marked in redand placeholder content is marked in <green>.

get

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/sslcertkey_binding/certkey_value<String>

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "sslcertkey_binding": [ {
```

```
"certkey":<String_value>,

"sslcertkey_sslprofile_binding":<sslcertkey_sslprofile_binding[]_value>,

"sslcertkey_sslocspresponder_binding":<sslcertkey_sslocspresponder_binding
[]_value>,

"sslcertkey_service_binding":<sslcertkey_service_binding[]_value>,

"sslcertkey_crldistribution_binding":<sslcertkey_crldistribution_binding[]
_value>,

"sslcertkey_sslvserver_binding":<sslcertkey_sslvserver_binding[]_value>

}]}
```

get (all)

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/sslcertkey_binding

Query-parameters:

bulkbindings

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/sslcertkey_binding?**bulkbindings=yes**

NITRO allows you to fetch bindings in bulk.

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "sslcertkey_binding": [ {

"certkey":<String_value>,

"sslcertkey_sslprofile_binding":<sslcertkey_sslprofile_binding[]_value>,

"sslcertkey_sslocspresponder_binding":<sslcertkey_sslocspresponder_binding
[]_value>,

"sslcertkey_service_binding":<sslcertkey_service_binding[]_value>,

"sslcertkey_crldistribution_binding":<sslcertkey_crldistribution_binding[]
_value>,

"sslcertkey_sslvserver_binding":<sslcertkey_sslvserver_binding[]_value>

}]}
```

sslcertkey_crldistribution_binding

Binding object showing the crldistribution that can be bound to sslcertkey.

Properties

(click to see Operations)

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
ca	<Boolean>	Read-write	The certificate-key pair being unbound is a Certificate Authority (CA) certificate. If you choose this option, the certificate-key pair is unbound from the list of CA certificates that were bound to the specified SSL virtual

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
			server or SSL service.
issuer	<String>	Read-write	Issuer name.
certkey	<String>	Read-write	Name of the certificate-key pair. Minimum length = 1
__count	<Double>	Read-write	count parameter

Operations

(click to see Properties)

GET| GET (ALL)| COUNT

Some options that you can use for each operations:

- **Getting warnings in response:**NITRO allows you to get warnings in an operation by specifying the "warning" query parameter as "yes". For example, to get warnings while connecting to the Citrix ADC appliance, the URL is as follows:

  http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/login?warning=yes

  If any, the warnings are displayed in the response payload with the HTTP code "209 X-NITRO-WARNING".

- **Authenticated access for individual NITRO operations:**NITRO allows you to logon to the Citrix ADC appliance to perform individual operations. You can use this option instead of creating a NITRO session (using the login object) and then using that session to perform all operations,

  To do this, you must specify the username and password in the request header of the NITRO request as follows:

  X-NITRO-USER:<username>

  X-NITRO-PASS:<password>

  Note: In such cases, make sure that the request header DOES not include the following:

  Cookie:NITRO_AUTH_TOKEN=<tokenvalue>

`*Note:` *

Mandatory parameters are marked in redand placeholder content is marked in <green>.

get

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/sslcertkey_crldistribution_binding/certkey_value<String>

Query-parameters:

filter

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/sslcertkey_crldistribution_binding/certkey_value<String>?**filter=property-name1:property-value1,property-name2:property-value2**

Use this query-parameter to get the filtered set of sslcertkey_crldistribution_binding resources configured on Citrix ADC. Filtering can be done on any of the properties of the resource.

pagination

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/sslcertkey_crldistribution_binding/certkey_value<String>?**pagesize=#no;pageno=#no**

Use this query-parameter to get the sslcertkey_crldistribution_binding resources in chunks.

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "sslcertkey_crldistribution_binding": [ {

"ca":<Boolean_value>,

"issuer":<String_value>,

"certkey":<String_value>

}]}
```

get (all)

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/sslcertkey_crldistribution_binding

Query-parameters:

bulkbindings

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/sslcertkey_crldistribution_binding?**bulkbindings=yes**

NITRO allows you to fetch bindings in bulk.

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "sslcertkey_crldistribution_binding": [ {

"ca":<Boolean_value>,

"issuer":<String_value>,

"certkey":<String_value>

}]}
```

count

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/sslcertkey_crldistribution_binding/certkey_value<String>?**count=yes**

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{"sslcertkey_crldistribution_binding": [ { "__count": "#no"} ] }
```

sslcertkey_service_binding

Binding object showing the service that can be bound to sslcertkey.

Properties

(click to see Operations)

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
servicegroupname	<String>	Read-write	The name of the SSL service group to which the certificate-key pair needs to be bound. Use the "add servicegroup" command to create this service. Minimum length = 1
ca	<Boolean>	Read-write	The certificate-key pair being unbound is a Certificate Authority (CA) certificate. If you choose this option, the certificate-key pair is unbound from the list of CA certificates that were bound to the specified SSL virtual server or SSL service.
service	<Boolean>	Read-write	Bind the certificate to the named SSL service or service group.
servicename	<String>	Read-write	Service name to which the certificate key pair is bound.
certkey	<String>	Read-write	Name of the certificate-key pair. Minimum length = 1

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
version	<Integer>	Read-only	Version.
data	<Double>	Read-only	Vserver Id.
__count	<Double>	Read-only	count parameter

Operations

(click to see Properties)

GET| GET (ALL)| COUNT

Some options that you can use for each operations:

- **Getting warnings in response:**NITRO allows you to get warnings in an operation by specifying the "warning" query parameter as "yes". For example, to get warnings while connecting to the Citrix ADC appliance, the URL is as follows:

  http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/login?warning=yes

  If any, the warnings are displayed in the response payload with the HTTP code "209 X-NITRO-WARNING".

- **Authenticated access for individual NITRO operations:**NITRO allows you to logon to the Citrix ADC appliance to perform individual operations. You can use this option instead of creating a NITRO session (using the login object) and then using that session to perform all operations,

  To do this, you must specify the username and password in the request header of the NITRO request as follows:

  X-NITRO-USER:<username>

  X-NITRO-PASS:<password>

  Note: In such cases, make sure that the request header DOES not include the following:

  Cookie:NITRO_AUTH_TOKEN=<tokenvalue>

`*Note:`*

Mandatory parameters are marked in redand placeholder content is marked in <green>.

get

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/sslcertkey_service_binding/certkey_value<String>

Query-parameters:

filter

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/sslcertkey_service_binding/certkey_value<String>?**filter=property-name1:property-value1,property-name2:property-value2**

Use this query-parameter to get the filtered set of sslcertkey_service_binding resources configured on Citrix ADC. Filtering can be done on any of the properties of the resource.

pagination

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/sslcertkey_service_binding/certkey_value<String>?**pagesize=#no;pageno=#no**

Use this query-parameter to get the sslcertkey_service_binding resources in chunks.

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "sslcertkey_service_binding": [ {

"servicegroupname":<String_value>,
```

```
"ca":<Boolean_value>,

"service":<Boolean_value>,

"servicename":<String_value>,

"certkey":<String_value>,

"version":<Integer_value>,

"data":<Double_value>

}]}
```

get (all)

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/sslcertkey_service_binding

Query-parameters:

bulkbindings

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/sslcertkey_service_binding?**bulkbindings=yes**

NITRO allows you to fetch bindings in bulk.

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "sslcertkey_service_binding": [ {

"servicegroupname":<String_value>,

"ca":<Boolean_value>,

"service":<Boolean_value>,

"servicename":<String_value>,

"certkey":<String_value>,

"version":<Integer_value>,

"data":<Double_value>

}]}
```

count

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/sslcertkey_service_binding/certkey_value<String>?**count=yes**

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{"sslcertkey_service_binding": [ { "__count": "#no"} ] }
```

sslcertkey_sslocspresponder_binding

Binding object showing the sslocspresponder that can be bound to sslcertkey.

Properties

(click to see Operations)

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
priority	<Double>	Read-write	ocsp priority.
ca	<Boolean>	Read-write	The certificate-key pair being unbound is a Certificate Authority (CA) certificate. If you choose this option, the certificate-key pair is unbound from the list of CA certificates that were bound to the specified SSL virtual server or SSL service.
certkey	<String>	Read-write	Name of the certificate-key pair. Minimum length = 1
ocspresponder	<String>	Read-write	OCSP responders bound to this certkey.
__count	<Double>	Read-write	count parameter

Operations

(click to see Properties)

ADD:| DELETE:| GET| GET (ALL)| COUNT

Some options that you can use for each operations:

- **Getting warnings in response:**NITRO allows you to get warnings in an operation by specifying the "warning" query parameter as "yes". For example, to get warnings while connecting to the Citrix ADC appliance, the URL is as follows:

  http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/login?warning=yes

  If any, the warnings are displayed in the response payload with the HTTP code "209 X-NITRO-WARNING".

- **Authenticated access for individual NITRO operations:**NITRO allows you to logon to the Citrix ADC appliance to perform individual operations. You can use this option instead of creating a NITRO session (using the login object) and then using that session to perform all operations,

  To do this, you must specify the username and password in the request header of the NITRO request as follows:

  X-NITRO-USER:<username>

  X-NITRO-PASS:<password>

  Note: In such cases, make sure that the request header DOES not include the following:

  Cookie:NITRO_AUTH_TOKEN=<tokenvalue>

`*Note:`*

Mandatory parameters are marked in redand placeholder content is marked in <green>.

add

URL:http://<Citrix-ADC-IP-address(NSIP)/nitro/v1/config/sslcertkey_sslocspresponder_binding

HTTP Method:PUT

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Content-Type:application/json

Request Payload:

```
{
"sslcertkey_sslocspresponder_binding":{
"certkey":<String_value>,
"ocspresponder":<String_value>,
"priority":<Double_value>
}}
```

Response:

HTTP Status Code on Success: 201 Created HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error

delete

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/sslcertkey_sslocspresponder_binding/certkey_value<String>

Query-parameters:

args

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/sslcertkey_sslocspresponder_binding/certkey_value<String>?**args=ocspresponder:<String_value>,ca:<Boolean_value>**

HTTP Method:DELETE

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue>

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error

get

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/sslcertkey_sslocspresponder_binding/certkey_value<String>

Query-parameters:

filter

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/sslcertkey_sslocspresponder_binding/certkey_value<String>?**filter=property-name1:property-value1,property-name2:property-value2**

Use this query-parameter to get the filtered set of sslcertkey_sslocspresponder_binding resources configured on Citrix ADC. Filtering can be done on any of the properties of the resource.

pagination

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/sslcertkey_sslocspresponder_binding/certkey_value<String>?**pagesize=#no;pageno=#no**

Use this query-parameter to get the sslcertkey_sslocspresponder_binding resources in chunks.

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "sslcertkey_sslocspresponder_binding": [ {

"priority":<Double_value>,

"ca":<Boolean_value>,

"certkey":<String_value>,

"ocspresponder":<String_value>

}]}
```

get (all)

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/sslcertkey_sslocspresponder_binding

Query-parameters:

bulkbindings

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/sslcertkey_sslocspresponder_binding?**bulkbindings=yes**

NITRO allows you to fetch bindings in bulk.

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "sslcertkey_sslocspresponder_binding": [ {

"priority":<Double_value>,

"ca":<Boolean_value>,

"certkey":<String_value>,

"ocspresponder":<String_value>

}]}
```

count

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/sslcertkey_sslocspresponder_binding/certkey_value<String>?**count=yes**

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{"sslcertkey_sslocspresponder_binding": [ { "__count": "#no"} ] }
```

sslcertkey_sslprofile_binding

Binding object showing the sslprofile that can be bound to sslcertkey.

Properties

(click to see Operations)

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
ca	<Boolean>	Read-write	The certificate-key pair being unbound is a Certificate Authority (CA) certificate. If you choose this option, the certificate-key pair is unbound from the list of CA certificates that were bound to the specified SSL virtual server or SSL service.
sslprofile	<String>	Read-write	SSL profile name to which the certificate key pair is bound.
certkey	<String>	Read-write	Name of the certificate-key pair. Minimum length = 1
__count	<Double>	Read-write	count parameter

Operations

(click to see Properties)

GET| GET (ALL)| COUNT

Some options that you can use for each operations:

- **Getting warnings in response:**NITRO allows you to get warnings in an operation by specifying the "warning" query parameter as "yes". For example, to get warnings while connecting to the Citrix ADC appliance, the URL is as follows:

  http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/login?warning=yes

  If any, the warnings are displayed in the response payload with the HTTP code "209 X-NITRO-WARNING".

- **Authenticated access for individual NITRO operations:**NITRO allows you to logon to the Citrix ADC appliance to perform individual operations. You can use this option instead of creating a NITRO session (using the login object) and then using that session to perform all operations,

  To do this, you must specify the username and password in the request header of the NITRO request as follows:

  X-NITRO-USER:<username>

  X-NITRO-PASS:<password>

  Note: In such cases, make sure that the request header DOES not include the following:

  Cookie:NITRO_AUTH_TOKEN=<tokenvalue>

`*Note:`*

Mandatory parameters are marked in redand placeholder content is marked in <green>.

get

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/sslcertkey_sslprofile_binding/certkey_value<String>

Query-parameters:

filter

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/sslcertkey_sslprofile_binding/certkey_value<String>?**filter=property-name1:property-value1,property-name2:property-value2**

Use this query-parameter to get the filtered set of sslcertkey_sslprofile_binding resources configured on Citrix ADC. Filtering can be done on any of the properties of the resource.

pagination

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/sslcertkey_sslprofile_binding/certkey_value<String>?**pagesize=#no;pageno=#no**

Use this query-parameter to get the sslcertkey_sslprofile_binding resources in chunks.

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "sslcertkey_sslprofile_binding": [ {

"ca":<Boolean_value>,

"sslprofile":<String_value>,

"certkey":<String_value>

}]}
```

get (all)

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/sslcertkey_sslprofile_binding

Query-parameters:

bulkbindings

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/sslcertkey_sslprofile_binding?**bulkbindings=yes**

NITRO allows you to fetch bindings in bulk.

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "sslcertkey_sslprofile_binding": [ {

"ca":<Boolean_value>,

"sslprofile":<String_value>,

"certkey":<String_value>

}]}
```

count

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/sslcertkey_sslprofile_binding/certkey_value<String>?**count=yes**

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{"sslcertkey_sslprofile_binding": [ { "__count": "#no"} ] }
```

sslcertkey_sslvserver_binding

Binding object showing the sslvserver that can be bound to sslcertkey.

Properties

(click to see Operations)

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
vserver	<Boolean>	Read-write	Specify this option to bind the certificate to an SSL virtual server. Note: The default option is -vServer.
ca	<Boolean>	Read-write	The certificate-key pair being unbound is a Certificate Authority (CA) certificate. If you choose this option, the certificate-key pair is unbound from the list of CA certificates that were bound to the

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
			specified SSL virtual server or SSL service.
vservername	<String>	Read-write	The name of the SSL virtual server name to which the certificate-key pair needs to be bound. Minimum length = 1
servername	<String>	Read-write	Vserver name to which the certificate key pair is bound.
certkey	<String>	Read-write	Name of the certificate-key pair. Minimum length = 1
version	<Integer>	Read-only	Version.
data	<Double>	Read-only	Vserver Id.
__count	<Double>	Read-only	count parameter

Operations

(click to see Properties)

GET| GET (ALL)| COUNT

Some options that you can use for each operations:

- **Getting warnings in response:**NITRO allows you to get warnings in an operation by specifying the "warning" query parameter as "yes". For example, to get warnings while connecting to the Citrix ADC appliance, the URL is as follows:

  http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/login?warning=yes

  If any, the warnings are displayed in the response payload with the HTTP code "209 X-NITRO-WARNING".

- **Authenticated access for individual NITRO operations:**NITRO allows you to logon to the Citrix ADC appliance to perform individual operations. You can use this option instead of creating a NITRO session (using the login object) and then using that session to perform all operations,

  To do this, you must specify the username and password in the request header of the NITRO request as follows:

  X-NITRO-USER:<username>

  X-NITRO-PASS:<password>

  Note: In such cases, make sure that the request header DOES not include the following:

  Cookie:NITRO_AUTH_TOKEN=<tokenvalue>

`*Note:`*

Mandatory parameters are marked in redand placeholder content is marked in <green>.

get

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/sslcertkey_sslvserver_binding/certkey_value<String>

Query-parameters:

filter

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/sslcertkey_sslvserver_binding/certkey_value<String>?**filter=property-name1:property-value1,property-name2:property-value2**

Use this query-parameter to get the filtered set of sslcertkey_sslvserver_binding resources configured on Citrix ADC. Filtering can be done on any of the properties of the resource.

pagination

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/sslcertkey_sslvserver_binding/certkey_value<String>?**pagesize=#no;pageno=#no**

Use this query-parameter to get the sslcertkey_sslvserver_binding resources in chunks.

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "sslcertkey_sslvserver_binding": [ {

"vserver":<Boolean_value>,

"ca":<Boolean_value>,

"vservername":<String_value>,

"servername":<String_value>,

"certkey":<String_value>,

"version":<Integer_value>,

"data":<Double_value>

}]}
```

get (all)

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/sslcertkey_sslvserver_binding

Query-parameters:

bulkbindings

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/sslcertkey_sslvserver_binding?**bulkbindings=yes**

NITRO allows you to fetch bindings in bulk.

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "sslcertkey_sslvserver_binding": [ {

"vserver":<Boolean_value>,

"ca":<Boolean_value>,

"vservername":<String_value>,

"servername":<String_value>,

"certkey":<String_value>,

"version":<Integer_value>,

"data":<Double_value>

}]}
```

count

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/sslcertkey_sslvserver_binding/certkey_value<String>?**count=yes**

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{"sslcertkey_sslvserver_binding": [ { "__count": "#no"} ] }
```

sslcertlink

Configuration for linked certificate resource.

Properties

(click to see Operations)

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
certkeyname	<String>	Read-only	Certificate key name.
linkcertkeyname	<String>	Read-only	Name of the Certificate-Authority.
__count	<Double>	Read-only	count parameter

Operations

(click to see Properties)

GET (ALL)| COUNT

Some options that you can use for each operations:

- **Getting warnings in response:**NITRO allows you to get warnings in an operation by specifying the "warning" query parameter as "yes". For example, to get warnings while connecting to the Citrix ADC appliance, the URL is as follows:

  http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/login?warning=yes

  If any, the warnings are displayed in the response payload with the HTTP code "209 X-NITRO-WARNING".

- **Authenticated access for individual NITRO operations:**NITRO allows you to logon to the Citrix ADC appliance to perform individual operations. You can use this option instead of creating a NITRO session (using the login object) and then using that session to perform all operations,

  To do this, you must specify the username and password in the request header of the NITRO request as follows:

  X-NITRO-USER:<username>

  X-NITRO-PASS:<password>

  Note: In such cases, make sure that the request header DOES not include the following:

  Cookie:NITRO_AUTH_TOKEN=<tokenvalue>

`*Note:`*

Mandatory parameters are marked in redand placeholder content is marked in <green>.

get (all)

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/sslcertlink

Query-parameters:

attrs

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/sslcertlink?**attrs=property-name1,property-name2**

Use this query parameter to specify the resource details that you want to retrieve.

filter

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/sslcertlink?**filter=property-name1:property-val1,property-name2:property-val2**

Use this query-parameter to get the filtered set of sslcertlink resources configured on Citrix ADC. Filtering can be done on any of the properties of the resource.

view

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/sslcertlink?**view=summary**

Note: By default, the retrieved results are displayed in detail view (?view=detail).

pagination

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/sslcertlink?**pagesize=#no;pageno=#no**

Use this query-parameter to get the sslcertlink resources in chunks.

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "sslcertlink": [ {
```

```
"certkeyname":<String_value>,

"linkcertkeyname":<String_value>

}]}
```

count

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/sslcertlink?**count=yes**

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "sslcertlink": [ { "__count": "#no"} ] }
```

sslcertreq

Configuration for certificate request resource.

Properties

(click to see Operations)

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
reqfile	<String>	Read-write	Name for and, optionally, path to the certificate signing request (CSR). /nsconfig/ssl/ is the default path. Maximum length = 63
keyfile	<String>	Read-write	Name of and, optionally, path to the private key used to create the certificate signing request, which then becomes part of the certificate-key pair. The private key can be either an RSA or a DSA key. The key must be present in the appliance's local storage. /nsconfig/ssl is the default path. Maximum length = 63
subjectaltname	<String>	Read-write	Subject Alternative Name (SAN) is an extension to X.509 that allows various values to be associated with a security certificate using a subjectAltName field. These values are called "Subject Alternative

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
			Names" (SAN). Names include: 1. Email addresses 2. IP addresses 3. URIs 4. DNS names (this is usually also provided as the Common Name RDN within the Subject field of the main certificate.) 5. Directory names (alternative Distinguished Names to that given in the Subject). Minimum length = 1
fipskeyname	<String>	Read-write	Name of the FIPS key used to create the certificate signing request. FIPS keys are created inside the Hardware Security Module of the FIPS card. Minimum length = 1 Maximum length = 31
keyform	<String>	Read-write	Format in which the key is stored on the appliance.

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
			Default value: PEM Possible values = DER, PEM
pempassphrase	<String>	Read-write	. Minimum length = 1 Maximum length = 31
countryname	<String>	Read-write	Two letter ISO code for your country. For example, US for United States. Minimum length = 2 Maximum length = 2
statename	<String>	Read-write	Full name of the state or province where your organization is located. Do not abbreviate. Minimum length = 1
organizationname	<String>	Read-write	Name of the organization that will use this certificate. The organization name (corporation, limited partnership, university, or government agency) must be registered with some authority at the national, state, or city level. Use the

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
			legal name under which the organization is registered. Do not abbreviate the organization name and do not use the following characters in the name: Angle brackets (< >) tilde (~), exclamation mark, at (@), pound (#), zero (0), caret (\^), asterisk (*), forward slash (/), square brackets ([ ]), question mark (?). Minimum length = 1
organizationunitname	<String>	Read-write	Name of the division or section in the organization that will use the certificate. Minimum length = 1
localityname	<String>	Read-write	Name of the city or town in which your organization's head office is located. Minimum length = 1
commonname	<String>	Read-write	Fully qualified domain name for the company or web site. The common name

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
			must match the name used by DNS servers to do a DNS lookup of your server. Most browsers use this information for authenticating the server's certificate during the SSL handshake. If the server name in the URL does not match the common name as given in the server certificate, the browser terminates the SSL handshake or prompts the user with a warning message. Do not use wildcard characters, such as asterisk (*) or question mark (?), and do not use an IP address as the common name. The common name must not contain the protocol specifier <http://> or <https://>. Minimum length = 1
emailaddress	<String>	Read-write	Contact person's e-mail address. This address is publically displayed as part of

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
			the certificate. Provide an e-mail address that is monitored by an administrator who can be contacted about the certificate. Minimum length = 1
challengepassword	<String>	Read-write	Pass phrase, embedded in the certificate signing request that is shared only between the client or server requesting the certificate and the SSL certificate issuer (typically the certificate authority). This pass phrase can be used to authenticate a client or server that is requesting a certificate from the certificate authority. Minimum length = 4
companyname	<String>	Read-write	Additional name for the company or web site. Minimum length = 1
digestmethod	<String>	Read-write	Digest algorithm used in creating CSR. Possible values = SHA1, SHA256

Operations

(click to see Properties)

CREATE

Some options that you can use for each operations:

- **Getting warnings in response:**NITRO allows you to get warnings in an operation by specifying the "warning" query parameter as "yes". For example, to get warnings while connecting to the Citrix ADC appliance, the URL is as follows:

  http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/login?warning=yes

  If any, the warnings are displayed in the response payload with the HTTP code "209 X-NITRO-WARNING".

- **Authenticated access for individual NITRO operations:**NITRO allows you to logon to the Citrix ADC appliance to perform individual operations. You can use this option instead of creating a NITRO session (using the login object) and then using that session to perform all operations,

  To do this, you must specify the username and password in the request header of the NITRO request as follows:

  X-NITRO-USER:<username>

  X-NITRO-PASS:<password>

  Note: In such cases, make sure that the request header DOES not include the following:

  Cookie:NITRO_AUTH_TOKEN=<tokenvalue>

`*Note:`*

Mandatory parameters are marked in redand placeholder content is marked in <green>.

create

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/sslcertreq?**action=create**

HTTP Method:POST

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Content-Type:application/json

Request Payload:

```
{"sslcertreq":{
<b>"reqfile":<String_value>,
</b>"keyfile":<String_value>,
"subjectaltname":<String_value>,
"fipskeyname":<String_value>,
"keyform":<String_value>,
"pempassphrase":<String_value>,
<b>"countryname":<String_value>,
</b><b>"statename":<String_value>,
</b><b>"organizationname":<String_value>,
</b>"organizationunitname":<String_value>,
"localityname":<String_value>,
"commonname":<String_value>,
"emailaddress":<String_value>,
"challengepassword":<String_value>,
"companyname":<String_value>,
"digestmethod":<String_value>
}}
```

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error

sslcipher

Configuration for cipher resource.

Properties

(click to see Operations)

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
ciphergroupname	<String>	Read-write	Name for the user-defined cipher group. Must begin with an ASCII alphanumeric or underscore (_) character, and must contain only ASCII alphanumeric, underscore, hash (#), period (.), space, colon (:), at (@), equals (=), and hyphen (-) characters. Cannot be changed after the cipher group is created. The following requirement applies only to the Citrix ADC CLI: If the name includes one or more spaces, enclose the name in double or single quotation marks (for example,

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
			"my ciphergroup" or 'my ciphergroup'). Minimum length = 1
ciphgrpalias	<String>	Read-write	The individual cipher name(s), a user-defined cipher group, or a system predefined cipher alias that will be added to the predefined cipher alias that will be added to the group cipherGroupName. If a cipher alias or a cipher group is specified, all the individual ciphers in the cipher alias or group will be added to the user-defined cipher group. Minimum length = 1
ciphername	<String>	Read-write	Cipher name.
cipherpriority	<Double>	Read-write	This indicates priority assigned to the particular cipher. Minimum value = 1
sslprofile	<String>	Read-write	Name of the profile to which cipher is attached.
__count	<Double>	Read-only	count parameter

Operations

(click to see Properties)

ADD| UPDATE| UNSET| DELETE| GET (ALL)| GET| COUNT

Some options that you can use for each operations:

- **Getting warnings in response:**NITRO allows you to get warnings in an operation by specifying the "warning" query parameter as "yes". For example, to get warnings while connecting to the Citrix ADC appliance, the URL is as follows:

  http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/login?warning=yes

  If any, the warnings are displayed in the response payload with the HTTP code "209 X-NITRO-WARNING".

- **Authenticated access for individual NITRO operations:**NITRO allows you to logon to the Citrix ADC appliance to perform individual operations. You can use this option instead of creating a NITRO session (using the login object) and then using that session to perform all operations,

  To do this, you must specify the username and password in the request header of the NITRO request as follows:

  X-NITRO-USER:<username>

  X-NITRO-PASS:<password>

  Note: In such cases, make sure that the request header DOES not include the following:

  Cookie:NITRO_AUTH_TOKEN=<tokenvalue>

`*Note:`*

Mandatory parameters are marked in redand placeholder content is marked in <green>.

add

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/sslcipher

HTTP Method:POST

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Content-Type:application/json

Request Payload:

```
{"sslcipher":{

<b>"ciphergroupname":<String_value>,

</b>"ciphgrpalias":<String_value>

}}
```

Response:

HTTP Status Code on Success: 201 Created HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error

update

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/sslcipher

HTTP Method:PUT

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Content-Type:application/json

Request Payload:

```
{"sslcipher":{

<b>"ciphergroupname":<String_value>,

</b>"ciphername":<String_value>,

"cipherpriority":<Double_value>

}}
```

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error

unset

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/sslcipher?**action=unset**

HTTP Method:POST

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Content-Type:application/json

Request Payload:

```
{"sslcipher":{

<b>"ciphergroupname":<String_value>,

</b>"ciphername":true,

"cipherpriority":true

}}
```

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error

delete

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/sslcipher/ciphergroupname_value<String>

HTTP Method:DELETE

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue>

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error

get (all)

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/sslcipher

Query-parameters:

args

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/sslcipher?**args=ciphergroupname:<String_value>,sslprofile:<String_value>**

Use this query-parameter to get sslcipher resources based on additional properties.

attrs

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/sslcipher?**attrs=property-name1,property-name2**

Use this query parameter to specify the resource details that you want to retrieve.

filter

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/sslcipher?**filter=property-name1:property-val1,property-name2:property-val2**

Use this query-parameter to get the filtered set of sslcipher resources configured on Citrix ADC. Filtering can be done on any of the properties of the resource.

view

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/sslcipher?**view=summary**

Note: By default, the retrieved results are displayed in detail view (?view=detail).

pagination

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/sslcipher?**pagesize=#no;pageno=#no**

Use this query-parameter to get the sslcipher resources in chunks.

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "sslcipher": [ {

ciphergroupname:<String_value>,sslprofile:<String_value>"ciphername":<String_value>,

"cipherpriority":<Double_value>

}]}
```

get

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/sslcipher/ciphergroupname_value<String>

Query-parameters:

attrs

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/sslcipher/ciphergroupname_value<String>?**attrs=property-name1,property-name2**

Use this query parameter to specify the resource details that you want to retrieve.

view

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/sslcipher/ciphergroupname_value<String>?**view=summary**

Note: By default, the retrieved results are displayed in detail view (?view=detail).

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "sslcipher": [ {

ciphergroupname:<String_value>,sslprofile:<String_value>"ciphername":<Stri
ng_value>,

"cipherpriority":<Double_value>

}]}
```

count

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/sslcipher?**count=yes**

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "sslcipher": [ { "__count": "#no"} ] }
```

sslciphersuite

Configuration for ciphersuite resource.

Properties

(click to see Operations)

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
ciphername	<String>	Read-write	Name of the cipher suite for which to show detailed information.
description	<String>	Read-only	Cipher suite description.
__count	<Double>	Read-only	count parameter

Operations

(click to see Properties)

GET (ALL)| GET| COUNT

Some options that you can use for each operations:

- **Getting warnings in response:**NITRO allows you to get warnings in an operation by specifying the "warning" query parameter as "yes". For example, to get warnings while connecting to the Citrix ADC appliance, the URL is as follows:

  http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/login?warning=yes

  If any, the warnings are displayed in the response payload with the HTTP code "209 X-NITRO-WARNING".

- **Authenticated access for individual NITRO operations:**NITRO allows you to logon to the Citrix ADC appliance to perform individual operations. You can use this option instead of creating a NITRO session (using the login object) and then using that session to perform all operations,

  To do this, you must specify the username and password in the request header of the NITRO request as follows:

  X-NITRO-USER:<username>

  X-NITRO-PASS:<password>

  Note: In such cases, make sure that the request header DOES not include the following:

  Cookie:NITRO_AUTH_TOKEN=<tokenvalue>

`*Note:`*

Mandatory parameters are marked in redand placeholder content is marked in <green>.

get (all)

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/sslciphersuite

Query-parameters:

attrs

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/sslciphersuite?**attrs=property-name1,property-name2**

Use this query parameter to specify the resource details that you want to retrieve.

filter

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/sslciphersuite?**filter=property-name1:property-val1,property-name2:property-val2**

Use this query-parameter to get the filtered set of sslciphersuite resources configured on Citrix ADC. Filtering can be done on any of the properties of the resource.

view

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/sslciphersuite?**view=summary**

Note: By default, the retrieved results are displayed in detail view (?view=detail).

pagination

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/sslciphersuite?**pagesize=#no;pageno=#no**

Use this query-parameter to get the sslciphersuite resources in chunks.

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "sslciphersuite": [ {

"ciphername":<String_value>,

"description":<String_value>

}]}
```

get

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/sslciphersuite/ciphername_value<String>

Query-parameters:

attrs

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/sslciphersuite/ciphername_value<String>?**attrs=property-name1,property-name2**

Use this query parameter to specify the resource details that you want to retrieve.

view

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/sslciphersuite/ciphername_value<String>?**view=summary**

Note: By default, the retrieved results are displayed in detail view (?view=detail).

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "sslciphersuite": [ {

"ciphername":<String_value>,

"description":<String_value>
```

```
}]}
```

count

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/sslciphersuite?**count=yes**

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "sslciphersuite": [ { "__count": "#no"} ] }
```

sslcipher_binding

Binding object which returns the resources bound to sslcipher.

Properties

(click to see Operations)

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
ciphergroupname	<String>	Read-write	Name of the cipher group for which to show detailed information. Minimum length = 1

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
sslcipher_sslprofile_binding	<sslcipher_sslprofile_binding[]>	Read-only	sslprofile that can be bound to sslcipher.
sslcipher_individualcipher_binding	<sslcipher_individualcipher_binding[]>	Read-only	individualcipher that can be bound to sslcipher.
sslcipher_sslciphersuite_binding	<sslcipher_sslciphersuite_binding[]>	Read-only	sslciphersuite that can be bound to sslcipher.

Operations

(click to see Properties)

GET| GET (ALL)

Some options that you can use for each operations:

- **Getting warnings in response:**NITRO allows you to get warnings in an operation by specifying the "warning" query parameter as "yes". For example, to get warnings while connecting to the Citrix ADC appliance, the URL is as follows:

  http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/login?warning=yes

  If any, the warnings are displayed in the response payload with the HTTP code "209 X-NITRO-WARNING".

- **Authenticated access for individual NITRO operations:**NITRO allows you to logon to the Citrix ADC appliance to perform individual operations. You can use this option instead of creating a NITRO session (using the login object) and then using that session to perform all operations,

  To do this, you must specify the username and password in the request header of the NITRO request as follows:

  X-NITRO-USER:<username>

  X-NITRO-PASS:<password>

  Note: In such cases, make sure that the request header DOES not include the following:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue>

`*Note:` *

Mandatory parameters are marked in redand placeholder content is marked in <green>.

get

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/sslcipher_binding/ciphergroupname_value<String>

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "sslcipher_binding": [ {

"ciphergroupname":<String_value>,

"sslcipher_sslciphersuite_binding":<sslcipher_sslciphersuite_binding[]_value>,

"sslcipher_individualcipher_binding":<sslcipher_individualcipher_binding[]_value>,

"sslcipher_sslprofile_binding":<sslcipher_sslprofile_binding[]_value>

}]}
```

get (all)

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/sslcipher_binding

Query-parameters:

bulkbindings

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/sslcipher_binding?**bulkbindings=yes**

NITRO allows you to fetch bindings in bulk.

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "sslcipher_binding": [ {

"ciphergroupname":<String_value>,

"sslcipher_sslciphersuite_binding":<sslcipher_sslciphersuite_binding[]_val
ue>,

"sslcipher_individualcipher_binding":<sslcipher_individualcipher_binding[]
_value>,

"sslcipher_sslprofile_binding":<sslcipher_sslprofile_binding[]_value>

}]}
```

sslcipher_individualcipher_binding

Binding object showing the individualcipher that can be bound to sslcipher.

Properties

(click to see Operations)

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
ciphername	<String>	Read-write	Cipher name.
ciphgrpals	<String>	Read-write	A cipher-suite can consist of an individual cipher name, the system predefined cipher-alias name, or user defined cipher-group name. Minimum length = 1
cipherpriority	<Double>	Read-write	This indicates priority assigned to the particular cipher. Minimum value = 1
description	<String>	Read-write	Cipher suite description.
ciphergroupname	<String>	Read-write	Name of the user-defined cipher group. Minimum length = 1
cipheroperation	<String>	Read-write	The operation that is performed when adding the cipher-suite. Possible cipher operations are: ADD - Appends the given cipher-suite to the existing one configured for the virtual server. REM -

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
			Removes the given cipher-suite from the existing one configured for the virtual server. ORD - Overrides the current configured cipher-suite for the virtual server with the given cipher-suite. Default value: 0 Possible values = ADD, REM, ORD
__count	<Double>	Read-write	count parameter

Operations

(click to see Properties)

GET| GET (ALL)| COUNT

Some options that you can use for each operations:

- **Getting warnings in response:**NITRO allows you to get warnings in an operation by specifying the "warning" query parameter as "yes". For example, to get warnings while connecting to the Citrix ADC appliance, the URL is as follows:

  http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/login?warning=yes

  If any, the warnings are displayed in the response payload with the HTTP code "209 X-NITRO-WARNING".

- **Authenticated access for individual NITRO operations:**NITRO allows you to logon to the Citrix ADC appliance to perform individual operations. You can use this option instead of creating a NITRO session (using the login object) and then using that session to perform all operations,

  To do this, you must specify the username and password in the request header of the NITRO request as follows:

X-NITRO-USER:<username>

X-NITRO-PASS:<password>

Note: In such cases, make sure that the request header DOES not include the following:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue>

`*Note:`*

Mandatory parameters are marked in redand placeholder content is marked in <green>.

get

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/sslcipher_individualcipher_binding/ciphergroupname_value<String>

Query-parameters:

filter

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/sslcipher_individualcipher_binding/ciphergroupname_value<String>?**filter=property-name1:property-value1,property-name2:property-value2**

Use this query-parameter to get the filtered set of sslcipher_individualcipher_binding resources configured on Citrix ADC. Filtering can be done on any of the properties of the resource.

pagination

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/sslcipher_individualcipher_binding/ciphergroupname_value<String>?**pagesize=#no;pageno=#no**

Use this query-parameter to get the sslcipher_individualcipher_binding resources in chunks.

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "sslcipher_individualcipher_binding": [ {

"ciphername":<String_value>,

"ciphgrpals":<String_value>,

"cipherpriority":<Double_value>,

"description":<String_value>,

"ciphergroupname":<String_value>,

"cipheroperation":<String_value>

}]}
```

get (all)

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/sslcipher_individualcipher_binding

Query-parameters:

bulkbindings

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/sslcipher_individualcipher_binding?**bulkbindings=yes**

NITRO allows you to fetch bindings in bulk.

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "sslcipher_individualcipher_binding": [ {

"ciphername":<String_value>,

"ciphgrpals":<String_value>,

"cipherpriority":<Double_value>,

"description":<String_value>,

"ciphergroupname":<String_value>,

"cipheroperation":<String_value>

}]}
```

count

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/sslcipher_individualcipher_binding/ciphergroupname_value<String>?**count=yes**

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{"sslcipher_individualcipher_binding": [ { "__count": "#no"} ] }
```

sslcipher_servicegroup_binding

Binding object showing the servicegroup that can be bound to sslcipher.

Properties

(click to see Operations)

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
cipherpriority	<Double>	Read-write	Priority of the cipher to be added. Minimum value = 1 Maximum value = 1000
ciphgrpals	<String>	Read-write	A cipher-suite can consist of an individual cipher name, the system predefined cipher-alias name, or user defined cipher-group name. Minimum length = 1
servicegroupname	<String>	Read-write	The name of the SSL service name to which the cipher-suite is to be bound. Minimum length = 1

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
servicegroup	<Boolean>	Read-write	Indicates that the cipher operation is to be performed on the named SSL service or service group.
service	<Boolean>	Read-write	Indicates that the cipher operation is to be performed on the named SSL service or service group.
servicename	<String>	Read-write	The name of the SSL service name to which the cipher-suite is to be bound. Minimum length = 1
ciphergroupname	<String>	Read-write	Name of the user-defined cipher group. Minimum length = 1
cipheroperation	<String>	Read-write	The operation that is performed when adding the cipher-suite. Possible cipher operations are: ADD - Appends the given cipher-suite to the existing one configured for the virtual server. REM - Removes the given cipher-suite from the existing one configured for the virtual server. ORD - Overrides the current configured cipher-suite for the virtual

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
			server with the given cipher-suite. Default value: 0 Possible values = ADD, REM, ORD

Operations

(click to see Properties)

Some options that you can use for each operations:

- **Getting warnings in response:**NITRO allows you to get warnings in an operation by specifying the "warning" query parameter as "yes". For example, to get warnings while connecting to the Citrix ADC appliance, the URL is as follows:

  http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/login?warning=yes

  If any, the warnings are displayed in the response payload with the HTTP code "209 X-NITRO-WARNING".

- **Authenticated access for individual NITRO operations:**NITRO allows you to logon to the Citrix ADC appliance to perform individual operations. You can use this option instead of creating a NITRO session (using the login object) and then using that session to perform all operations,

  To do this, you must specify the username and password in the request header of the NITRO request as follows:

  X-NITRO-USER:<username>

  X-NITRO-PASS:<password>

  Note: In such cases, make sure that the request header DOES not include the following:

  Cookie:NITRO_AUTH_TOKEN=<tokenvalue>

`*Note:`*

Mandatory parameters are marked in redand placeholder content is marked in <green>.

sslcipher_service_binding

Binding object showing the service that can be bound to sslcipher.

Properties

(click to see Operations)

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
cipherpriority	<Double>	Read-write	Priority of the cipher to be added. Minimum value = 1 Maximum value = 1000
ciphgrpals	<String>	Read-write	A cipher-suite can consist of an individual cipher name, the system predefined cipher-alias name, or user defined cipher-group name. Minimum length = 1
servicegroupname	<String>	Read-write	The name of the SSL service name to which the cipher-suite is to be bound. Minimum length = 1
servicegroup	<Boolean>	Read-write	Indicates that the cipher operation is to be performed on the named SSL service or service group.
service	<Boolean>	Read-write	Indicates that the cipher operation is to be performed on the

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
			named SSL service or service group.
servicename	<String>	Read-write	The name of the SSL service name to which the cipher-suite is to be bound. Minimum length = 1
ciphergroupname	<String>	Read-write	Name of the user-defined cipher group. Minimum length = 1
cipheroperation	<String>	Read-write	The operation that is performed when adding the cipher-suite. Possible cipher operations are: ADD - Appends the given cipher-suite to the existing one configured for the virtual server. REM - Removes the given cipher-suite from the existing one configured for the virtual server. ORD - Overrides the current configured cipher-suite for the virtual server with the given cipher-suite. Default value: 0 Possible values = ADD, REM, ORD

Operations

(click to see Properties)

Some options that you can use for each operations:

- **Getting warnings in response:**NITRO allows you to get warnings in an operation by specifying the "warning" query parameter as "yes". For example, to get warnings while connecting to the Citrix ADC appliance, the URL is as follows:

  http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/login?warning=yes

  If any, the warnings are displayed in the response payload with the HTTP code "209 X-NITRO-WARNING".

- **Authenticated access for individual NITRO operations:**NITRO allows you to logon to the Citrix ADC appliance to perform individual operations. You can use this option instead of creating a NITRO session (using the login object) and then using that session to perform all operations,

  To do this, you must specify the username and password in the request header of the NITRO request as follows:

  X-NITRO-USER:<username>

  X-NITRO-PASS:<password>

  Note: In such cases, make sure that the request header DOES not include the following:

  Cookie:NITRO_AUTH_TOKEN=<tokenvalue>

`*Note:`*

Mandatory parameters are marked in redand placeholder content is marked in <green>.

sslcipher_sslciphersuite_binding

Binding object showing the sslciphersuite that can be bound to sslcipher.

Properties

(click to see Operations)

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
ciphername	<String>	Read-write	Cipher name.
ciphgrpals	<String>	Read-write	A cipher-suite can consist of an

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
			individual cipher name, the system predefined cipher-alias name, or user defined cipher-group name. Minimum length = 1
cipherpriority	<Double>	Read-write	This indicates priority assigned to the particular cipher. Minimum value = 1
description	<String>	Read-write	Cipher suite description.
ciphergroupname	<String>	Read-write	Name for the user-defined cipher group. Must begin with an ASCII alphanumeric or underscore (_) character, and must contain only ASCII alphanumeric, underscore, hash (#), period (.), space, colon (:), at (@), equals (=), and hyphen (-) characters. Cannot be changed after the cipher group is created. The following requirement applies only to the Citrix ADC CLI: If the name includes one or more spaces, enclose the name in double or single

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
			quotation marks (for example, "my ciphergroup" or 'my ciphergroup'). Minimum length = 1
cipheroperation	<String>	Read-write	The operation that is performed when adding the cipher-suite. Possible cipher operations are: ADD - Appends the given cipher-suite to the existing one configured for the virtual server. REM - Removes the given cipher-suite from the existing one configured for the virtual server. ORD - Overrides the current configured cipher-suite for the virtual server with the given cipher-suite. Default value: 0 Possible values = ADD, REM, ORD
__count	<Double>	Read-write	count parameter

Operations

(click to see Properties)

ADD:| DELETE:| GET| GET (ALL)| COUNT

Some options that you can use for each operations:

- **Getting warnings in response:**NITRO allows you to get warnings in an operation by specifying the "warning" query parameter as "yes". For example, to get warnings while connecting to the Citrix ADC appliance, the URL is as follows:

  http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/login?warning=yes

  If any, the warnings are displayed in the response payload with the HTTP code "209 X-NITRO-WARNING".

- **Authenticated access for individual NITRO operations:**NITRO allows you to logon to the Citrix ADC appliance to perform individual operations. You can use this option instead of creating a NITRO session (using the login object) and then using that session to perform all operations,

  To do this, you must specify the username and password in the request header of the NITRO request as follows:

  X-NITRO-USER:<username>

  X-NITRO-PASS:<password>

  Note: In such cases, make sure that the request header DOES not include the following:

  Cookie:NITRO_AUTH_TOKEN=<tokenvalue>

`*Note:`*

Mandatory parameters are marked in redand placeholder content is marked in <green>.

add

URL:http://<Citrix-ADC-IP-address(NSIP)/nitro/v1/config/sslcipher_sslciphersuite_binding

HTTP Method:PUT

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Content-Type:application/json

Request Payload:

```
{
```

```
"sslcipher_sslciphersuite_binding":{

"ciphergroupname":<String_value>,

"cipheroperation":<String_value>,

"ciphgrpals":<String_value>,

"ciphername":<String_value>,

"cipherpriority":<Double_value>

}}
```

Response:

HTTP Status Code on Success: 201 Created HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error

delete

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/sslcipher_sslciphersuite_binding

Query-parameters:

args

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/sslcipher_sslciphersuite_binding?**args=ciphergroupname:<String_value>,ciphername:<String_value>**

HTTP Method:DELETE

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue>

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error

get

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/sslcipher_sslciphersuite_binding

Query-parameters:

filter

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/sslcipher_sslciphersuite_binding?**filter=property-name1:property-value1,property-name2:property-value2**

Use this query-parameter to get the filtered set of sslcipher_sslciphersuite_binding resources configured on Citrix ADC. Filtering can be done on any of the properties of the resource.

pagination

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/sslcipher_sslciphersuite_binding?**pagesize=#no;pageno=#no**

Use this query-parameter to get the sslcipher_sslciphersuite_binding resources in chunks.

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "sslcipher_sslciphersuite_binding": [ {

"ciphername":<String_value>,

"ciphgrpals":<String_value>,
```

```
"cipherpriority":<Double_value>,

"description":<String_value>,

"ciphergroupname":<String_value>,

"cipheroperation":<String_value>

}]}
```

get (all)

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/sslcipher_sslciphersuite_binding

Query-parameters:

bulkbindings

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/sslcipher_sslciphersuite_binding?**bulkbindings=yes**

NITRO allows you to fetch bindings in bulk.

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "sslcipher_sslciphersuite_binding": [ {
```

```
"ciphername":<String_value>,

"ciphgrpals":<String_value>,

"cipherpriority":<Double_value>,

"description":<String_value>,

"ciphergroupname":<String_value>,

"cipheroperation":<String_value>

}]}
```

count

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/sslcipher_sslciphersuite_binding?**args=;count=yes**

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "sslcipher_sslciphersuite_binding": [ { "__count": "#no"} ] }
```

sslcipher_sslprofile_binding

Binding object showing the sslprofile that can be bound to sslcipher.

Properties

(click to see Operations)

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
cipherpriority	<Double>	Read-write	Priority of the cipher to be added. Minimum value = 1 Maximum value = 1000
ciphgrpals	<String>	Read-write	A cipher-suite can consist of an individual cipher name, the system predefined cipher-alias name, or user defined cipher-group name. Minimum length = 1
description	<String>	Read-write	Cipher suite description.
sslprofile	<String>	Read-write	Name of the profile to which cipher is attached.
ciphergroupname	<String>	Read-write	Name of the user-defined cipher group. Minimum length = 1
cipheroperation	<String>	Read-write	The operation that is performed when adding the cipher-suite. Possible cipher operations are: ADD - Appends the given

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
			cipher-suite to the existing one configured for the virtual server. REM - Removes the given cipher-suite from the existing one configured for the virtual server. ORD - Overrides the current configured cipher-suite for the virtual server with the given cipher-suite. Default value: 0 Possible values = ADD, REM, ORD
__count	<Double>	Read-write	count parameter

Operations

(click to see Properties)

GET| GET (ALL)| COUNT

Some options that you can use for each operations:

- **Getting warnings in response:**NITRO allows you to get warnings in an operation by specifying the "warning" query parameter as "yes". For example, to get warnings while connecting to the Citrix ADC appliance, the URL is as follows:

  http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/login?warning=yes

  If any, the warnings are displayed in the response payload with the HTTP code "209 X-NITRO-WARNING".

- **Authenticated access for individual NITRO operations:**NITRO allows you to logon to the Citrix ADC appliance to perform individual operations. You can use this option instead of

creating a NITRO session (using the login object) and then using that session to perform all operations,

To do this, you must specify the username and password in the request header of the NITRO request as follows:

X-NITRO-USER:<username>

X-NITRO-PASS:<password>

Note: In such cases, make sure that the request header DOES not include the following:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue>

`*Note:`*

Mandatory parameters are marked in redand placeholder content is marked in <green>.

get

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/sslcipher_sslprofile_binding/ciphergroupname_value<String>

Query-parameters:

filter

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/sslcipher_sslprofile_binding/ciphergroupname_value<String>?**filter=property-name1:property-value1,property-name2:property-value2**

Use this query-parameter to get the filtered set of sslcipher_sslprofile_binding resources configured on Citrix ADC. Filtering can be done on any of the properties of the resource.

pagination

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/sslcipher_sslprofile_binding/ciphergroupname_value<String>?**pagesize=#no;pageno=#no**

Use this query-parameter to get the sslcipher_sslprofile_binding resources in chunks.

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "sslcipher_sslprofile_binding": [ {

"cipherpriority":<Double_value>,

"ciphgrpals":<String_value>,

"description":<String_value>,

"sslprofile":<String_value>,

"ciphergroupname":<String_value>,

"cipheroperation":<String_value>

}]}
```

get (all)

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/sslcipher_sslprofile_binding

Query-parameters:

bulkbindings

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/sslcipher_sslprofile_binding?**bulkbindings=yes**

NITRO allows you to fetch bindings in bulk.

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "sslcipher_sslprofile_binding": [ {

"cipherpriority":<Double_value>,

"ciphgrpals":<String_value>,

"description":<String_value>,

"sslprofile":<String_value>,

"ciphergroupname":<String_value>,

"cipheroperation":<String_value>

}]}
```

count

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/sslcipher_sslprofile_binding/ciphergroupname_value<String>?**count=yes**

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{"sslcipher_sslprofile_binding": [ { "__count": "#no"} ] }
```

sslcipher_sslvserver_binding

Binding object showing the sslvserver that can be bound to sslcipher.

Properties

(click to see Operations)

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
cipherpriority	<Double>	Read-write	Priority of the cipher to be added. Minimum value = 1 Maximum value = 1000
ciphgrpals	<String>	Read-write	A cipher-suite can consist of an individual cipher name, the system predefined cipher-alias name, or user defined cipher-group name. Minimum length = 1

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
vserver	<Boolean>	Read-write	Select the -vServer flag when the cipher operation is performed on an SSL virtual server. Note: By default the bind ssl cipher command internally assumes the flag of -vServer argument. Hence, while working with the SSL vserver, you need not specify this flag.
vservername	<String>	Read-write	The name of the SSL virtual server to which the cipher-suite is to be bound. Minimum length = 1
ciphergroupname	<String>	Read-write	Name of the user-defined cipher group. Minimum length = 1
cipheroperation	<String>	Read-write	The operation that is performed when adding the cipher-suite. Possible cipher operations are: ADD - Appends the given cipher-suite to the existing one configured for the virtual server. REM - Removes the given cipher-suite from the existing one configured for the virtual server. ORD - Overrides the current

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
			configured cipher-suite for the virtual server with the given cipher-suite. Default value: 0 Possible values = ADD, REM, ORD

Operations

(click to see Properties)

Some options that you can use for each operations:

- **Getting warnings in response:**NITRO allows you to get warnings in an operation by specifying the "warning" query parameter as "yes". For example, to get warnings while connecting to the Citrix ADC appliance, the URL is as follows:

  http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/login?warning=yes

  If any, the warnings are displayed in the response payload with the HTTP code "209 X-NITRO-WARNING".

- **Authenticated access for individual NITRO operations:**NITRO allows you to logon to the Citrix ADC appliance to perform individual operations. You can use this option instead of creating a NITRO session (using the login object) and then using that session to perform all operations,

  To do this, you must specify the username and password in the request header of the NITRO request as follows:

  X-NITRO-USER:<username>

  X-NITRO-PASS:<password>

  Note: In such cases, make sure that the request header DOES not include the following:

  Cookie:NITRO_AUTH_TOKEN=<tokenvalue>

`*Note:` *

Mandatory parameters are marked in redand placeholder content is marked in <green>.

sslcrl

Configuration for Certificate Revocation List resource.

Properties

(click to see Operations)

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
crlname	<String>	Read-write	Name for the Certificate Revocation List (CRL). Must begin with an ASCII alphanumeric or underscore (_) character, and must contain only ASCII alphanumeric, underscore, hash (#), period (.), space, colon (:), at (@), equals (=), and hyphen (-) characters. Cannot be changed after the CRL is created. The following requirement applies only to the Citrix ADC CLI: If the name includes one or more spaces, enclose the name in double or single quotation marks (for example, "my crl" or 'my crl'). Minimum length = 1

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
crlpath	<String>	Read-write	Path to the CRL file. /var/netscaler/ssl/ is the default path. Minimum length = 1
inform	<String>	Read-write	Input format of the CRL file. The two formats supported on the appliance are: PEM - Privacy Enhanced Mail. DER - Distinguished Encoding Rule. Default value: PEM Possible values = DER, PEM
refresh	<String>	Read-write	Set CRL auto refresh. Possible values = ENABLED, DISABLED
cacert	<String>	Read-write	CA certificate that has issued the CRL. Required if CRL Auto Refresh is selected. Install the CA certificate on the appliance before adding the CRL. Minimum length = 1
method	<String>	Read-write	Method for CRL refresh. If LDAP is selected, specify the

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
			method, CA certificate, base DN, port, and LDAP server name. If HTTP is selected, specify the CA certificate, method, URL, and port. Cannot be changed after a CRL is added. Possible values = HTTP, LDAP
server	<String>	Read-write	IP address of the LDAP server from which to fetch the CRLs. Minimum length = 1
url	<String>	Read-write	URL of the CRL distribution point.
port	<Integer>	Read-write	Port for the LDAP server. Minimum value = 1
basedn	<String>	Read-write	Base distinguished name (DN), which is used in an LDAP search to search for a CRL. Citrix recommends searching for the Base DN instead of the Issuer Name from the CA certificate, because the Issuer Name field might not exactly match the LDAP

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
			directory structure's DN. Minimum length = 1
scope	<String>	Read-write	Extent of the search operation on the LDAP server. Available settings function as follows: One - One level below Base DN. Base - Exactly the same level as Base DN. Default value: One Possible values = Base, One
interval	<String>	Read-write	CRL refresh interval. Use the NONE setting to unset this parameter. Possible values = MONTHLY, WEEKLY, DAILY, NONE
day	<Integer>	Read-write	Day on which to refresh the CRL, or, if the Interval parameter is not set, the number of days after which to refresh the CRL. If Interval is set to MONTHLY, specify the date. If Interval is set to

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
			WEEKLY, specify the day of the week (for example, Sun=0 and Sat=6). This parameter is not applicable if the Interval is set to DAILY. Minimum value = 0 Maximum value = 31
time	<String>	Read-write	Time, in hours (1-24) and minutes (1-60), at which to refresh the CRL.
binddn	<String>	Read-write	Bind distinguished name (DN) to be used to access the CRL object in the LDAP repository if access to the LDAP repository is restricted or anonymous access is not allowed. Minimum length = 1
password	<String>	Read-write	Password to access the CRL in the LDAP repository if access to the LDAP repository is restricted or anonymous access is not allowed. Minimum length = 1

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
binary	<String>	Read-write	Set the LDAP-based CRL retrieval mode to binary. Default value: NO Possible values = YES, NO
cacertfile	<String>	Read-write	Name of and, optionally, path to the CA certificate file. /nsconfig/ssl/ is the default path. Maximum length = 63
cakeyfile	<String>	Read-write	Name of and, optionally, path to the CA key file. /nsconfig/ssl/ is the default path. Maximum length = 63
indexfile	<String>	Read-write	Name of and, optionally, path to the file containing the serial numbers of all the certificates that are revoked. Revoked certificates are appended to the file. /nsconfig/ssl/ is the default path. Maximum length = 63
revoke	<String>	Read-write	Name of and, optionally, path to the

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
			certificate to be revoked. /nsconfig/ssl/ is the default path. Maximum length = 63
gencrl	<String>	Read-write	Name of and, optionally, path to the CRL file to be generated. The list of certificates that have been revoked is obtained from the index file. /nsconfig/ssl/ is the default path. Maximum length = 63
flags	<Integer>	Read-only	CRL status flag.
lastupdatetime	<Integer>	Read-only	Last CRL refresh time.
version	<Integer>	Read-only	CRL version.
signaturealgo	<String>	Read-only	Signature algorithm.
issuer	<String>	Read-only	Issuer name.
lastupdate	<String>	Read-only	Last update time.
nextupdate	<String>	Read-only	Next update time.
daystoexpiration	<Integer>	Read-only	Number of days remaining for the CRL to expire.
__count	<Double>	Read-only	count parameter

Operations

(click to see Properties)

ADD| CREATE| DELETE| UPDATE| UNSET| GET (ALL)| GET| COUNT

Some options that you can use for each operations:

- **Getting warnings in response:**NITRO allows you to get warnings in an operation by specifying the "warning" query parameter as "yes". For example, to get warnings while connecting to the Citrix ADC appliance, the URL is as follows:

  http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/login?warning=yes

  If any, the warnings are displayed in the response payload with the HTTP code "209 X-NITRO-WARNING".

- **Authenticated access for individual NITRO operations:**NITRO allows you to logon to the Citrix ADC appliance to perform individual operations. You can use this option instead of creating a NITRO session (using the login object) and then using that session to perform all operations,

  To do this, you must specify the username and password in the request header of the NITRO request as follows:

  X-NITRO-USER:<username>

  X-NITRO-PASS:<password>

  Note: In such cases, make sure that the request header DOES not include the following:

  Cookie:NITRO_AUTH_TOKEN=<tokenvalue>

`*Note:`*

Mandatory parameters are marked in redand placeholder content is marked in <green>.

add

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/sslcrl

HTTP Method:POST

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Content-Type:application/json

Request Payload:

```
{"sslcrl":{

<b>"crlname":<String_value>,

</b><b>"crlpath":<String_value>,

</b>"inform":<String_value>,

"refresh":<String_value>,

"cacert":<String_value>,

"method":<String_value>,

"server":<String_value>,

"url":<String_value>,

"port":<Integer_value>,

"basedn":<String_value>,

"scope":<String_value>,

"interval":<String_value>,

"day":<Integer_value>,

"time":<String_value>,

"binddn":<String_value>,

"password":<String_value>,

"binary":<String_value>

}}
```

Response:

HTTP Status Code on Success: 201 Created HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error

create

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/sslcrl?**action=create**

HTTP Method:POST

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Content-Type:application/json

Request Payload:

```
{"sslcrl":{

<b>"cacertfile":<String_value>,

</b><b>"cakeyfile":<String_value>,

</b><b>"indexfile":<String_value>,

</b>"revoke":<String_value>,

"gencrl":<String_value>,

"password":<String_value>

}}
```

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error

delete

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/sslcrl/crlname_value<String>

HTTP Method:DELETE

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue>

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error

update

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/sslcrl

HTTP Method:PUT

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Content-Type:application/json

Request Payload:

```
{"sslcrl":{

<b>"crlname":<String_value>,

</b>"refresh":<String_value>,

"cacert":<String_value>,

"server":<String_value>,

"method":<String_value>,

"url":<String_value>,

"port":<Integer_value>,
```

```
"basedn":<String_value>,

"scope":<String_value>,

"interval":<String_value>,

"day":<Integer_value>,

"time":<String_value>,

"binddn":<String_value>,

"password":<String_value>,

"binary":<String_value>

}}
```

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error

unset

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/sslcrl?**action=unset**

HTTP Method:POST

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Content-Type:application/json

Request Payload:

```
{"sslcrl":{

<b>"crlname":<String_value>,
```

```
</b>"refresh":true,

"cacert":true,

"server":true,

"method":true,

"url":true,

"port":true,

"basedn":true,

"scope":true,

"interval":true,

"day":true,

"time":true,

"binddn":true,

"password":true,

"binary":true

}}
```

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error

get (all)

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/sslcrl

Query-parameters:

attrs

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/sslcrl?**attrs=property-name1,property-name2**

Use this query parameter to specify the resource details that you want to retrieve.

filter

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/sslcrl?**filter=property-name1:property-val1,property-name2:property-val2**

Use this query-parameter to get the filtered set of sslcrl resources configured on Citrix ADC. Filtering can be done on any of the properties of the resource.

view

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/sslcrl?**view=summary**

Note: By default, the retrieved results are displayed in detail view (?view=detail).

pagination

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/sslcrl?**pagesize=#no;pageno=#no**

Use this query-parameter to get the sslcrl resources in chunks.

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "sslcrl": [ {
"crlname":<String_value>,
"crlpath":<String_value>,
"inform":<String_value>,
"cacert":<String_value>,
"refresh":<String_value>,
"scope":<String_value>,
"server":<String_value>,
"port":<Integer_value>,
"url":<String_value>,
"method":<String_value>,
"basedn":<String_value>,
"interval":<String_value>,
"day":<Integer_value>,
"time":<String_value>,
"binddn":<String_value>,
"password":<String_value>,
"flags":<Integer_value>,
"lastupdatetime":<Integer_value>,
"version":<Integer_value>,
"signaturealgo":<String_value>,
```

```
"issuer":<String_value>,

"lastupdate":<String_value>,

"nextupdate":<String_value>,

"binary":<String_value>,

"daystoexpiration":<Integer_value>

}]}
```

get

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/sslcrl/crlname_value<String>

Query-parameters:

attrs

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/sslcrl/crlname_value<String>?**attrs=property-name1,property-name2**

Use this query parameter to specify the resource details that you want to retrieve.

view

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/sslcrl/crlname_value<String>?**view=summary**

Note: By default, the retrieved results are displayed in detail view (?view=detail).

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "sslcrl": [ {

"crlname":<String_value>,

"crlpath":<String_value>,

"inform":<String_value>,

"cacert":<String_value>,

"refresh":<String_value>,

"scope":<String_value>,

"server":<String_value>,

"port":<Integer_value>,

"url":<String_value>,

"method":<String_value>,

"basedn":<String_value>,

"interval":<String_value>,

"day":<Integer_value>,

"time":<String_value>,

"binddn":<String_value>,

"password":<String_value>,

"flags":<Integer_value>,

"lastupdatetime":<Integer_value>,
```

```
"version":<Integer_value>,

"signaturealgo":<String_value>,

"issuer":<String_value>,

"lastupdate":<String_value>,

"nextupdate":<String_value>,

"binary":<String_value>,

"daystoexpiration":<Integer_value>

}]}
```

count

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/sslcrl?**count=yes**

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "sslcrl": [ { "__count": "#no"} ] }
```

sslcrlfile

Configuration for Imported crl files resource.

Properties

(click to see Operations)

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
name	<String>	Read-write	Name to assign to the imported CRL file. Must begin with an ASCII alphanumeric or underscore (_) character, and must contain only ASCII alphanumeric, underscore, hash (#), period (.), space, colon (:), at (@), equals (=), and hyphen (-) characters. The following requirement applies only to the Citrix ADC CLI: If the name includes one or more spaces, enclose the name in double or single quotation marks (for example, "my file" or 'my file'). Minimum length = 1 Maximum length = 31
src	<String>	Read-write	URL specifying the protocol, host, and path, including file

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
			name to the CRL file to be imported. For example, http://www.example.com/crl_file. NOTE: The import fails if the object to be imported is on an HTTPS server that requires client certificate authentication for access. Minimum length = 1 Maximum length = 2047
__count	<Double>	Read-only	count parameter

Operations

(click to see Properties)

IMPORT| DELETE| GET (ALL)| COUNT

Some options that you can use for each operations:

- **Getting warnings in response:**NITRO allows you to get warnings in an operation by specifying the "warning" query parameter as "yes". For example, to get warnings while connecting to the Citrix ADC appliance, the URL is as follows:

  http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/login?warning=yes

  If any, the warnings are displayed in the response payload with the HTTP code "209 X-NITRO-WARNING".
- **Authenticated access for individual NITRO operations:**NITRO allows you to logon to the Citrix ADC appliance to perform individual operations. You can use this option instead of

creating a NITRO session (using the login object) and then using that session to perform all operations,

To do this, you must specify the username and password in the request header of the NITRO request as follows:

X-NITRO-USER:<username>

X-NITRO-PASS:<password>

Note: In such cases, make sure that the request header DOES not include the following:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue>

`*Note:`*

Mandatory parameters are marked in redand placeholder content is marked in <green>.

Import

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/sslcrlfile?**action=Import**

HTTP Method:POST

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Content-Type:application/json

Request Payload:

```
{"sslcrlfile":{

<b>"name":<String_value>,

</b><b>"src":<String_value>

</b>}}
```

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error

delete

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/sslcrlfile

HTTP Method:DELETE

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue>

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error

get (all)

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/sslcrlfile

Query-parameters:

attrs

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/sslcrlfile?**attrs=property-name1,property-name2**

Use this query parameter to specify the resource details that you want to retrieve.

filter

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/sslcrlfile?**filter=property-name1:property-val1,property-name2:property-val2**

Use this query-parameter to get the filtered set of sslcrlfile resources configured on Citrix ADC. Filtering can be done on any of the properties of the resource.

view

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/sslcrlfile?**view=summary**

Note: By default, the retrieved results are displayed in detail view (?view=detail).

pagination

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/sslcrlfile?**pagesize=#no;pageno=#no**

Use this query-parameter to get the sslcrlfile resources in chunks.

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "sslcrlfile": [ {

"name":<String_value>,

"src":<String_value>

}]}
```

count

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/sslcrlfile?**count=yes**

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "sslcrlfile": [ { "__count": "#no"} ] }
```

sslcrl_binding

Binding object which returns the resources bound to sslcrl.

Properties

(click to see Operations)

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
crlname	<String>	Read-write	Name of the CRL for which to show detailed information. Minimum length = 1
sslcrl_serialnumber_binding	<sslcrl_serialnumber_binding[]>	Read-only	serialnumber that can be bound to sslcrl.

Operations

(click to see Properties)

GET| GET (ALL)

Some options that you can use for each operations:

- **Getting warnings in response:**NITRO allows you to get warnings in an operation by specifying the "warning" query parameter as "yes". For example, to get warnings while connecting to the Citrix ADC appliance, the URL is as follows:

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/login?warning=yes

If any, the warnings are displayed in the response payload with the HTTP code "209 X-NITRO-WARNING".

- **Authenticated access for individual NITRO operations:**NITRO allows you to logon to the Citrix ADC appliance to perform individual operations. You can use this option instead of creating a NITRO session (using the login object) and then using that session to perform all operations,

  To do this, you must specify the username and password in the request header of the NITRO request as follows:

  X-NITRO-USER:<username>

  X-NITRO-PASS:<password>

  Note: In such cases, make sure that the request header DOES not include the following:

  Cookie:NITRO_AUTH_TOKEN=<tokenvalue>

`*Note:`*

Mandatory parameters are marked in redand placeholder content is marked in <green>.

get

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/sslcrl_binding/crlname_value<String>

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "sslcrl_binding": [ {

"crlname":<String_value>,

"sslcrl_serialnumber_binding":<sslcrl_serialnumber_binding[]_value>

}]}
```

get (all)

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/sslcrl_binding

Query-parameters:

bulkbindings

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/sslcrl_binding?**bulkbindings=yes**

NITRO allows you to fetch bindings in bulk.

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "sslcrl_binding": [ {

"crlname":<String_value>,
```

```
"sslcrl_serialnumber_binding":<sslcrl_serialnumber_binding[]_value>

}]}
```

sslcrl_serialnumber_binding

Binding object showing the serialnumber that can be bound to sslcrl.

Properties

(click to see Operations)

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
number	<String>	Read-write	Certificate Serial number.
crlname	<String>	Read-write	Name of the CRL for which to show detailed information. Minimum length = 1
date	<String>	Read-only	Certificate Revocation date.
__count	<Double>	Read-only	count parameter

Operations

(click to see Properties)

GET| GET (ALL)| COUNT

Some options that you can use for each operations:

- **Getting warnings in response:**NITRO allows you to get warnings in an operation by specifying the "warning" query parameter as "yes". For example, to get warnings while connecting to the Citrix ADC appliance, the URL is as follows:

  http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/login?warning=yes

If any, the warnings are displayed in the response payload with the HTTP code "209 X-NITRO-WARNING".

- **Authenticated access for individual NITRO operations:**NITRO allows you to logon to the Citrix ADC appliance to perform individual operations. You can use this option instead of creating a NITRO session (using the login object) and then using that session to perform all operations,

  To do this, you must specify the username and password in the request header of the NITRO request as follows:

  X-NITRO-USER:<username>

  X-NITRO-PASS:<password>

  Note: In such cases, make sure that the request header DOES not include the following:

  Cookie:NITRO_AUTH_TOKEN=<tokenvalue>

`*Note:`*

Mandatory parameters are marked in redand placeholder content is marked in <green>.

get

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/sslcrl_serialnumber_binding/crlname_value<String>

Query-parameters:

filter

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/sslcrl_serialnumber_binding/crlname_value<String>?**filter=property-name1:property-value1,property-name2:property-value2**

Use this query-parameter to get the filtered set of sslcrl_serialnumber_binding resources configured on Citrix ADC. Filtering can be done on any of the properties of the resource.

pagination

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/sslcrl_serialnumber_binding/crlname_value<String>?**pagesize=#no;pageno=#no**

Use this query-parameter to get the sslcrl_serialnumber_binding resources in chunks.

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "sslcrl_serialnumber_binding": [ {

"number":<String_value>,

"crlname":<String_value>,

"date":<String_value>

}]}
```

get (all)

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/sslcrl_serialnumber_binding

Query-parameters:

bulkbindings

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/sslcrl_serialnumber_binding?**bulkbindings=yes**

NITRO allows you to fetch bindings in bulk.

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "sslcrl_serialnumber_binding": [ {

"number":<String_value>,

"crlname":<String_value>,

"date":<String_value>

}]}
```

count

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/sslcrl_serialnumber_binding/crlname_value<String>?**count=yes**

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{"sslcrl_serialnumber_binding": [ { "__count": "#no"} ] }
```

ssldhfile

Configuration for dh imported file resource.

Properties

(click to see Operations)

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
name	<String>	Read-write	Name to assign to the imported DH file. Must begin with an ASCII alphanumeric or underscore (_) character, and must contain only ASCII alphanumeric, underscore, hash (#), period (.), space, colon (:), at (@), equals (=), and hyphen (-) characters. The following requirement applies only to the Citrix ADC CLI: If the name includes one or more spaces, enclose the name in double or single quotation marks (for example, "my file" or 'my file').

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
			Minimum length = 1 Maximum length = 31
src	<String>	Read-write	URL specifying the protocol, host, and path, including file name, to the DH file to be imported. For example, http://www.example.com/dh_file. NOTE: The import fails if the file is on an HTTPS server that requires client certificate authentication for access. Minimum length = 1 Maximum length = 2047
__count	<Double>	Read-only	count parameter

Operations

(click to see Properties)

IMPORT| DELETE| GET (ALL)| COUNT

Some options that you can use for each operations:

- **Getting warnings in response:**NITRO allows you to get warnings in an operation by specifying the "warning" query parameter as "yes". For example, to get warnings while connecting to the Citrix ADC appliance, the URL is as follows:

  http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/login?warning=yes

If any, the warnings are displayed in the response payload with the HTTP code "209 X-NITRO-WARNING".

- **Authenticated access for individual NITRO operations:**NITRO allows you to logon to the Citrix ADC appliance to perform individual operations. You can use this option instead of creating a NITRO session (using the login object) and then using that session to perform all operations,

  To do this, you must specify the username and password in the request header of the NITRO request as follows:

  X-NITRO-USER:<username>

  X-NITRO-PASS:<password>

  Note: In such cases, make sure that the request header DOES not include the following:

  Cookie:NITRO_AUTH_TOKEN=<tokenvalue>

`*Note:`*

Mandatory parameters are marked in redand placeholder content is marked in <green>.

Import

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/ssldhfile?**action=Import**

HTTP Method:POST

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Content-Type:application/json

Request Payload:

```
{"ssldhfile":{

<b>"name":<String_value>,

</b><b>"src":<String_value>

</b>}}
```

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error

delete

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/ssldhfile

HTTP Method:DELETE

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue>

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error

get (all)

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/ssldhfile

Query-parameters:

attrs

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/ssldhfile?**attrs=property-name1,property-name2**

Use this query parameter to specify the resource details that you want to retrieve.

filter

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/ssldhfile?**filter=property-name1:property-val1,property-name2:property-val2**

Use this query-parameter to get the filtered set of ssldhfile resources configured on Citrix ADC. Filtering can be done on any of the properties of the resource.

view

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/ssldhfile?**view=summary**

Note: By default, the retrieved results are displayed in detail view (?view=detail).

pagination

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/ssldhfile?**pagesize=#no;pageno=#no**

Use this query-parameter to get the ssldhfile resources in chunks.

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "ssldhfile": [ {

"name":<String_value>,

"src":<String_value>

}]}
```

count

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/ssldhfile?**count=yes**

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "ssldhfile": [ { "__count": "#no"} ] }
```

ssldhparam

Configuration for dh Parameter resource.

Properties

(click to see Operations)

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
dhfile	<String>	Read-write	Name of and, optionally, path to the DH key file. /nsconfig/ssl/ is the default path. Maximum length = 63
bits	<Double>	Read-write	Size, in bits, of the DH key being generated. Minimum value = 512

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
			Maximum value = 2048
gen	<String>	Read-write	Random number required for generating the DH key. Required as part of the DH key generation algorithm. Default value: 2 Possible values = 2, 5

Operations

(click to see Properties)

CREATE

Some options that you can use for each operations:

- **Getting warnings in response:**NITRO allows you to get warnings in an operation by specifying the "warning" query parameter as "yes". For example, to get warnings while connecting to the Citrix ADC appliance, the URL is as follows:

  http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/login?warning=yes

  If any, the warnings are displayed in the response payload with the HTTP code "209 X-NITRO-WARNING".
- **Authenticated access for individual NITRO operations:**NITRO allows you to logon to the Citrix ADC appliance to perform individual operations. You can use this option instead of creating a NITRO session (using the login object) and then using that session to perform all operations,

  To do this, you must specify the username and password in the request header of the NITRO request as follows:

  X-NITRO-USER:<username>

  X-NITRO-PASS:<password>

Note: In such cases, make sure that the request header DOES not include the following:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue>

`*Note:` *

Mandatory parameters are marked in redand placeholder content is marked in <green>.

create

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/ssldhparam?**action=create**

HTTP Method:POST

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Content-Type:application/json

Request Payload:

```
{"ssldhparam":{

<b>"dhfile":<String_value>,

</b>"bits":<Double_value>,

"gen":<String_value>

}}
```

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error

ssldsakey

Configuration for dsa key resource.

Properties

(click to see Operations)

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
keyfile	<String>	Read-write	Name for and, optionally, path to the DSA key file. /nsconfig/ssl/ is the default path. Maximum length = 63
bits	<Double>	Read-write	Size, in bits, of the DSA key. Minimum value = 512 Maximum value = 2048
keyform	<String>	Read-write	Format in which the DSA key file is stored on the appliance. Default value: PEM Possible values = DER, PEM
des	<Boolean>	Read-write	Encrypt the generated DSA key by using the DES algorithm. On the command line, you are prompted to enter the pass phrase (password) that will be used to encrypt the key.
des3	<Boolean>	Read-write	Encrypt the generated DSA key

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
			by using the Triple-DES algorithm. On the command line, you are prompted to enter the pass phrase (password) that will be used to encrypt the key.
aes256	<Boolean>	Read-write	Encrypt the generated DSA key by using the AES algorithm.
password	<String>	Read-write	Pass phrase to use for encryption if DES or DES3 option is selected. Minimum length = 1 Maximum length = 31
pkcs8	<Boolean>	Read-write	Create the private key in PKCS#8 format.

Operations

(click to see Properties)

CREATE

Some options that you can use for each operations:

- **Getting warnings in response:**NITRO allows you to get warnings in an operation by specifying the "warning" query parameter as "yes". For example, to get warnings while connecting to the Citrix ADC appliance, the URL is as follows:

  http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/login?warning=yes

If any, the warnings are displayed in the response payload with the HTTP code "209 X-NITRO-WARNING".

- **Authenticated access for individual NITRO operations:**NITRO allows you to logon to the Citrix ADC appliance to perform individual operations. You can use this option instead of creating a NITRO session (using the login object) and then using that session to perform all operations,

  To do this, you must specify the username and password in the request header of the NITRO request as follows:

  X-NITRO-USER:<username>

  X-NITRO-PASS:<password>

  Note: In such cases, make sure that the request header DOES not include the following:

  Cookie:NITRO_AUTH_TOKEN=<tokenvalue>

`*Note:`*

Mandatory parameters are marked in redand placeholder content is marked in <green>.

create

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/ssldsakey?**action=create**

HTTP Method:POST

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Content-Type:application/json

Request Payload:

```
{"ssldsakey":{

<b>"keyfile":<String_value>,

</b><b>"bits":<Double_value>,

</b>"keyform":<String_value>,
```

```
"des":<Boolean_value>,

"des3":<Boolean_value>,

"aes256":<Boolean_value>,

"password":<String_value>,

"pkcs8":<Boolean_value>

}}
```

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error

ssldtlsprofile

Configuration for DTLS profile resource.

Properties

(click to see Operations)

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
name	<String>	Read-write	Name for the DTLS profile. Must begin with an ASCII alphanumeric or underscore (_) character, and must contain only ASCII alphanumeric, underscore, hash (#), period (.), space, colon (:), at (@),equals sign (=), and hyphen (-)

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
			characters. Cannot be changed after the profile is created. Minimum length = 1 Maximum length = 127
pmtudiscovery	<String>	Read-write	Source for the maximum record size value. If ENABLED, the value is taken from the PMTU table. If DISABLED, the value is taken from the profile. Default value: DISABLED Possible values = ENABLED, DISABLED
maxrecordsize	<Double>	Read-write	Maximum size of records that can be sent if PMTU is disabled. Default value: 1459 Minimum value = 250 Maximum value = 1459
maxretrytime	<Double>	Read-write	Wait for the specified time, in seconds, before resending the request.

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
			Default value: 3
helloverifyrequest	<String>	Read-write	Send a Hello Verify request to validate the client. Default value: DISABLED Possible values = ENABLED, DISABLED
terminatesession	<String>	Read-write	Terminate the session if the message authentication code (MAC) of the client and server do not match. Default value: DISABLED Possible values = ENABLED, DISABLED
maxpacketsize	<Double>	Read-write	Maximum number of packets to reassemble. This value helps protect against a fragmented packet attack. Default value: 120 Minimum value = 0 Maximum value = 86400
__count	<Double>	Read-only	count parameter

Operations

(click to see Properties)

ADD| DELETE| UPDATE| UNSET| GET (ALL)| GET| COUNT

Some options that you can use for each operations:

- **Getting warnings in response:**NITRO allows you to get warnings in an operation by specifying the "warning" query parameter as "yes". For example, to get warnings while connecting to the Citrix ADC appliance, the URL is as follows:

  http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/login?warning=yes

  If any, the warnings are displayed in the response payload with the HTTP code "209 X-NITRO-WARNING".

- **Authenticated access for individual NITRO operations:**NITRO allows you to logon to the Citrix ADC appliance to perform individual operations. You can use this option instead of creating a NITRO session (using the login object) and then using that session to perform all operations,

  To do this, you must specify the username and password in the request header of the NITRO request as follows:

  X-NITRO-USER:<username>

  X-NITRO-PASS:<password>

  Note: In such cases, make sure that the request header DOES not include the following:

  Cookie:NITRO_AUTH_TOKEN=<tokenvalue>

`*Note:`*

Mandatory parameters are marked in redand placeholder content is marked in <green>.

add

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/ssldtlsprofile

HTTP Method:POST

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Content-Type:application/json

Request Payload:

```
{"ssldtlsprofile":{

<b>"name":<String_value>,

</b>"pmtudiscovery":<String_value>,

"maxrecordsize":<Double_value>,

"maxretrytime":<Double_value>,

"helloverifyrequest":<String_value>,

"terminatesession":<String_value>,

"maxpacketsize":<Double_value>

}}
```

Response:

HTTP Status Code on Success: 201 Created HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error

delete

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/ssldtlsprofile/name_value<String>

HTTP Method:DELETE

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue>

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error

update

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/ssldtlsprofile

HTTP Method:PUT

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Content-Type:application/json

Request Payload:

```
{"ssldtlsprofile":{

<b>"name":<String_value>,

</b>"pmtudiscovery":<String_value>,

"maxrecordsize":<Double_value>,

"maxretrytime":<Double_value>,

"helloverifyrequest":<String_value>,

"terminatesession":<String_value>,

"maxpacketsize":<Double_value>

}}
```

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error

unset

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/ssldtlsprofile?**action=unset**

HTTP Method:POST

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Content-Type:application/json

Request Payload:

```
{"ssldtlsprofile":{

<b>"name":<String_value>,

</b>"pmtudiscovery":true,

"maxrecordsize":true,

"maxretrytime":true,

"helloverifyrequest":true,

"terminatesession":true,

"maxpacketsize":true

}}
```

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error

get (all)

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/ssldtlsprofile

Query-parameters:

attrs

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/ssldtlsprofile?**attrs=property-name1,property-name2**

Use this query parameter to specify the resource details that you want to retrieve.

filter

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/ssldtlsprofile?**filter=property-name1:property-val1,property-name2:property-val2**

Use this query-parameter to get the filtered set of ssldtlsprofile resources configured on Citrix ADC. Filtering can be done on any of the properties of the resource.

view

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/ssldtlsprofile?**view=summary**

Note: By default, the retrieved results are displayed in detail view (?view=detail).

pagination

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/ssldtlsprofile?**pagesize=#no;pageno=#no**

Use this query-parameter to get the ssldtlsprofile resources in chunks.

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "ssldtlsprofile": [ {

"name":<String_value>,

"pmtudiscovery":<String_value>,
```

```
"maxrecordsize":<Double_value>,

"maxretrytime":<Double_value>,

"helloverifyrequest":<String_value>,

"terminatesession":<String_value>,

"maxpacketsize":<Double_value>

}]}
```

get

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/ssldtlsprofile/name_value<String>

Query-parameters:

attrs

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/ssldtlsprofile/name_value<String>?**attrs=property-name1,property-name2**

Use this query parameter to specify the resource details that you want to retrieve.

view

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/ssldtlsprofile/name_value<String>?**view=summary**

Note: By default, the retrieved results are displayed in detail view (?view=detail).

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "ssldtlsprofile": [ {

"name":<String_value>,

"pmtudiscovery":<String_value>,

"maxrecordsize":<Double_value>,

"maxretrytime":<Double_value>,

"helloverifyrequest":<String_value>,

"terminatesession":<String_value>,

"maxpacketsize":<Double_value>

}]}
```

count

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/ssldtlsprofile?**count=yes**

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "ssldtlsprofile": [ { "__count": "#no"} ] }
```

sslfips

Configuration for fips resource.

Properties

(click to see Operations)

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
inithsm	<String>	Read-write	FIPS initialization level. The appliance currently supports Level-2 (FIPS 140-2). Possible values = Level-2
sopassword	<String>	Read-write	Security officer password that will be in effect after you have configured the HSM. Minimum length = 1
oldsopassword	<String>	Read-write	Old password for the security officer. Minimum length = 1
userpassword	<String>	Read-write	The Hardware Security Module's (HSM) User password. Minimum length = 1

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
hsmlabel	<String>	Read-write	Label to identify the Hardware Security Module (HSM). Minimum length = 1
fipsfw	<String>	Read-write	Path to the FIPS firmware file. Minimum length = 1
erasedata	<String>	Read-only	Erase data. Default value: FIPS_ERASE Minimum length = 1
serial	<Integer>	Read-only	FIPS card serial number.
majorversion	<Integer>	Read-only	Firmware major version.
minorversion	<Integer>	Read-only	Firmware minor version.
fipshwmajorversion	<Integer>	Read-only	FIPS card hardware major version.
fipshwminorversion	<Integer>	Read-only	FIPS card hardware minor version.
fipshwversionstring	<String>	Read-only	FIPS card hardware extended version string.
flashmemorytotal	<Integer>	Read-only	Total size of the flash memory on card.
flashmemoryfree	<Integer>	Read-only	Total size of free flash memory.

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
sramtotal	<Integer>	Read-only	Total size of the SRAM memory on card.
sramfree	<Integer>	Read-only	Total size of free SRAM memory.
status	<Integer>	Read-only	Status.
flag	<Integer>	Read-only	Internal Flags.
serialno	<String>	Read-only	FIPS card serial number.
model	<String>	Read-only	FIPS card model info.
state	<Integer>	Read-only	FIPS card state.
firmwarereleasedate	<String>	Read-only	FIPS card firmware revision date.
coresmax	<Integer>	Read-only	Maximum number of crypto cores present in the FIPS card.
coresenabled	<Integer>	Read-only	Number of crypto cores enabled in the FIPS card.

Operations

(click to see Properties)

UPDATE| UNSET| RESET| CHANGE| GET (ALL)

Some options that you can use for each operations:

- **Getting warnings in response:**NITRO allows you to get warnings in an operation by specifying the "warning" query parameter as "yes". For example, to get warnings while connecting to the Citrix ADC appliance, the URL is as follows:

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/login?warning=yes

If any, the warnings are displayed in the response payload with the HTTP code "209 X-NITRO-WARNING".

- **Authenticated access for individual NITRO operations:**NITRO allows you to logon to the Citrix ADC appliance to perform individual operations. You can use this option instead of creating a NITRO session (using the login object) and then using that session to perform all operations,

  To do this, you must specify the username and password in the request header of the NITRO request as follows:

  X-NITRO-USER:<username>

  X-NITRO-PASS:<password>

  Note: In such cases, make sure that the request header DOES not include the following:

  Cookie:NITRO_AUTH_TOKEN=<tokenvalue>

`*Note:`*

Mandatory parameters are marked in redand placeholder content is marked in <green>.

update

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/sslfips

HTTP Method:PUT

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Content-Type:application/json

Request Payload:

```
{"sslfips":{

<b>"inithsm":<String_value>,

</b><b>"sopassword":<String_value>,

</b><b>"oldsopassword":<String_value>,
```

```
</b><b>"userpassword":<String_value>,

</b>"hsmlabel":<String_value>

}}
```

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error

unset

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/sslfips?**action=unset**

HTTP Method:POST

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Content-Type:application/json

Request Payload:

```
{"sslfips":{

"hsmlabel":true

}}
```

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error

reset

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/sslfips?**action=reset**

HTTP Method:POST

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Content-Type:application/json

Request Payload:

```
{"sslfips":{

}}
```

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error

change

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/sslfips?**action=update**

HTTP Method:POST

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Content-Type:application/json

Request Payload:

```
{"sslfips":{

<b>"fipsfw":<String_value>

</b>}}
```

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error

get (all)

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/sslfips

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "sslfips": [ {

"inithsm":<String_value>,

"sopassword":<String_value>,

"userpassword":<String_value>,

"oldsopassword":<String_value>,

"erasedata":<String_value>,

"hsmlabel":<String_value>,

"serial":<Integer_value>,

"majorversion":<Integer_value>,
```

```
"minorversion":<Integer_value>,

"fipshwmajorversion":<Integer_value>,

"fipshwminorversion":<Integer_value>,

"fipshwversionstring":<String_value>,

"flashmemorytotal":<Integer_value>,

"flashmemoryfree":<Integer_value>,

"sramtotal":<Integer_value>,

"sramfree":<Integer_value>,

"status":<Integer_value>,

"flag":<Integer_value>,

"serialno":<String_value>,

"model":<String_value>,

"state":<Integer_value>,

"firmwarereleasedate":<String_value>,

"coresmax":<Integer_value>,

"coresenabled":<Integer_value>

}]}
```

sslfipskey

Configuration for FIPS key resource.

Properties

(click to see Operations)

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
fipskeyname	<String>	Read-write	Name for the FIPS key. Must begin with an ASCII alphanumeric or underscore (_) character, and must contain only ASCII alphanumeric, underscore, hash (#), period (.), space, colon (:), at (@), equals (=), and hyphen (-) characters. Cannot be changed after the FIPS key is created. The following requirement applies only to the Citrix ADC CLI: If the name includes one or more spaces, enclose the name in double or single quotation marks (for example, "my fipskey" or 'my fipskey'). Minimum length = 1
keytype	<String>	Read-write	Only RSA key and ECDSA Key are supported. Default value: RSA Possible values = RSA, ECDSA

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
exponent	<String>	Read-write	Exponent value for the FIPS key to be created. Available values function as follows: 3=3 (hexadecimal) F4=10001 (hexadecimal). Default value: 3 Possible values = 3, F4
modulus	<Double>	Read-write	Modulus, in multiples of 64, of the FIPS key to be created. Minimum value = 0 Maximum value = 4096
curve	<String>	Read-write	Only p_256 (prime256v1) and P_384 (secp384r1) are supported. Default value: P_256 Possible values = P_256, P_384
key	<String>	Read-write	Name of and, optionally, path to the key file to be imported. /nsconfig/ssl/ is the default path.

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
			Minimum length = 1
inform	<String>	Read-write	Input format of the key file. Available formats are: SIM - Secure Information Management; select when importing a FIPS key. If the external FIPS key is encrypted, first decrypt it, and then import it. PEM - Privacy Enhanced Mail; select when importing a non-FIPS key. Default value: SIM Possible values = SIM, DER, PEM
wrapkeyname	<String>	Read-write	Name of the wrap key to use for importing the key. Required for importing a non-FIPS key. Minimum length = 1
iv	<String>	Read-write	Initialization Vector (IV) to use for importing the key. Required for

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
			importing a non-FIPS key. Minimum length = 1
size	<Double>	Read-only	Size.
__count	<Double>	Read-only	count parameter

Operations

(click to see Properties)

CREATE| DELETE| IMPORT| EXPORT| GET (ALL)| GET| COUNT

Some options that you can use for each operations:

- **Getting warnings in response:**NITRO allows you to get warnings in an operation by specifying the "warning" query parameter as "yes". For example, to get warnings while connecting to the Citrix ADC appliance, the URL is as follows:

  http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/login?warning=yes

  If any, the warnings are displayed in the response payload with the HTTP code "209 X-NITRO-WARNING".

- **Authenticated access for individual NITRO operations:**NITRO allows you to logon to the Citrix ADC appliance to perform individual operations. You can use this option instead of creating a NITRO session (using the login object) and then using that session to perform all operations,

  To do this, you must specify the username and password in the request header of the NITRO request as follows:

  X-NITRO-USER:<username>

  X-NITRO-PASS:<password>

  Note: In such cases, make sure that the request header DOES not include the following:

  Cookie:NITRO_AUTH_TOKEN=<tokenvalue>

`*Note:`*

Mandatory parameters are marked in redand placeholder content is marked in <green>.

create

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/sslfipskey?**action=create**

HTTP Method:POST

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Content-Type:application/json

Request Payload:

```
{"sslfipskey":{

<b>"fipskeyname":<String_value>,

</b><b>"keytype":<String_value>,

</b>"exponent":<String_value>,

"modulus":<Double_value>,

"curve":<String_value>

}}
```

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error

delete

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/sslfipskey/fipskeyname_value<String>

HTTP Method:DELETE

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue>

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error

Import

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/sslfipskey?**action=Import**

HTTP Method:POST

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Content-Type:application/json

Request Payload:

```
{"sslfipskey":{

<b>"fipskeyname":<String_value>,

</b><b>"key":<String_value>,

</b>"inform":<String_value>,

"wrapkeyname":<String_value>,

"iv":<String_value>,

"exponent":<String_value>

}}
```

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error

export

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/sslfipskey?**action=export**

HTTP Method:POST

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Content-Type:application/json

Request Payload:

```
{"sslfipskey":{

<b>"fipskeyname":<String_value>,

</b><b>"key":<String_value>

</b>}}
```

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error

get (all)

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/sslfipskey

Query-parameters:

attrs

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/sslfipskey?**attrs=property-name1,property-name2**

Use this query parameter to specify the resource details that you want to retrieve.

filter

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/sslfipskey?**filter=property-name1:property-val1,property-name2:property-val2**

Use this query-parameter to get the filtered set of sslfipskey resources configured on Citrix ADC. Filtering can be done on any of the properties of the resource.

view

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/sslfipskey?**view=summary**

Note: By default, the retrieved results are displayed in detail view (?view=detail).

pagination

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/sslfipskey?**pagesize=#no;pageno=#no**

Use this query-parameter to get the sslfipskey resources in chunks.

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "sslfipskey": [ {

"fipskeyname":<String_value>,

"modulus":<Double_value>,

"exponent":<String_value>,

"curve":<String_value>,
```

```
"keytype":<String_value>,

"size":<Double_value>

}]}
```

get

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/sslfipskey/fipskeyname_value<String>

Query-parameters:

attrs

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/sslfipskey/fipskeyname_value<String>?**attrs=property-name1,property-name2**

Use this query parameter to specify the resource details that you want to retrieve.

view

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/sslfipskey/fipskeyname_value<String>?**view=summary**

Note: By default, the retrieved results are displayed in detail view (?view=detail).

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "sslfipskey": [ {

"fipskeyname":<String_value>,

"modulus":<Double_value>,

"exponent":<String_value>,

"curve":<String_value>,

"keytype":<String_value>,

"size":<Double_value>

}]}
```

count

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/sslfipskey?**count=yes**

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "sslfipskey": [ { "__count": "#no"} ] }
```

sslfipssimsource

Configuration for FIPsSIM source resource.

Properties

(click to see Operations)

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
targetsecret	<String>	Read-write	Name of and, optionally, path to the target FIPS appliance's secret data. /nsconfig/ssl/ is the default path. Minimum length = 1
sourcesecret	<String>	Read-write	Name for and, optionally, path to the source FIPS appliance's secret data. /nsconfig/ssl/ is the default path. Minimum length = 1
certfile	<String>	Read-write	Name for and, optionally, path to the source FIPS appliance's certificate file. /nsconfig/ssl/ is the default path. Minimum length = 1

Operations

(click to see Properties)

ENABLE| INIT

Some options that you can use for each operations:

- **Getting warnings in response:**NITRO allows you to get warnings in an operation by specifying the "warning" query parameter as "yes". For example, to get warnings while connecting to the Citrix ADC appliance, the URL is as follows:

  http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/login?warning=yes

  If any, the warnings are displayed in the response payload with the HTTP code "209 X-NITRO-WARNING".

- **Authenticated access for individual NITRO operations:**NITRO allows you to logon to the Citrix ADC appliance to perform individual operations. You can use this option instead of creating a NITRO session (using the login object) and then using that session to perform all operations,

  To do this, you must specify the username and password in the request header of the NITRO request as follows:

  X-NITRO-USER:<username>

  X-NITRO-PASS:<password>

  Note: In such cases, make sure that the request header DOES not include the following:

  Cookie:NITRO_AUTH_TOKEN=<tokenvalue>

`*Note:`*

Mandatory parameters are marked in redand placeholder content is marked in <green>.

enable

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/sslfipssimsource?**action=enable**

HTTP Method:POST

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Content-Type:application/json

Request Payload:

```
{"sslfipssimsource":{

<b>"targetsecret":<String_value>,

</b><b>"sourcesecret":<String_value>

</b>}}
```

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error

init

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/sslfipssimsource?**action=init**

HTTP Method:POST

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Content-Type:application/json

Request Payload:

```
{"sslfipssimsource":{

<b>"certfile":<String_value>

</b>}}
```

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error

sslfipssimtarget

Configuration for FIPS SIM Target resource.

Properties

(click to see Operations)

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
keyvector	<String>	Read-write	Name of and, optionally, path to the target FIPS appliance's key vector. /nsconfig/ssl/ is the default path. Minimum length = 1
sourcesecret	<String>	Read-write	Name of and, optionally, path to the source FIPS appliance's secret data. /nsconfig/ssl/ is the default path. Minimum length = 1
certfile	<String>	Read-write	Name of and, optionally, path to the source FIPS appliance's certificate file. /nsconfig/ssl/ is the default path. Minimum length = 1
targetsecret	<String>	Read-write	Name for and, optionally, path to the target FIPS appliance's secret data. The default input path for the secret data is /nsconfig/ssl/.

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
			Minimum length = 1

Operations

(click to see Properties)

ENABLE| INIT

Some options that you can use for each operations:

- **Getting warnings in response:**NITRO allows you to get warnings in an operation by specifying the "warning" query parameter as "yes". For example, to get warnings while connecting to the Citrix ADC appliance, the URL is as follows:

  http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/login?warning=yes

  If any, the warnings are displayed in the response payload with the HTTP code "209 X-NITRO-WARNING".

- **Authenticated access for individual NITRO operations:**NITRO allows you to logon to the Citrix ADC appliance to perform individual operations. You can use this option instead of creating a NITRO session (using the login object) and then using that session to perform all operations,

  To do this, you must specify the username and password in the request header of the NITRO request as follows:

  X-NITRO-USER:<username>

  X-NITRO-PASS:<password>

  Note: In such cases, make sure that the request header DOES not include the following:

  Cookie:NITRO_AUTH_TOKEN=<tokenvalue>

`*Note:`*

Mandatory parameters are marked in redand placeholder content is marked in <green>.

enable

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/sslfipssimtarget?**action=enable**

HTTP Method:POST

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Content-Type:application/json

Request Payload:

```
{"sslfipssimtarget":{

<b>"keyvector":<String_value>,

</b><b>"sourcesecret":<String_value>

</b>}}
```

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error

init

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/sslfipssimtarget?**action=init**

HTTP Method:POST

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Content-Type:application/json

Request Payload:

```
{"sslfipssimtarget":{

<b>"certfile":<String_value>,

</b><b>"keyvector":<String_value>,

</b><b>"targetsecret":<String_value>
```

```
</b>}}
```

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error

sslglobal_binding

Binding object which returns the resources bound to sslglobal.

Properties

(click to see Operations)

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
sslglobal_sslpolicy_binding	<sslglobal_sslpolicy_binding[]>	Read-only	sslpolicy that can be bound to sslglobal.

Operations

(click to see Properties)

GET| GET (ALL)

Some options that you can use for each operations:

- **Getting warnings in response:**NITRO allows you to get warnings in an operation by specifying the "warning" query parameter as "yes". For example, to get warnings while connecting to the Citrix ADC appliance, the URL is as follows:

  http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/login?warning=yes

  If any, the warnings are displayed in the response payload with the HTTP code "209 X-NITRO-WARNING".
- **Authenticated access for individual NITRO operations:**NITRO allows you to logon to the Citrix ADC appliance to perform individual operations. You can use this option instead of

creating a NITRO session (using the login object) and then using that session to perform all operations,

To do this, you must specify the username and password in the request header of the NITRO request as follows:

X-NITRO-USER:<username>

X-NITRO-PASS:<password>

Note: In such cases, make sure that the request header DOES not include the following:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue>

`*Note:`*

Mandatory parameters are marked in redand placeholder content is marked in <green>.

get

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/sslglobal_binding

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "sslglobal_binding": [ {

"sslglobal_sslpolicy_binding":<sslglobal_sslpolicy_binding[]_value>

}]}
```

get (all)

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/sslglobal_binding

Query-parameters:

bulkbindings

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/sslglobal_binding?**bulkbindings=yes**

NITRO allows you to fetch bindings in bulk.

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "sslglobal_binding": [ {

"sslglobal_sslpolicy_binding":<sslglobal_sslpolicy_binding[]_value>

}]}
```

sslglobal_sslpolicy_binding

Binding object showing the sslpolicy that can be bound to sslglobal.

Properties

(click to see Operations)

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
priority	<Double>	Read-write	The priority of the policy binding.
globalbindtype	<String>	Read-write	. Default value: SYSTEM_GLOBAL Possible values = SYSTEM_GLOBAL, VPN_GLOBAL, RNAT_GLOBAL
policyname	<String>	Read-write	The name for the SSL policy.
labelname	<String>	Read-write	Name of the virtual server or user-defined policy label to invoke if the policy evaluates to TRUE.
gotopriorityexpression	<String>	Read-write	Expression or other value specifying the next policy to be evaluated if the current policy evaluates to TRUE. Specify one of the following values: * NEXT - Evaluate the policy with the next higher priority number. * END - End policy evaluation. * USE_INVOCATION_RESULT - Applicable if this policy invokes

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
			another policy label. If the final goto in the invoked policy label has a value of END, the evaluation stops. If the final goto is anything other than END, the current policy label performs a NEXT. * An expression that evaluates to a number. If you specify an expression, the number to which it evaluates determines the next policy to evaluate, as follows: * If the expression evaluates to a higher numbered priority, the policy with that priority is evaluated next. * If the expression evaluates to the priority of the current policy, the policy with the next higher numbered priority is evaluated next. * If the expression evaluates to a number that is larger than the largest numbered priority, policy evaluation ends. An UNDEF event is triggered if: * The

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
			expression is invalid. * The expression evaluates to a priority number that is numerically lower than the current policy's priority. * The expression evaluates to a priority number that is between the current policy's priority number (say, 30) and the highest priority number (say, 100), but does not match any configured priority number (for example, the expression evaluates to the number 85). This example assumes that the priority number increments by 10 for every successive policy, and therefore a priority number of 85 does not exist in the policy label. Default value: "END"
type	<String>	Read-write	Global bind point to which the policy is bound. Possible values = CONTROL_OVERRIDE, CONTROL_DEFAU

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
			LT, DATA_OVERRIDE, DATA_DEFAULT
invoke	<Boolean>	Read-write	Invoke policies bound to a virtual server, service, or policy label. After the invoked policies are evaluated, the flow returns to the policy with the next priority.
labeltype	<String>	Read-write	Type of policy label to invoke. Specify virtual server for a policy label associated with a virtual server, or policy label for a user-defined policy label. Possible values = vserver, service, policylabel
__count	<Double>	Read-write	count parameter

Operations

(click to see Properties)

ADD:| DELETE:| GET| GET (ALL)| COUNT

Some options that you can use for each operations:

- **Getting warnings in response:**NITRO allows you to get warnings in an operation by specifying the "warning" query parameter as "yes". For example, to get warnings while connecting to the Citrix ADC appliance, the URL is as follows:

  http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/login?warning=yes

If any, the warnings are displayed in the response payload with the HTTP code "209 X-NITRO-WARNING".

- **Authenticated access for individual NITRO operations:**NITRO allows you to logon to the Citrix ADC appliance to perform individual operations. You can use this option instead of creating a NITRO session (using the login object) and then using that session to perform all operations,

  To do this, you must specify the username and password in the request header of the NITRO request as follows:

  X-NITRO-USER:<username>

  X-NITRO-PASS:<password>

  Note: In such cases, make sure that the request header DOES not include the following:

  Cookie:NITRO_AUTH_TOKEN=<tokenvalue>

`*Note:`*

Mandatory parameters are marked in redand placeholder content is marked in <green>.

add

URL:http://<Citrix-ADC-IP-address(NSIP)/nitro/v1/config/sslglobal_sslpolicy_binding

HTTP Method:PUT

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Content-Type:application/json

Request Payload:

```
{

"sslglobal_sslpolicy_binding":{

"policyname":<String_value>,

"priority":<Double_value>,
```

```
"gotopriorityexpression":<String_value>,

"type":<String_value>,

"invoke":<Boolean_value>,

"labeltype":<String_value>,

"labelname":<String_value>

}}
```

Response:

HTTP Status Code on Success: 201 Created HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error

delete

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/sslglobal_sslpolicy_binding

Query-parameters:

args

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/sslglobal_sslpolicy_binding?**args=policyname:<String_value>,type:<String_value>,priority:<Double_value>**

HTTP Method:DELETE

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue>

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error

get

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/sslglobal_sslpolicy_binding

Query-parameters:

filter

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/sslglobal_sslpolicy_binding?**filter=property-name1:property-value1,property-name2:property-value2**

Use this query-parameter to get the filtered set of sslglobal_sslpolicy_binding resources configured on Citrix ADC. Filtering can be done on any of the properties of the resource.

pagination

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/sslglobal_sslpolicy_binding?**pagesize=#no;pageno=#no**

Use this query-parameter to get the sslglobal_sslpolicy_binding resources in chunks.

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "sslglobal_sslpolicy_binding": [ {

"priority":<Double_value>,

"globalbindtype":<String_value>,
```

```
"policyname":<String_value>,

"labelname":<String_value>,

"gotopriorityexpression":<String_value>,

"type":<String_value>,

"invoke":<Boolean_value>,

"labeltype":<String_value>

}]}
```

get (all)

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/sslglobal_sslpolicy_binding

Query-parameters:

bulkbindings

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/sslglobal_sslpolicy_binding?**bulkbindings=yes**

NITRO allows you to fetch bindings in bulk.

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "sslglobal_sslpolicy_binding": [ {

"priority":<Double_value>,

"globalbindtype":<String_value>,

"policyname":<String_value>,

"labelname":<String_value>,

"gotopriorityexpression":<String_value>,

"type":<String_value>,

"invoke":<Boolean_value>,

"labeltype":<String_value>

}]}
```

count

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/sslglobal_sslpolicy_binding?**count=yes**

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "sslglobal_sslpolicy_binding": [ { "__count": "#no"} ] }
```

sslhsmkey

Configuration for HSM key resource.

Properties

(click to see Operations)

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
hsmkeyname	<String>	Read-write	. Minimum length = 1
hsmtype	<String>	Read-write	Type of HSM. Default value: THALES Possible values = THALES, SAFENET, KEYVAULT
key	<String>	Read-write	Name of the key. optionally, for Thales, path to the HSM key file; /var/opt/nfast/kmdata/local/ is the default path. Applies when HSMTYPE is THALES or KEYVAULT. Maximum length = 63
serialnum	<String>	Read-write	Serial number of the partition on which the key is present.

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
			Applies only to SafeNet HSM. Maximum length = 16
password	<String>	Read-write	Password for a partition. Applies only to SafeNet HSM. Minimum length = 1
keystore	<String>	Read-write	Name of keystore object representing HSM where key is stored. For example, name of keyvault object or azurekeyvault authentication object. Applies only to KEYVAULT type HSM. Minimum length = 1
state	<String>	Read-only	Current state of key. Possible values = Created, Access Token Unavailable, Unauthorized, Does not exist, Unreachable, Marked down, Key operations successful, Key operations failed, Key operation throttled
__count	<Double>	Read-only	count parameter

Operations

(click to see Properties)

ADD| DELETE| GET (ALL)| GET| COUNT

Some options that you can use for each operations:

- **Getting warnings in response:**NITRO allows you to get warnings in an operation by specifying the "warning" query parameter as "yes". For example, to get warnings while connecting to the Citrix ADC appliance, the URL is as follows:

  http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/login?warning=yes

  If any, the warnings are displayed in the response payload with the HTTP code "209 X-NITRO-WARNING".

- **Authenticated access for individual NITRO operations:**NITRO allows you to logon to the Citrix ADC appliance to perform individual operations. You can use this option instead of creating a NITRO session (using the login object) and then using that session to perform all operations,

  To do this, you must specify the username and password in the request header of the NITRO request as follows:

  X-NITRO-USER:<username>

  X-NITRO-PASS:<password>

  Note: In such cases, make sure that the request header DOES not include the following:

  Cookie:NITRO_AUTH_TOKEN=<tokenvalue>

`*Note:`*

Mandatory parameters are marked in redand placeholder content is marked in <green>.

add

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/sslhsmkey

HTTP Method:POST

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Content-Type:application/json

Request Payload:

```
{"sslhsmkey":{

<b>"hsmkeyname":<String_value>,

</b>"hsmtype":<String_value>,

"key":<String_value>,

"serialnum":<String_value>,

"password":<String_value>,

"keystore":<String_value>

}}
```

Response:

HTTP Status Code on Success: 201 Created HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error

delete

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/sslhsmkey/hsmkeyname_value<String>

HTTP Method:DELETE

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue>

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error

get (all)

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/sslhsmkey

Query-parameters:

attrs

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/sslhsmkey?**attrs=property-name1,property-name2**

Use this query parameter to specify the resource details that you want to retrieve.

filter

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/sslhsmkey?**filter=property-name1:property-val1,property-name2:property-val2**

Use this query-parameter to get the filtered set of sslhsmkey resources configured on Citrix ADC. Filtering can be done on any of the properties of the resource.

view

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/sslhsmkey?**view=summary**

Note: By default, the retrieved results are displayed in detail view (?view=detail).

pagination

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/sslhsmkey?**pagesize=#no;pageno=#no**

Use this query-parameter to get the sslhsmkey resources in chunks.

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "sslhsmkey": [ {

"hsmkeyname":<String_value>,

"hsmtype":<String_value>,

"key":<String_value>,

"serialnum":<String_value>,

"keystore":<String_value>,

"state":<String_value>

}]}
```

get

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/sslhsmkey/hsmkeyname_value<String>

Query-parameters:

attrs

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/sslhsmkey/hsmkeyname_value<String>?**attrs=property-name1,property-name2**

Use this query parameter to specify the resource details that you want to retrieve.

view

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/sslhsmkey/hsmkeyname_value<String>?**view=summary**

Note: By default, the retrieved results are displayed in detail view (?view=detail).

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "sslhsmkey": [ {

"hsmkeyname":<String_value>,

"hsmtype":<String_value>,

"key":<String_value>,

"serialnum":<String_value>,

"keystore":<String_value>,

"state":<String_value>

}]}
```

count

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/sslhsmkey?**count=yes**

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "sslhsmkey": [ { "__count": "#no"} ] }
```

sslkeyfile

Configuration for Imported ssl key files resource.

Properties

(click to see Operations)

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
name	<String>	Read-write	Name to assign to the imported key file. Must begin with an ASCII alphanumeric or underscore(_) character, and must contain only ASCII alphanumeric, underscore, hash (#), period (.), space, colon (:), at (@),equals (=), and hyphen (-) characters. The following requirement applies only to the Citrix ADC CLI: If the name includes one or more spaces,

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
			enclose the name in double or single quotation marks (for example, "my file" or 'my file'). Minimum length = 1 Maximum length = 31
src	<String>	Read-write	URL specifying the protocol, host, and path, including file name, to the key file to be imported. For example, http://www.example.com/key_file. NOTE: The import fails if the object to be imported is on an HTTPS server that requires client certificate authentication for access. Minimum length = 1 Maximum length = 2047
password	<String>	Read-write	.
__count	<Double>	Read-only	count parameter

Operations

(click to see Properties)

IMPORT| DELETE| GET (ALL)| COUNT

Some options that you can use for each operations:

- **Getting warnings in response:**NITRO allows you to get warnings in an operation by specifying the "warning" query parameter as "yes". For example, to get warnings while connecting to the Citrix ADC appliance, the URL is as follows:

  http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/login?warning=yes

  If any, the warnings are displayed in the response payload with the HTTP code "209 X-NITRO-WARNING".

- **Authenticated access for individual NITRO operations:**NITRO allows you to logon to the Citrix ADC appliance to perform individual operations. You can use this option instead of creating a NITRO session (using the login object) and then using that session to perform all operations,

  To do this, you must specify the username and password in the request header of the NITRO request as follows:

  X-NITRO-USER:<username>

  X-NITRO-PASS:<password>

  Note: In such cases, make sure that the request header DOES not include the following:

  Cookie:NITRO_AUTH_TOKEN=<tokenvalue>

`*Note:`*

Mandatory parameters are marked in redand placeholder content is marked in <green>.

Import

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/sslkeyfile?**action=Import**

HTTP Method:POST

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Content-Type:application/json

Request Payload:

```
{"sslkeyfile":{

<b>"name":<String_value>,

</b><b>"src":<String_value>,

</b>"password":<String_value>

}}
```

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error

delete

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/sslkeyfile

HTTP Method:DELETE

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue>

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error

get (all)

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/sslkeyfile

Query-parameters:

attrs

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/sslkeyfile?**attrs=property-name1,property-name2**

Use this query parameter to specify the resource details that you want to retrieve.

filter

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/sslkeyfile?**filter=property-name1:property-val1,property-name2:property-val2**

Use this query-parameter to get the filtered set of sslkeyfile resources configured on Citrix ADC. Filtering can be done on any of the properties of the resource.

view

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/sslkeyfile?**view=summary**

Note: By default, the retrieved results are displayed in detail view (?view=detail).

pagination

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/sslkeyfile?**pagesize=#no;pageno=#no**

Use this query-parameter to get the sslkeyfile resources in chunks.

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "sslkeyfile": [ {

"name":<String_value>,

"src":<String_value>
```

```
}]}
```

count

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/sslkeyfile?**count=yes**

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "sslkeyfile": [ { "__count": "#no"} ] }
```

sslocspresponder

Configuration for OCSP responser resource.

Properties

(click to see Operations)

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
name	<String>	Read-write	Name for the OCSP responder. Cannot begin with a hash (#) or space character and must contain

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
			only ASCII alphanumeric, underscore (_), hash (#), period (.), space, colon (:), at (@), equals (=), and hyphen (-) characters. Cannot be changed after the responder is created. The following requirement applies only to the Citrix ADC CLI: If the name includes one or more spaces, enclose the name in double or single quotation marks (for example, "my responder" or 'my responder'). Minimum length = 1 Maximum length = 127
url	<String>	Read-write	URL of the OCSP responder. Minimum length = 1 Maximum length = 127
cache	<String>	Read-write	Enable caching of responses. Caching of responses received from the

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
			OCSP responder enables faster responses to the clients and reduces the load on the OCSP responder. Default value: DISABLED Possible values = ENABLED, DISABLED
cachetimeout	<Double>	Read-write	Timeout for caching the OCSP response. After the timeout, the Citrix ADC sends a fresh request to the OCSP responder for the certificate status. If a timeout is not specified, the timeout provided in the OCSP response applies. Default value: 1 Minimum value = 1 Maximum value = 43200
batchingdepth	<Double>	Read-write	Number of client certificates to batch together into one OCSP request. Batching avoids overloading the OCSP responder. A value of 1 signifies that each request is

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
			queried independently. For a value greater than 1, specify a timeout (batching delay) to avoid inordinately delaying the processing of a single certificate. Minimum value = 1 Maximum value = 8
batchingdelay	<Double>	Read-write	Maximum time, in milliseconds, to wait to accumulate OCSP requests to batch. Does not apply if the Batching Depth is 1. Minimum value = 1 Maximum value = 10000
resptimeout	<Double>	Read-write	Time, in milliseconds, to wait for an OCSP response. When this time elapses, an error message appears or the transaction is forwarded, depending on the settings on the virtual server. Includes Batching Delay time. Minimum value = 100

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
			Maximum value = 120000
ocspurlresolvetimeout	<Double>	Read-write	Time, in milliseconds, to wait for an OCSP URL Resolution. When this time elapses, an error message appears or the transaction is forwarded, depending on the settings on the virtual server. Minimum value = 100 Maximum value = 2000
respondercert	<String>	Read-write	. Minimum length = 1
trustresponder	<Boolean>	Read-write	A certificate to use to validate OCSP responses. Alternatively, if -trustResponder is specified, no verification will be done on the reponse. If both are omitted, only the response times (producedAt, lastUpdate, nextUpdate) will be verified.

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
producedattimeskew	<Double>	Read-write	Time, in seconds, for which the Citrix ADC waits before considering the response as invalid. The response is considered invalid if the Produced At time stamp in the OCSP response exceeds or precedes the current Citrix ADC clock time by the amount of time specified. Default value: 300 Minimum value = 0 Maximum value = 86400
signingcert	<String>	Read-write	Certificate-key pair that is used to sign OCSP requests. If this parameter is not set, the requests are not signed. Minimum length = 1
usenonce	<String>	Read-write	Enable the OCSP nonce extension, which is designed to prevent replay attacks. Possible values = YES, NO
insertclientcert	<String>	Read-write	Include the complete client certificate in the OCSP request.

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
			Possible values = YES, NO
httpmethod	<String>	Read-write	HTTP method used to send ocsp request. POST is the default httpmethod. If request length is > 255, POST wil be used even if GET is set as httpMethod. Default value: POST Possible values = GET, POST
ocspaiarefcount	<Double>	Read-only	No of CA certs referencing this AIA responder.
ocspipaddrstr	<String>	Read-only	DNS resolved IP address.
port	<Integer>	Read-only	Port number on which OCSP Server listens. Range 1 - 65535 * in CLI is represented as 65535 in NITRO API
__count	<Double>	Read-only	count parameter

Operations

(click to see Properties)

ADD| DELETE| UPDATE| UNSET| GET (ALL)| GET| COUNT

Some options that you can use for each operations:

- **Getting warnings in response:**NITRO allows you to get warnings in an operation by specifying the "warning" query parameter as "yes". For example, to get warnings while connecting to the Citrix ADC appliance, the URL is as follows:

  http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/login?warning=yes

  If any, the warnings are displayed in the response payload with the HTTP code "209 X-NITRO-WARNING".

- **Authenticated access for individual NITRO operations:**NITRO allows you to logon to the Citrix ADC appliance to perform individual operations. You can use this option instead of creating a NITRO session (using the login object) and then using that session to perform all operations,

  To do this, you must specify the username and password in the request header of the NITRO request as follows:

  X-NITRO-USER:<username>

  X-NITRO-PASS:<password>

  Note: In such cases, make sure that the request header DOES not include the following:

  Cookie:NITRO_AUTH_TOKEN=<tokenvalue>

`*Note:`*

Mandatory parameters are marked in redand placeholder content is marked in <green>.

add

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/sslocspresponder

HTTP Method:POST

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Content-Type:application/json

Request Payload:

```
{"sslocspresponder":{

<b>"name":<String_value>,

</b><b>"url":<String_value>,

</b>"cache":<String_value>,

"cachetimeout":<Double_value>,

"batchingdepth":<Double_value>,

"batchingdelay":<Double_value>,

"resptimeout":<Double_value>,

"ocspurlresolvetimeout":<Double_value>,

"respondercert":<String_value>,

"trustresponder":<Boolean_value>,

"producedattimeskew":<Double_value>,

"signingcert":<String_value>,

"usenonce":<String_value>,

"insertclientcert":<String_value>,

"httpmethod":<String_value>

}}
```

Response:

HTTP Status Code on Success: 201 Created HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error

delete

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/sslocspresponder/name_value<String>

HTTP Method:DELETE

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue>

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error

update

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/sslocspresponder

HTTP Method:PUT

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Content-Type:application/json

Request Payload:

```
{"sslocspresponder":{

<b>"name":<String_value>,

</b>"url":<String_value>,

"cache":<String_value>,

"cachetimeout":<Double_value>,

"batchingdepth":<Double_value>,

"batchingdelay":<Double_value>,

"resptimeout":<Double_value>,
```

```
"ocspurlresolvetimeout":<Double_value>,

"respondercert":<String_value>,

"trustresponder":<Boolean_value>,

"producedattimeskew":<Double_value>,

"signingcert":<String_value>,

"usenonce":<String_value>,

"insertclientcert":<String_value>,

"httpmethod":<String_value>

}}
```

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error

unset

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/sslocspresponder?**action=unset**

HTTP Method:POST

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Content-Type:application/json

Request Payload:

```
{"sslocspresponder":{

<b>"name":<String_value>,
```

```
</b>"trustresponder":true,

"insertclientcert":true,

"cache":true,

"cachetimeout":true,

"batchingdepth":true,

"batchingdelay":true,

"resptimeout":true,

"ocspurlresolvetimeout":true,

"respondercert":true,

"producedattimeskew":true,

"signingcert":true,

"usenonce":true,

"httpmethod":true

}}
```

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error

get (all)

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/sslocspresponder

Query-parameters:

attrs

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/sslocspresponder?**attrs=property-name1,property-name2**

Use this query parameter to specify the resource details that you want to retrieve.

filter

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/sslocspresponder?**filter=property-name1:property-val1,property-name2:property-val2**

Use this query-parameter to get the filtered set of sslocspresponder resources configured on Citrix ADC. Filtering can be done on any of the properties of the resource.

view

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/sslocspresponder?**view=summary**

Note: By default, the retrieved results are displayed in detail view (?view=detail).

pagination

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/sslocspresponder?**pagesize=#no;pageno=#no**

Use this query-parameter to get the sslocspresponder resources in chunks.

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "sslocspresponder": [ {
```

```
"name":<String_value>,

"url":<String_value>,

"cache":<String_value>,

"cachetimeout":<Double_value>,

"batchingdepth":<Double_value>,

"batchingdelay":<Double_value>,

"ocspurlresolvetimeout":<Double_value>,

"resptimeout":<Double_value>,

"producedattimeskew":<Double_value>,

"respondercert":<String_value>,

"trustresponder":<Boolean_value>,

"signingcert":<String_value>,

"usenonce":<String_value>,

"insertclientcert":<String_value>,

"ocspaiarefcount":<Double_value>,

"httpmethod":<String_value>,

"ocspipaddrstr":<String_value>,

"port":<Integer_value>

}]}
```

get

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/sslocspresponder/name_value<String>

Query-parameters:

attrs

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/sslocspresponder/name_value<String>?**attrs=property-name1,property-name2**

Use this query parameter to specify the resource details that you want to retrieve.

view

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/sslocspresponder/name_value<String>?**view=summary**

Note: By default, the retrieved results are displayed in detail view (?view=detail).

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "sslocspresponder": [ {

"name":<String_value>,

"url":<String_value>,

"cache":<String_value>,

"cachetimeout":<Double_value>,

"batchingdepth":<Double_value>,
```

```
"batchingdelay":<Double_value>,

"ocspurlresolvetimeout":<Double_value>,

"resptimeout":<Double_value>,

"producedattimeskew":<Double_value>,

"respondercert":<String_value>,

"trustresponder":<Boolean_value>,

"signingcert":<String_value>,

"usenonce":<String_value>,

"insertclientcert":<String_value>,

"ocspaiarefcount":<Double_value>,

"httpmethod":<String_value>,

"ocspipaddrstr":<String_value>,

"port":<Integer_value>

}]}
```

count

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/sslocspresponder?**count=yes**

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "sslocspresponder": [ { "__count": "#no"} ] }
```

sslparameter

Configuration for SSL parameter resource.

Properties

(click to see Operations)

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
quantumsize	<String>	Read-write	Amount of data to collect before the data is pushed to the crypto hardware for encryption. For large downloads, a larger quantum size better utilizes the crypto resources. Default value: 8192 Possible values = 4096, 8192, 16384
crlmemorysizemb	<Double>	Read-write	Maximum memory size to use for certificate revocation lists (CRLs). This parameter reserves memory for a CRL

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
			but sets a limit to the maximum memory that the CRLs loaded on the appliance can consume. Default value: 256 Minimum value = 10 Maximum value = 1024
strictcachecks	<String>	Read-write	Enable strict CA certificate checks on the appliance. Default value: NO Possible values = YES, NO
ssltriggertimeout	<Double>	Read-write	Time, in milliseconds, after which encryption is triggered for transactions that are not tracked on the Citrix ADC because their length is not known. There can be a delay of up to 10ms from the specified timeout value before the packet is pushed into the queue. Default value: 100 Minimum value = 1 Maximum value = 200

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
sendclosenotify	<String>	Read-write	Send an SSL Close-Notify message to the client at the end of a transaction. Default value: YES Possible values = YES, NO
encrypttriggerpktcount	<Double>	Read-write	Maximum number of queued packets after which encryption is triggered. Use this setting for SSL transactions that send small packets from server to Citrix ADC. Default value: 45 Minimum value = 10 Maximum value = 50
denysslreneg	<String>	Read-write	Deny renegotiation in specified circumstances. Available settings function as follows: * NO - Allow SSL renegotiation. * FRONTEND_CLIENT - Deny secure and nonsecure SSL renegotiation

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
			initiated by the client. * FRONTEND_CLIENTSERVER - Deny secure and nonsecure SSL renegotiation initiated by the client or the Citrix ADC during policy-based client authentication. * ALL - Deny all secure and nonsecure SSL renegotiation. * NONSECURE - Deny nonsecure SSL renegotiation. Allows only clients that support RFC 5746. Default value: ALL Possible values = NO, FRONTEND_CLIENT, FRONTEND_CLIENTSERVER, ALL, NONSECURE

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
insertionencoding	<String>	Read-write	Encoding method used to insert the subject or issuer's name in HTTP requests to servers. Default value: Unicode Possible values = Unicode, UTF-8
ocspcachesize	<Double>	Read-write	Size, per packet engine, in megabytes, of the OCSP cache. A maximum of 10% of the packet engine memory can be assigned. Because the maximum allowed packet engine memory is 4GB, the maximum value that can be assigned to the OCSP cache is approximately 410 MB. Default value: 10 Minimum value = 0 Maximum value = 512
pushflag	<Double>	Read-write	Insert PUSH flag into decrypted, encrypted, or all records. If the PUSH flag is set to a value other than 0, the

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
			buffered records are forwarded on the basis of the value of the PUSH flag. Available settings function as follows: 0 - Auto (PUSH flag is not set.) 1 - Insert PUSH flag into every decrypted record. 2 -Insert PUSH flag into every encrypted record. 3 - Insert PUSH flag into every decrypted and encrypted record. Minimum value = 0 Maximum value = 3
dropreqwithnohostheader	<String>	Read-write	Host header check for SNI enabled sessions. If this check is enabled and the HTTP request does not contain the host header for SNI enabled sessions, the request is dropped. Also, please note that when the check is disabled, the request will not be dropped even if there is a

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
			mis-match between SNI and HTTP host-header. Default value: NO Possible values = YES, NO
pushenctriggertimeout	<Double>	Read-write	PUSH encryption trigger timeout value. The timeout value is applied only if you set the Push Encryption Trigger parameter to Timer in the SSL virtual server settings. Default value: 1 Minimum value = 1 Maximum value = 200
cryptodevdisablelimit	<Double>	Read-write	Limit to the number of disabled SSL chips after which the ADC restarts. A value of zero implies that the ADC does not automatically restart. Default value: 0
undefactioncontrol	<String>	Read-write	Name of the undefined built-in control action: CLIENTAUTH, NOCLIENTAUTH, NOOP, RESET, or DROP.

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
			Default value: "CLIENTAUTH"
undefactiondata	<String>	Read-write	Name of the undefined built-in data action: NOOP, RESET or DROP. Default value: "NOOP"
defaultprofile	<String>	Read-write	Global parameter used to enable default profile feature. Default value: DISABLED Possible values = ENABLED, DISABLED
softwarecryptothreshold	<Double>	Read-write	Citrix ADC CPU utilization threshold (in percentage) beyond which crypto operations are not done in software. A value of zero implies that CPU is not utilized for doing crypto in software. Default value: 0 Minimum value = 0 Maximum value = 100

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
hybridfipsmode	<String>	Read-write	When this mode is enabled, system will use additional crypto hardware to accelerate symmetric crypto operations. Default value: DISABLED Possible values = ENABLED, DISABLED
sigdigesttype	<String[]>	Read-write	Signature Digest Algorithms that are supported by appliance. Default value is "ALL" and it will enable the following algorithms depending on the platform. On VPX: ECDSA-SHA1 ECDSA-SHA224 ECDSA-SHA256 ECDSA-SHA384 ECDSA-SHA512 RSA-SHA1 RSA-SHA224 RSA-SHA256 RSA-SHA384 RSA-SHA512 DSA-SHA1 DSA-SHA224 DSA-SHA256 DSA-SHA384 DSA-SHA512

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
			On MPX with Nitrox-III and coleto cards: RSA-SHA1 RSA-SHA224 RSA-SHA256 RSA-SHA384 RSA-SHA512 ECDSA-SHA1 ECDSA-SHA224 ECDSA-SHA256 ECDSA-SHA384 ECDSA-SHA512 Others: RSA-SHA1 RSA-SHA224 RSA-SHA256 RSA-SHA384 RSA-SHA512. Note:ALL doesnot include RSA-MD5 for any platform. Default value: ALL Possible values = ALL, RSA-MD5, RSA-SHA1, RSA-SHA224, RSA-SHA256, RSA-SHA384, RSA-SHA512, DSA-SHA1, DSA-SHA224, DSA-SHA256, DSA-SHA384, DSA-SHA512, ECDSA-

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
			SHA1, ECDSA-SHA224, ECDSA-SHA256, ECDSA-SHA384, ECDSA-SHA512
sslierrorcache	<String>	Read-write	Enable or disable dynamically learning and caching the learned information to make the subsequent interception or bypass decision. When enabled, NS does the lookup of this cached data to do early bypass. Default value: DISABLED Possible values = ENABLED, DISABLED
sslimaxerrorcachemem	<Double>	Read-write	Specify the maximum memory that can be used for caching the learned data. This memory is used as a LRU cache so that the old entries gets replaced with new entry once the set memory limit is fully utilised. A value of 0 decides the limit automatically.

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
			Default value: 0 Minimum value = 0 Maximum value = 4294967294
insertcertspace	<String>	Read-write	To insert space between lines in the certificate header of request. Default value: YES Possible values = YES, NO
ndcppcompliancecertcheck	<String>	Read-write	Applies when the Citrix ADC appliance acts as a client (back-end connection). Settings apply as follows: YES - During certificate verification, ignore the common name if SAN is present in the certificate. NO - Do not ignore common name. Default value: NO Possible values = YES, NO

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
svctls1112disable	<String>	Read-only	Disable TLS 1.1 and 1.2 for dynamic and VPN created services. Default value: NO Possible values = YES, NO
montls1112disable	<String>	Read-only	Disable TLS 1.1 and 1.2 for secure (https) monitors bound to SSL_BRIDGE services. Default value: NO Possible values = YES, NO

Operations

(click to see Properties)

UPDATE| UNSET| GET (ALL)

Some options that you can use for each operations:

- **Getting warnings in response:**NITRO allows you to get warnings in an operation by specifying the "warning" query parameter as "yes". For example, to get warnings while connecting to the Citrix ADC appliance, the URL is as follows:

  http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/login?warning=yes

  If any, the warnings are displayed in the response payload with the HTTP code "209 X-NITRO-WARNING".

- **Authenticated access for individual NITRO operations:**NITRO allows you to logon to the Citrix ADC appliance to perform individual operations. You can use this option instead of creating a NITRO session (using the login object) and then using that session to perform all operations,

To do this, you must specify the username and password in the request header of the NITRO request as follows:

X-NITRO-USER:<username>

X-NITRO-PASS:<password>

Note: In such cases, make sure that the request header DOES not include the following:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue>

`*Note:`*

Mandatory parameters are marked in redand placeholder content is marked in <green>.

update

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/sslparameter

HTTP Method:PUT

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Content-Type:application/json

Request Payload:

```
{"sslparameter":{

"quantumsize":<String_value>,

"crlmemorysizemb":<Double_value>,

"strictcachecks":<String_value>,

"ssltriggertimeout":<Double_value>,

"sendclosenotify":<String_value>,

"encrypttriggerpktcount":<Double_value>,

"denysslreneg":<String_value>,
```

```
"insertionencoding":<String_value>,

"ocspcachesize":<Double_value>,

"pushflag":<Double_value>,

"dropreqwithnohostheader":<String_value>,

"pushenctriggertimeout":<Double_value>,

"cryptodevdisablelimit":<Double_value>,

"undefactioncontrol":<String_value>,

"undefactiondata":<String_value>,

"defaultprofile":<String_value>,

"softwarecryptothreshold":<Double_value>,

"hybridfipsmode":<String_value>,

"sigdigesttype":<String[]_value>,

"sslierrorcache":<String_value>,

"sslimaxerrorcachemem":<Double_value>,

"insertcertspace":<String_value>,

"ndcppcompliancecertcheck":<String_value>

}}
```

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error

unset

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/sslparameter?**action=unset**

HTTP Method:POST

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Content-Type:application/json

Request Payload:

```
{"sslparameter":{

"quantumsize":true,

"crlmemorysizemb":true,

"strictcachecks":true,

"ssltriggertimeout":true,

"sendclosenotify":true,

"encrypttriggerpktcount":true,

"denysslreneg":true,

"insertionencoding":true,

"ocspcachesize":true,

"pushflag":true,

"dropreqwithnohostheader":true,

"pushenctriggertimeout":true,

"cryptodevdisablelimit":true,

"undefactioncontrol":true,
```

```
"undefactiondata":true,

"defaultprofile":true,

"softwarecryptothreshold":true,

"hybridfipsmode":true,

"sigdigesttype":true,

"sslierrorcache":true,

"sslimaxerrorcachemem":true,

"insertcertspace":true,

"ndcppcompliancecertcheck":true

}}
```

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error

get (all)

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/sslparameter

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "sslparameter": [ {

"quantumsize":<String_value>,

"crlmemorysizemb":<Double_value>,

"strictcachecks":<String_value>,

"ssltriggertimeout":<Double_value>,

"sendclosenotify":<String_value>,

"encrypttriggerpktcount":<Double_value>,

"denysslreneg":<String_value>,

"insertionencoding":<String_value>,

"ocspcachesize":<Double_value>,

"pushflag":<Double_value>,

"dropreqwithnohostheader":<String_value>,

"pushenctriggertimeout":<Double_value>,

"cryptodevdisablelimit":<Double_value>,

"undefactioncontrol":<String_value>,

"undefactiondata":<String_value>,

"defaultprofile":<String_value>,

"svctls1112disable":<String_value>,

"montls1112disable":<String_value>,
```

```
"softwarecryptothreshold":<Double_value>,

"hybridfipsmode":<String_value>,

"sigdigesttype":<String[]_value>,

"sslierrorcache":<String_value>,

"sslimaxerrorcachemem":<Double_value>,

"insertcertspace":<String_value>,

"ndcppcompliancecertcheck":<String_value>

}]}
```

sslpkcs12

Configuration for pkcs12 resource.

Properties

(click to see Operations)

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
outfile	<String>	Read-write	Name for and, optionally, path to, the output file that contains the certificate and the private key after converting from PKCS#12 to PEM format. /nsconfig/ssl/ is the default path. If importing, the certificate-key pair is stored in PEM format. If exporting, the certificate-key

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
			pair is stored in PKCS#12 format. Maximum length = 63
Import	<Boolean>	Read-write	Convert the certificate and private-key from PKCS#12 format to PEM format.
pkcs12file	<String>	Read-write	Name for and, optionally, path to, the PKCS#12 file. If importing, specify the input file name that contains the certificate and the private key in PKCS#12 format. If exporting, specify the output file name that contains the certificate and the private key after converting from PEM to PKCS#12 format. /nsconfig/ssl/ is the default path. During the import operation, if the key is encrypted, you are prompted to enter the pass phrase used for encrypting the key.

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
			Maximum length = 63
des	<Boolean>	Read-write	Encrypt the private key by using the DES algorithm in CBC mode during the import operation. On the command line, you are prompted to enter the pass phrase.
des3	<Boolean>	Read-write	Encrypt the private key by using the Triple-DES algorithm in EDE CBC mode (168-bit key) during the import operation. On the command line, you are prompted to enter the pass phrase.
aes256	<Boolean>	Read-write	Encrypt the private key by using the AES algorithm (256-bit key) during the import operation. On the command line, you are prompted to enter the pass phrase.
export	<Boolean>	Read-write	Convert the certificate and private key from PEM format to PKCS#12 format. On the command line, you are prompted to

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
			enter the pass phrase.
certfile	<String>	Read-write	Certificate file to be converted from PEM to PKCS#12 format. Maximum length = 63
keyfile	<String>	Read-write	Name of the private key file to be converted from PEM to PKCS#12 format. If the key file is encrypted, you are prompted to enter the pass phrase used for encrypting the key. Maximum length = 63
password	<String>	Read-write	. Minimum length = 1 Maximum length = 31
pempassphrase	<String>	Read-write	. Minimum length = 1 Maximum length = 31

Operations

(click to see Properties)

CONVERT

Some options that you can use for each operations:

- **Getting warnings in response:**NITRO allows you to get warnings in an operation by specifying the "warning" query parameter as "yes". For example, to get warnings while connecting to the Citrix ADC appliance, the URL is as follows:

  http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/login?warning=yes

  If any, the warnings are displayed in the response payload with the HTTP code "209 X-NITRO-WARNING".

- **Authenticated access for individual NITRO operations:**NITRO allows you to logon to the Citrix ADC appliance to perform individual operations. You can use this option instead of creating a NITRO session (using the login object) and then using that session to perform all operations,

  To do this, you must specify the username and password in the request header of the NITRO request as follows:

  X-NITRO-USER:<username>

  X-NITRO-PASS:<password>

  Note: In such cases, make sure that the request header DOES not include the following:

  Cookie:NITRO_AUTH_TOKEN=<tokenvalue>

`*Note:`*

Mandatory parameters are marked in redand placeholder content is marked in <green>.

convert

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/sslpkcs12?**action=convert**

HTTP Method:POST

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Content-Type:application/json

Request Payload:

```
{"sslpkcs12":{
```

```
<b>"outfile":<String_value>,

</b>"Import":<Boolean_value>,

"pkcs12file":<String_value>,

"des":<Boolean_value>,

"des3":<Boolean_value>,

"aes256":<Boolean_value>,

"export":<Boolean_value>,

"certfile":<String_value>,

"keyfile":<String_value>,

<b>"password":<String_value>,

</b>"pempassphrase":<String_value>

}}
```

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error

sslpkcs8

Configuration for pkcs8 resource.

Properties

(click to see Operations)

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
pkcs8file	<String>	Read-write	Name for and, optionally, path to, the output file where the PKCS#8 format key file is stored. /nsconfig/ssl/ is the default path. Maximum length = 63
keyfile	<String>	Read-write	Name of and, optionally, path to the input key file to be converted from PEM or DER format to PKCS#8 format. /nsconfig/ssl/ is the default path. Maximum length = 63
keyform	<String>	Read-write	Format in which the key file is stored on the appliance. Default value: PEM Possible values = DER, PEM
password	<String>	Read-write	Password to assign to the file if the key is encrypted. Applies only for PEM format files. Minimum length = 1 Maximum length = 31

Operations

(click to see Properties)

CONVERT

Some options that you can use for each operations:

- **Getting warnings in response:**NITRO allows you to get warnings in an operation by specifying the "warning" query parameter as "yes". For example, to get warnings while connecting to the Citrix ADC appliance, the URL is as follows:

  http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/login?warning=yes

  If any, the warnings are displayed in the response payload with the HTTP code "209 X-NITRO-WARNING".

- **Authenticated access for individual NITRO operations:**NITRO allows you to logon to the Citrix ADC appliance to perform individual operations. You can use this option instead of creating a NITRO session (using the login object) and then using that session to perform all operations,

  To do this, you must specify the username and password in the request header of the NITRO request as follows:

  X-NITRO-USER:<username>

  X-NITRO-PASS:<password>

  Note: In such cases, make sure that the request header DOES not include the following:

  Cookie:NITRO_AUTH_TOKEN=<tokenvalue>

`*Note:`*

Mandatory parameters are marked in redand placeholder content is marked in <green>.

convert

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/sslpkcs8?**action=convert**

HTTP Method:POST

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Content-Type:application/json

Request Payload:

```
{"sslpkcs8":{

<b>"pkcs8file":<String_value>,

</b><b>"keyfile":<String_value>,

</b>"keyform":<String_value>,

"password":<String_value>

}}
```

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error

sslpolicy

Configuration for SSL policy resource.

Properties

(click to see Operations)

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
name	<String>	Read-write	Name for the new SSL policy. Must begin with an ASCII alphanumeric or underscore (_) character, and must contain only ASCII alphanumeric, underscore, hash (#), period (.), space,

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
			colon (:), at (@), equals (=), and hyphen (-) characters. Cannot be changed after the policy is created. The following requirement applies only to the Citrix ADC CLI: If the name includes one or more spaces, enclose the name in double or single quotation marks (for example, "my policy" or 'my policy'). Minimum length = 1
rule	<String>	Read-write	Expression, against which traffic is evaluated. The following requirements apply only to the Citrix ADC CLI: * If the expression includes one or more spaces, enclose the entire expression in double quotation marks.

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
			* If the expression itself includes double quotation marks, escape the quotations by using the character. * Alternatively, you can use single quotation marks to enclose the rule, in which case you do not have to escape the double quotation marks.
reqaction	<String>	Read-write	The name of the action to be performed on the request. Refer to 'add ssl action' command to add a new action. Builtin actions like NOOP, RESET, DROP, CLIENTAUTH and NOCLIENTAUTH are also allowed. Minimum length = 1
action	<String>	Read-write	Name of the built-in or user-defined action to perform on the request. Available built-in actions are NOOP, RESET, DROP, CLIENTAUTH, NOCLIENTAUTH,

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
			INTERCEPT AND BYPASS.
undefaction	<String>	Read-write	Name of the action to be performed when the result of rule evaluation is undefined. Possible values for control policies: CLIENTAUTH, NOCLIENTAUTH, NOOP, RESET, DROP. Possible values for data policies: NOOP, RESET, DROP and BYPASS.
comment	<String>	Read-write	Any comments associated with this policy.
hits	<Double>	Read-only	Number of hits for this policy.
undefhits	<Double>	Read-only	Number of Undef hits.
description	<String>	Read-only	Description of the policy.
policytype	<String>	Read-only	. Possible values = Classic Policy, Advanced Policy
__count	<Double>	Read-only	count parameter

Operations

(click to see Properties)

ADD| DELETE| UPDATE| UNSET| GET (ALL)| GET| COUNT

Some options that you can use for each operations:

- **Getting warnings in response:**NITRO allows you to get warnings in an operation by specifying the "warning" query parameter as "yes". For example, to get warnings while connecting to the Citrix ADC appliance, the URL is as follows:

  http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/login?warning=yes

  If any, the warnings are displayed in the response payload with the HTTP code "209 X-NITRO-WARNING".

- **Authenticated access for individual NITRO operations:**NITRO allows you to logon to the Citrix ADC appliance to perform individual operations. You can use this option instead of creating a NITRO session (using the login object) and then using that session to perform all operations,

  To do this, you must specify the username and password in the request header of the NITRO request as follows:

  X-NITRO-USER:<username>

  X-NITRO-PASS:<password>

  Note: In such cases, make sure that the request header DOES not include the following:

  Cookie:NITRO_AUTH_TOKEN=<tokenvalue>

`*Note:`*

Mandatory parameters are marked in redand placeholder content is marked in <green>.

add

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/sslpolicy

HTTP Method:POST

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Content-Type:application/json

Request Payload:

```
{"sslpolicy":{

<b>"name":<String_value>,

</b><b>"rule":<String_value>,

</b>"reqaction":<String_value>,

"action":<String_value>,

"undefaction":<String_value>,

"comment":<String_value>

}}
```

Response:

HTTP Status Code on Success: 201 Created HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error

delete

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/sslpolicy/name_value<String>

HTTP Method:DELETE

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue>

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error

update

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/sslpolicy

HTTP Method:PUT

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Content-Type:application/json

Request Payload:

```
{"sslpolicy":{

<b>"name":<String_value>,

</b>"rule":<String_value>,

"action":<String_value>,

"undefaction":<String_value>,

"comment":<String_value>

}}
```

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error

unset

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/sslpolicy?**action=unset**

HTTP Method:POST

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Content-Type:application/json

Request Payload:

```
{"sslpolicy":{

<b>"name":<String_value>,

</b>"undefaction":true,

"comment":true

}}
```

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error

get (all)

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/sslpolicy

Query-parameters:

attrs

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/sslpolicy?**attrs=property-name1,property-name2**

Use this query parameter to specify the resource details that you want to retrieve.

filter

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/sslpolicy?**filter=property-name1:property-val1,property-name2:property-val2**

Use this query-parameter to get the filtered set of sslpolicy resources configured on Citrix ADC. Filtering can be done on any of the properties of the resource.

view

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/sslpolicy?**view=summary**

Note: By default, the retrieved results are displayed in detail view (?view=detail).

pagination

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/sslpolicy?**pagesize=#no;pageno=#no**

Use this query-parameter to get the sslpolicy resources in chunks.

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "sslpolicy": [ {

"name":<String_value>,

"rule":<String_value>,

"action":<String_value>,

"undefaction":<String_value>,

"hits":<Double_value>,

"undefhits":<Double_value>,

"description":<String_value>,

"comment":<String_value>,
```

```
"policytype":<String_value>

}]}
```

get

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/sslpolicy/name_value<String>

Query-parameters:

attrs

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/sslpolicy/name_value<String>?**attrs=property-name1,property-name2**

Use this query parameter to specify the resource details that you want to retrieve.

view

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/sslpolicy/name_value<String>?**view=summary**

Note: By default, the retrieved results are displayed in detail view (?view=detail).

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "sslpolicy": [ {
```

```
"name":<String_value>,

"rule":<String_value>,

"action":<String_value>,

"undefaction":<String_value>,

"hits":<Double_value>,

"undefhits":<Double_value>,

"description":<String_value>,

"comment":<String_value>,

"policytype":<String_value>

}]}
```

count

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/sslpolicy?**count=yes**

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "sslpolicy": [ { "__count": "#no"} ] }
```

sslpolicylabel

Configuration for SSL policy label resource.

Properties

(click to see Operations)

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
labelname	<String>	Read-write	Name for the SSL policy label. Must begin with an ASCII alphanumeric or underscore (_) character, and must contain only ASCII alphanumeric, underscore, hash (#), period (.), space, colon (:), at (@), equals (=), and hyphen (-) characters. Cannot be changed after the policy label is created. The following requirement applies only to the Citrix ADC CLI: If the name includes one or more spaces, enclose the name in double or single quotation

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
			marks (for example, "my label" or 'my label').
type	<String>	Read-write	Type of policies that the policy label can contain. Possible values = CONTROL, DATA
numpol	<Double>	Read-only	Number of polices bound to label.
hits	<Double>	Read-only	Number of times policy label was invoked.
policyname	<String>	Read-only	Name of the SSL policy to bind to the policy label.
priority	<Double>	Read-only	Specifies the priority of the policy.
gotopriorityexpression	<String>	Read-only	Expression specifying the priority of the next policy which will get evaluated if the current policy rule evaluates to TRUE.
invoke	<Boolean>	Read-only	Invoke flag.
labeltype	<String>	Read-only	Type of policy label invocation. Possible values = vserver, service, policylabel

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
invoke_labelname	<String>	Read-only	Name of the label to invoke if the current policy rule evaluates to TRUE.
flowtype	<Double>	Read-only	Flowtype of the bound SSL policy.
description	<String>	Read-only	Description of the policylabel.
__count	<Double>	Read-only	count parameter

Operations

(click to see Properties)

ADD| DELETE| GET (ALL)| GET| COUNT

Some options that you can use for each operations:

- **Getting warnings in response:**NITRO allows you to get warnings in an operation by specifying the "warning" query parameter as "yes". For example, to get warnings while connecting to the Citrix ADC appliance, the URL is as follows:

  http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/login?warning=yes

  If any, the warnings are displayed in the response payload with the HTTP code "209 X-NITRO-WARNING".

- **Authenticated access for individual NITRO operations:**NITRO allows you to logon to the Citrix ADC appliance to perform individual operations. You can use this option instead of creating a NITRO session (using the login object) and then using that session to perform all operations,

  To do this, you must specify the username and password in the request header of the NITRO request as follows:

  X-NITRO-USER:<username>

  X-NITRO-PASS:<password>

Note: In such cases, make sure that the request header DOES not include the following:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue>

`*Note:` *

Mandatory parameters are marked in redand placeholder content is marked in <green>.

add

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/sslpolicylabel

HTTP Method:POST

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Content-Type:application/json

Request Payload:

```
{"sslpolicylabel":{

<b>"labelname":<String_value>,

</b><b>"type":<String_value>

</b>}}
```

Response:

HTTP Status Code on Success: 201 Created HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error

delete

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/sslpolicylabel/labelname_value<String>

HTTP Method:DELETE

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue>

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error

get (all)

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/sslpolicylabel

Query-parameters:

attrs

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/sslpolicylabel?**attrs=property-name1,property-name2**

Use this query parameter to specify the resource details that you want to retrieve.

filter

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/sslpolicylabel?**filter=property-name1:property-val1,property-name2:property-val2**

Use this query-parameter to get the filtered set of sslpolicylabel resources configured on Citrix ADC. Filtering can be done on any of the properties of the resource.

view

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/sslpolicylabel?**view=summary**

Note: By default, the retrieved results are displayed in detail view (?view=detail).

pagination

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/sslpolicylabel?**pagesize=#no;pageno=#no**

Use this query-parameter to get the sslpolicylabel resources in chunks.

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "sslpolicylabel": [ {

"labelname":<String_value>,

"type":<String_value>,

"numpol":<Double_value>,

"hits":<Double_value>,

"policyname":<String_value>,

"priority":<Double_value>,

"gotopriorityexpression":<String_value>,

"invoke":<Boolean_value>,

"labeltype":<String_value>,

"invoke_labelname":<String_value>,

"flowtype":<Double_value>,

"description":<String_value>

}]}
```

get

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/sslpolicylabel/labelname_value<String>

Query-parameters:

attrs

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/sslpolicylabel/labelname_value<String>?**attrs=property-name1,property-name2**

Use this query parameter to specify the resource details that you want to retrieve.

view

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/sslpolicylabel/labelname_value<String>?**view=summary**

Note: By default, the retrieved results are displayed in detail view (?view=detail).

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "sslpolicylabel": [ {

"labelname":<String_value>,

"type":<String_value>,

"numpol":<Double_value>,

"hits":<Double_value>,

"policyname":<String_value>,
```

```
"priority":<Double_value>,

"gotopriorityexpression":<String_value>,

"invoke":<Boolean_value>,

"labeltype":<String_value>,

"invoke_labelname":<String_value>,

"flowtype":<Double_value>,

"description":<String_value>

}]}
```

count

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/sslpolicylabel?**count=yes**

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "sslpolicylabel": [ { "__count": "#no"} ] }
```

sslpolicylabel_binding

Binding object which returns the resources bound to sslpolicylabel.

Properties

(click to see Operations)

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
labelname	<String>	Read-write	Name of the SSL policy label for which to show detailed information.
sslpolicylabel_sslpolicy_binding	<sslpolicylabel_sslpolicy_binding[]>	Read-only	sslpolicy that can be bound to sslpolicylabel.

Operations

(click to see Properties)

GET| GET (ALL)

Some options that you can use for each operations:

- **Getting warnings in response:**NITRO allows you to get warnings in an operation by specifying the "warning" query parameter as "yes". For example, to get warnings while connecting to the Citrix ADC appliance, the URL is as follows:

  http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/login?warning=yes

  If any, the warnings are displayed in the response payload with the HTTP code "209 X-NITRO-WARNING".

- **Authenticated access for individual NITRO operations:**NITRO allows you to logon to the Citrix ADC appliance to perform individual operations. You can use this option instead of creating a NITRO session (using the login object) and then using that session to perform all operations,

  To do this, you must specify the username and password in the request header of the NITRO request as follows:

  X-NITRO-USER:<username>

X-NITRO-PASS:<password>

Note: In such cases, make sure that the request header DOES not include the following:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue>

`*Note:`*

Mandatory parameters are marked in redand placeholder content is marked in <green>.

get

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/sslpolicylabel_binding/labelname_value<String>

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "sslpolicylabel_binding": [ {

"labelname":<String_value>,

"sslpolicylabel_sslpolicy_binding":<sslpolicylabel_sslpolicy_binding[]_value>

}]}
```

get (all)

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/sslpolicylabel_binding

Query-parameters:

bulkbindings

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/sslpolicylabel_binding?**bulkbindings=yes**

NITRO allows you to fetch bindings in bulk.

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "sslpolicylabel_binding": [ {

"labelname":<String_value>,

"sslpolicylabel_sslpolicy_binding":<sslpolicylabel_sslpolicy_binding[]_val
ue>

}]}
```

sslpolicylabel_sslpolicy_binding

Binding object showing the sslpolicy that can be bound to sslpolicylabel.

Properties

(click to see Operations)

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
priority	<Double>	Read-write	Specifies the priority of the policy.
gotopriorityexpression	<String>	Read-write	Expression specifying the priority of the next policy which will get evaluated if the current policy rule evaluates to TRUE.
policyname	<String>	Read-write	Name of the SSL policy to bind to the policy label.
labeltype	<String>	Read-write	Type of policy label invocation. Possible values = vserver, service, policylabel
labelname	<String>	Read-write	Name of the SSL policy label to which to bind policies.
invoke_labelname	<String>	Read-write	Name of the label to invoke if the current policy rule evaluates to TRUE.
invoke	<Boolean>	Read-write	Invoke policies bound to a policy label. After the invoked policies are evaluated, the flow returns to the policy with the next priority.

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
__count	<Double>	Read-write	count parameter

Operations

(click to see Properties)

ADD:| DELETE:| GET| GET (ALL)| COUNT

Some options that you can use for each operations:

- **Getting warnings in response:**NITRO allows you to get warnings in an operation by specifying the "warning" query parameter as "yes". For example, to get warnings while connecting to the Citrix ADC appliance, the URL is as follows:

  http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/login?warning=yes

  If any, the warnings are displayed in the response payload with the HTTP code "209 X-NITRO-WARNING".
- **Authenticated access for individual NITRO operations:**NITRO allows you to logon to the Citrix ADC appliance to perform individual operations. You can use this option instead of creating a NITRO session (using the login object) and then using that session to perform all operations,

  To do this, you must specify the username and password in the request header of the NITRO request as follows:

  X-NITRO-USER:<username>

  X-NITRO-PASS:<password>

  Note: In such cases, make sure that the request header DOES not include the following:

  Cookie:NITRO_AUTH_TOKEN=<tokenvalue>

`*Note:`*

Mandatory parameters are marked in redand placeholder content is marked in <green>.

add

URL:http://<Citrix-ADC-IP-address(NSIP)/nitro/v1/config/sslpolicylabel_sslpolicy_binding

HTTP Method:PUT

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Content-Type:application/json

Request Payload:

```
{

"sslpolicylabel_sslpolicy_binding":{

<b>"labelname":<String_value>,

</b><b>"policyname":<String_value>,

</b><b>"priority":<Double_value>,

</b>"gotopriorityexpression":<String_value>,

"invoke":<Boolean_value>,

"labeltype":<String_value>,

"invoke_labelname":<String_value>

}}
```

Response:

HTTP Status Code on Success: 201 Created HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error

delete

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/sslpolicylabel_sslpolicy_binding/labelname_value<String>

Query-parameters:

args

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/sslpolicylabel_sslpolicy_binding/labelname_value<String>?**args=policyname:<String_value>,priority:<Double_value>**

HTTP Method:DELETE

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue>

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error

get

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/sslpolicylabel_sslpolicy_binding/labelname_value<String>

Query-parameters:

filter

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/sslpolicylabel_sslpolicy_binding/labelname_value<String>?**filter=property-name1:property-value1,property-name2:property-value2**

Use this query-parameter to get the filtered set of sslpolicylabel_sslpolicy_binding resources configured on Citrix ADC. Filtering can be done on any of the properties of the resource.

pagination

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/sslpolicylabel_sslpolicy_binding/labelname_value<String>?**pagesize=#no;pageno=#no**

Use this query-parameter to get the sslpolicylabel_sslpolicy_binding resources in chunks.

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "sslpolicylabel_sslpolicy_binding": [ {

"priority":<Double_value>,

"gotopriorityexpression":<String_value>,

"policyname":<String_value>,

"labeltype":<String_value>,

"labelname":<String_value>,

"invoke_labelname":<String_value>,

"invoke":<Boolean_value>

}]}
```

get (all)

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/sslpolicylabel_sslpolicy_binding

Query-parameters:

bulkbindings

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/sslpolicylabel_sslpolicy_binding?**bulkbindings=yes**

NITRO allows you to fetch bindings in bulk.

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "sslpolicylabel_sslpolicy_binding": [ {

"priority":<Double_value>,

"gotopriorityexpression":<String_value>,

"policyname":<String_value>,

"labeltype":<String_value>,

"labelname":<String_value>,

"invoke_labelname":<String_value>,

"invoke":<Boolean_value>

}]}
```

count

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/sslpolicylabel_sslpolicy_binding/labelname_value<String>?**count=yes**

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{"sslpolicylabel_sslpolicy_binding": [ { "__count": "#no"} ] }
```

sslpolicy_binding

Binding object which returns the resources bound to sslpolicy.

Properties

(click to see Operations)

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
name	<String>	Read-write	Name of the SSL policy for which to display detailed information. Minimum length = 1
sslpolicy_csvserver_binding	<sslpolicy_csvserver_binding[]>	Read-only	csvserver that can be bound to sslpolicy.
sslpolicy_sslvserver_binding	<sslpolicy_sslvserver_binding[]>	Read-only	sslvserver that can be bound to sslpolicy.

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
sslpolicy_sslpolicylabel_binding	<sslpolicy_sslpolicylabel_binding[]>	Read-only	sslpolicylabel that can be bound to sslpolicy.
sslpolicy_sslservice_binding	<sslpolicy_sslservice_binding[]>	Read-only	sslservice that can be bound to sslpolicy.
sslpolicy_lbvserver_binding	<sslpolicy_lbvserver_binding[]>	Read-only	lbvserver that can be bound to sslpolicy.
sslpolicy_sslglobal_binding	<sslpolicy_sslglobal_binding[]>	Read-only	sslglobal that can be bound to sslpolicy.

Operations

(click to see Properties)

GET| GET (ALL)

Some options that you can use for each operations:

- **Getting warnings in response:**NITRO allows you to get warnings in an operation by specifying the "warning" query parameter as "yes". For example, to get warnings while connecting to the Citrix ADC appliance, the URL is as follows:

  http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/login?warning=yes

  If any, the warnings are displayed in the response payload with the HTTP code "209 X-NITRO-WARNING".

- **Authenticated access for individual NITRO operations:**NITRO allows you to logon to the Citrix ADC appliance to perform individual operations. You can use this option instead of creating a NITRO session (using the login object) and then using that session to perform all operations,

  To do this, you must specify the username and password in the request header of the NITRO request as follows:

  X-NITRO-USER:<username>

  X-NITRO-PASS:<password>

Note: In such cases, make sure that the request header DOES not include the following:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue>

`*Note:` *

Mandatory parameters are marked in redand placeholder content is marked in <green>.

get

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/sslpolicy_binding/name_value<String>

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "sslpolicy_binding": [ {

"name":<String_value>,

"sslpolicy_csvserver_binding":<sslpolicy_csvserver_binding[]_value>,

"sslpolicy_sslvserver_binding":<sslpolicy_sslvserver_binding[]_value>,

"sslpolicy_lbvserver_binding":<sslpolicy_lbvserver_binding[]_value>,

"sslpolicy_sslservice_binding":<sslpolicy_sslservice_binding[]_value>,

"sslpolicy_sslpolicylabel_binding":<sslpolicy_sslpolicylabel_binding[]_val
ue>,
```

```
"sslpolicy_sslglobal_binding":<sslpolicy_sslglobal_binding[]_value>

}]}
```

get (all)

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/sslpolicy_binding

Query-parameters:

bulkbindings

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/sslpolicy_binding?**bulkbindings=yes**

NITRO allows you to fetch bindings in bulk.

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "sslpolicy_binding": [ {

"name":<String_value>,

"sslpolicy_csvserver_binding":<sslpolicy_csvserver_binding[]_value>,

"sslpolicy_sslvserver_binding":<sslpolicy_sslvserver_binding[]_value>,

"sslpolicy_lbvserver_binding":<sslpolicy_lbvserver_binding[]_value>,
```

```
"sslpolicy_sslservice_binding":<sslpolicy_sslservice_binding[]_value>,

"sslpolicy_sslpolicylabel_binding":<sslpolicy_sslpolicylabel_binding[]_val
ue>,

"sslpolicy_sslglobal_binding":<sslpolicy_sslglobal_binding[]_value>

}]}
```

sslpolicy_csvserver_binding

Binding object showing the csvserver that can be bound to sslpolicy.

Properties

(click to see Operations)

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
boundto	<String>	Read-write	The entity name to which policy is bound.
name	<String>	Read-write	Name of the SSL policy for which to display detailed information. Minimum length = 1
priority	<Double>	Read-only	.
labelname	<String>	Read-only	Name of the label to invoke if the current policy rule evaluates to TRUE.
gotopriorityexpression	<String>	Read-only	Expression specifying the priority of the next policy which will get evaluated if the

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
			current policy rule evaluates to TRUE.
labeltype	<String>	Read-only	Type of policy label invocation. Possible values = vserver, service, policylabel
activepolicy	<Double>	Read-only	.
__count	<Double>	Read-only	count parameter

Operations

(click to see Properties)

GET| GET (ALL)| COUNT

Some options that you can use for each operations:

- **Getting warnings in response:**NITRO allows you to get warnings in an operation by specifying the "warning" query parameter as "yes". For example, to get warnings while connecting to the Citrix ADC appliance, the URL is as follows:

  http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/login?warning=yes

  If any, the warnings are displayed in the response payload with the HTTP code "209 X-NITRO-WARNING".

- **Authenticated access for individual NITRO operations:**NITRO allows you to logon to the Citrix ADC appliance to perform individual operations. You can use this option instead of creating a NITRO session (using the login object) and then using that session to perform all operations,

  To do this, you must specify the username and password in the request header of the NITRO request as follows:

  X-NITRO-USER:<username>

  X-NITRO-PASS:<password>

Note: In such cases, make sure that the request header DOES not include the following:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue>

`*Note:` *

Mandatory parameters are marked in redand placeholder content is marked in <green>.

get

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/sslpolicy_csvserver_binding/name_value<String>

Query-parameters:

filter

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/sslpolicy_csvserver_binding/name_value<String>?**filter=property-name1:property-value1,property-name2:property-value2**

Use this query-parameter to get the filtered set of sslpolicy_csvserver_binding resources configured on Citrix ADC. Filtering can be done on any of the properties of the resource.

pagination

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/sslpolicy_csvserver_binding/name_value<String>?**pagesize=#no;pageno=#no**

Use this query-parameter to get the sslpolicy_csvserver_binding resources in chunks.

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "sslpolicy_csvserver_binding": [ {

"boundto":<String_value>,

"name":<String_value>,

"priority":<Double_value>,

"labelname":<String_value>,

"gotopriorityexpression":<String_value>,

"labeltype":<String_value>,

"activepolicy":<Double_value>

}]}
```

get (all)

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/sslpolicy_csvserver_binding

Query-parameters:

bulkbindings

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/sslpolicy_csvserver_binding?**bulkbindings=yes**

NITRO allows you to fetch bindings in bulk.

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "sslpolicy_csvserver_binding": [ {

"boundto":<String_value>,

"name":<String_value>,

"priority":<Double_value>,

"labelname":<String_value>,

"gotopriorityexpression":<String_value>,

"labeltype":<String_value>,

"activepolicy":<Double_value>

}]}
```

count

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/sslpolicy_csvserver_binding/name_value<String>?**count=yes**

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{"sslpolicy_csvserver_binding": [ { "__count": "#no"} ] }
```

sslpolicy_lbvserver_binding

Binding object showing the lbvserver that can be bound to sslpolicy.

Properties

(click to see Operations)

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
boundto	<String>	Read-write	The entity name to which policy is bound.
name	<String>	Read-write	Name of the SSL policy for which to display detailed information. Minimum length = 1
priority	<Double>	Read-only	.
labelname	<String>	Read-only	Name of the label to invoke if the current policy rule evaluates to TRUE.
gotopriorityexpression	<String>	Read-only	Expression specifying the priority of the next policy

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
			which will get evaluated if the current policy rule evaluates to TRUE.
labeltype	<String>	Read-only	Type of policy label invocation. Possible values = vserver, service, policylabel
activepolicy	<Double>	Read-only	.
__count	<Double>	Read-only	count parameter

Operations

(click to see Properties)

GET| GET (ALL)| COUNT

Some options that you can use for each operations:

- **Getting warnings in response:**NITRO allows you to get warnings in an operation by specifying the "warning" query parameter as "yes". For example, to get warnings while connecting to the Citrix ADC appliance, the URL is as follows:

  http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/login?warning=yes

  If any, the warnings are displayed in the response payload with the HTTP code "209 X-NITRO-WARNING".

- **Authenticated access for individual NITRO operations:**NITRO allows you to logon to the Citrix ADC appliance to perform individual operations. You can use this option instead of creating a NITRO session (using the login object) and then using that session to perform all operations,

  To do this, you must specify the username and password in the request header of the NITRO request as follows:

  X-NITRO-USER:<username>

X-NITRO-PASS:<password>

Note: In such cases, make sure that the request header DOES not include the following:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue>

`*Note:`*

Mandatory parameters are marked in redand placeholder content is marked in <green>.

get

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/sslpolicy_lbvserver_binding/name_value<String>

Query-parameters:

filter

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/sslpolicy_lbvserver_binding/name_value<String>?**filter=property-name1:property-value1,property-name2:property-value2**

Use this query-parameter to get the filtered set of sslpolicy_lbvserver_binding resources configured on Citrix ADC. Filtering can be done on any of the properties of the resource.

pagination

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/sslpolicy_lbvserver_binding/name_value<String>?**pagesize=#no;pageno=#no**

Use this query-parameter to get the sslpolicy_lbvserver_binding resources in chunks.

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "sslpolicy_lbvserver_binding": [ {

"boundto":<String_value>,

"name":<String_value>,

"priority":<Double_value>,

"labelname":<String_value>,

"gotopriorityexpression":<String_value>,

"labeltype":<String_value>,

"activepolicy":<Double_value>

}]}
```

get (all)

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/sslpolicy_lbvserver_binding

Query-parameters:

bulkbindings

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/sslpolicy_lbvserver_binding?**bulkbindings=yes**

NITRO allows you to fetch bindings in bulk.

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "sslpolicy_lbvserver_binding": [ {

"boundto":<String_value>,

"name":<String_value>,

"priority":<Double_value>,

"labelname":<String_value>,

"gotopriorityexpression":<String_value>,

"labeltype":<String_value>,

"activepolicy":<Double_value>

}]}
```

count

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/sslpolicy_lbvserver_binding/name_value<String>?**count=yes**

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{"sslpolicy_lbvserver_binding": [ { "__count": "#no"} ] }
```

sslpolicy_sslglobal_binding

Binding object showing the sslglobal that can be bound to sslpolicy.

Properties

(click to see Operations)

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
boundto	<String>	Read-write	The entity name to which policy is bound.
name	<String>	Read-write	Name of the SSL policy for which to display detailed information. Minimum length = 1
priority	<Double>	Read-only	.
labelname	<String>	Read-only	Name of the label to invoke if the current policy rule evaluates to TRUE.
gotopriorityexpression	<String>	Read-only	Expression specifying the priority of the next policy

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
			which will get evaluated if the current policy rule evaluates to TRUE.
labeltype	<String>	Read-only	Type of policy label invocation. Possible values = vserver, service, policylabel
activepolicy	<Double>	Read-only	.
__count	<Double>	Read-only	count parameter

Operations

(click to see Properties)

GET| GET (ALL)| COUNT

Some options that you can use for each operations:

- **Getting warnings in response:**NITRO allows you to get warnings in an operation by specifying the "warning" query parameter as "yes". For example, to get warnings while connecting to the Citrix ADC appliance, the URL is as follows:

  http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/login?warning=yes

  If any, the warnings are displayed in the response payload with the HTTP code "209 X-NITRO-WARNING".

- **Authenticated access for individual NITRO operations:**NITRO allows you to logon to the Citrix ADC appliance to perform individual operations. You can use this option instead of creating a NITRO session (using the login object) and then using that session to perform all operations,

  To do this, you must specify the username and password in the request header of the NITRO request as follows:

  X-NITRO-USER:<username>

X-NITRO-PASS:<password>

Note: In such cases, make sure that the request header DOES not include the following:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue>

`*Note:`*

Mandatory parameters are marked in redand placeholder content is marked in <green>.

get

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/sslpolicy_sslglobal_binding/name_value<String>

Query-parameters:

filter

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/sslpolicy_sslglobal_binding/name_value<String>?**filter=property-name1:property-value1,property-name2:property-value2**

Use this query-parameter to get the filtered set of sslpolicy_sslglobal_binding resources configured on Citrix ADC. Filtering can be done on any of the properties of the resource.

pagination

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/sslpolicy_sslglobal_binding/name_value<String>?**pagesize=#no;pageno=#no**

Use this query-parameter to get the sslpolicy_sslglobal_binding resources in chunks.

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "sslpolicy_sslglobal_binding": [ {

"boundto":<String_value>,

"name":<String_value>,

"priority":<Double_value>,

"labelname":<String_value>,

"gotopriorityexpression":<String_value>,

"labeltype":<String_value>,

"activepolicy":<Double_value>

}]}
```

get (all)

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/sslpolicy_sslglobal_binding

Query-parameters:

bulkbindings

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/sslpolicy_sslglobal_binding?**bulkbindings=yes**

NITRO allows you to fetch bindings in bulk.

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "sslpolicy_sslglobal_binding": [ {

"boundto":<String_value>,

"name":<String_value>,

"priority":<Double_value>,

"labelname":<String_value>,

"gotopriorityexpression":<String_value>,

"labeltype":<String_value>,

"activepolicy":<Double_value>

}]}
```

count

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/sslpolicy_sslglobal_binding/name_value<String>?**count=yes**

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{"sslpolicy_sslglobal_binding": [ { "__count": "#no"} ] }
```

sslpolicy_sslpolicylabel_binding

Binding object showing the sslpolicylabel that can be bound to sslpolicy.

Properties

(click to see Operations)

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
boundto	<String>	Read-write	The entity name to which policy is bound.
name	<String>	Read-write	Name of the SSL policy for which to display detailed information. Minimum length = 1
priority	<Double>	Read-only	.
labelname	<String>	Read-only	Name of the label to invoke if the current policy rule evaluates to TRUE.
gotopriorityexpression	<String>	Read-only	Expression specifying the priority of the next policy

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
			which will get evaluated if the current policy rule evaluates to TRUE.
labeltype	<String>	Read-only	Type of policy label invocation. Possible values = vserver, service, policylabel
activepolicy	<Double>	Read-only	.
__count	<Double>	Read-only	count parameter

Operations

(click to see Properties)

GET| GET (ALL)| COUNT

Some options that you can use for each operations:

- **Getting warnings in response:**NITRO allows you to get warnings in an operation by specifying the "warning" query parameter as "yes". For example, to get warnings while connecting to the Citrix ADC appliance, the URL is as follows:

  http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/login?warning=yes

  If any, the warnings are displayed in the response payload with the HTTP code "209 X-NITRO-WARNING".

- **Authenticated access for individual NITRO operations:**NITRO allows you to logon to the Citrix ADC appliance to perform individual operations. You can use this option instead of creating a NITRO session (using the login object) and then using that session to perform all operations,

  To do this, you must specify the username and password in the request header of the NITRO request as follows:

  X-NITRO-USER:<username>

X-NITRO-PASS:<password>

Note: In such cases, make sure that the request header DOES not include the following:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue>

`*Note:`*

Mandatory parameters are marked in redand placeholder content is marked in <green>.

get

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/sslpolicy_sslpolicylabel_binding/name_value<String>

Query-parameters:

filter

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/sslpolicy_sslpolicylabel_binding/name_value<String>?**filter=property-name1:property-value1,property-name2:property-value2**

Use this query-parameter to get the filtered set of sslpolicy_sslpolicylabel_binding resources configured on Citrix ADC. Filtering can be done on any of the properties of the resource.

pagination

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/sslpolicy_sslpolicylabel_binding/name_value<String>?**pagesize=#no;pageno=#no**

Use this query-parameter to get the sslpolicy_sslpolicylabel_binding resources in chunks.

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "sslpolicy_sslpolicylabel_binding": [ {

"boundto":<String_value>,

"name":<String_value>,

"priority":<Double_value>,

"labelname":<String_value>,

"gotopriorityexpression":<String_value>,

"labeltype":<String_value>,

"activepolicy":<Double_value>

}]}
```

get (all)

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/sslpolicy_sslpolicylabel_binding

Query-parameters:

bulkbindings

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/sslpolicy_sslpolicylabel_binding?**bulkbindings=yes**

NITRO allows you to fetch bindings in bulk.

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "sslpolicy_sslpolicylabel_binding": [ {

"boundto":<String_value>,

"name":<String_value>,

"priority":<Double_value>,

"labelname":<String_value>,

"gotopriorityexpression":<String_value>,

"labeltype":<String_value>,

"activepolicy":<Double_value>

}]}
```

count

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/sslpolicy_sslpolicylabel_binding/name_value<String>?**count=yes**

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{"sslpolicy_sslpolicylabel_binding": [ { "__count": "#no"} ] }
```

sslpolicy_sslservice_binding

Binding object showing the sslservice that can be bound to sslpolicy.

Properties

(click to see Operations)

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
boundto	<String>	Read-write	The entity name to which policy is bound.
name	<String>	Read-write	Name of the SSL policy for which to display detailed information. Minimum length = 1
priority	<Double>	Read-only	.
labelname	<String>	Read-only	Name of the label to invoke if the current policy rule evaluates to TRUE.
gotopriorityexpression	<String>	Read-only	Expression specifying the priority of the next policy

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
			which will get evaluated if the current policy rule evaluates to TRUE.
labeltype	<String>	Read-only	Type of policy label invocation. Possible values = vserver, service, policylabel
activepolicy	<Double>	Read-only	.
__count	<Double>	Read-only	count parameter

Operations

(click to see Properties)

GET| GET (ALL)| COUNT

Some options that you can use for each operations:

- **Getting warnings in response:**NITRO allows you to get warnings in an operation by specifying the "warning" query parameter as "yes". For example, to get warnings while connecting to the Citrix ADC appliance, the URL is as follows:

  http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/login?warning=yes

  If any, the warnings are displayed in the response payload with the HTTP code "209 X-NITRO-WARNING".

- **Authenticated access for individual NITRO operations:**NITRO allows you to logon to the Citrix ADC appliance to perform individual operations. You can use this option instead of creating a NITRO session (using the login object) and then using that session to perform all operations,

  To do this, you must specify the username and password in the request header of the NITRO request as follows:

  X-NITRO-USER:<username>

X-NITRO-PASS:<password>

Note: In such cases, make sure that the request header DOES not include the following:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue>

`*Note:`*

Mandatory parameters are marked in redand placeholder content is marked in <green>.

get

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/sslpolicy_sslservice_binding/name_value<String>

Query-parameters:

filter

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/sslpolicy_sslservice_binding/name_value<String>?**filter=property-name1:property-value1,property-name2:property-value2**

Use this query-parameter to get the filtered set of sslpolicy_sslservice_binding resources configured on Citrix ADC. Filtering can be done on any of the properties of the resource.

pagination

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/sslpolicy_sslservice_binding/name_value<String>?**pagesize=#no;pageno=#no**

Use this query-parameter to get the sslpolicy_sslservice_binding resources in chunks.

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "sslpolicy_sslservice_binding": [ {

"boundto":<String_value>,

"name":<String_value>,

"priority":<Double_value>,

"labelname":<String_value>,

"gotopriorityexpression":<String_value>,

"labeltype":<String_value>,

"activepolicy":<Double_value>

}]}
```

get (all)

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/sslpolicy_sslservice_binding

Query-parameters:

bulkbindings

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/sslpolicy_sslservice_binding?**bulkbindings=yes**

NITRO allows you to fetch bindings in bulk.

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "sslpolicy_sslservice_binding": [ {

"boundto":<String_value>,

"name":<String_value>,

"priority":<Double_value>,

"labelname":<String_value>,

"gotopriorityexpression":<String_value>,

"labeltype":<String_value>,

"activepolicy":<Double_value>

}]}
```

count

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/sslpolicy_sslservice_binding/name_value<String>?**count=yes**

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{"sslpolicy_sslservice_binding": [ { "__count": "#no"} ] }
```

sslpolicy_sslvserver_binding

Binding object showing the sslvserver that can be bound to sslpolicy.

Properties

(click to see Operations)

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
boundto	<String>	Read-write	The entity name to which policy is bound.
name	<String>	Read-write	Name of the SSL policy for which to display detailed information. Minimum length = 1
priority	<Double>	Read-only	.
labelname	<String>	Read-only	Name of the label to invoke if the current policy rule evaluates to TRUE.
gotopriorityexpression	<String>	Read-only	Expression specifying the priority of the next policy

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
			which will get evaluated if the current policy rule evaluates to TRUE.
labeltype	<String>	Read-only	Type of policy label invocation. Possible values = vserver, service, policylabel
activepolicy	<Double>	Read-only	.
__count	<Double>	Read-only	count parameter

Operations

(click to see Properties)

GET| GET (ALL)| COUNT

Some options that you can use for each operations:

- **Getting warnings in response:**NITRO allows you to get warnings in an operation by specifying the "warning" query parameter as "yes". For example, to get warnings while connecting to the Citrix ADC appliance, the URL is as follows:

  http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/login?warning=yes

  If any, the warnings are displayed in the response payload with the HTTP code "209 X-NITRO-WARNING".

- **Authenticated access for individual NITRO operations:**NITRO allows you to logon to the Citrix ADC appliance to perform individual operations. You can use this option instead of creating a NITRO session (using the login object) and then using that session to perform all operations,

  To do this, you must specify the username and password in the request header of the NITRO request as follows:

  X-NITRO-USER:<username>

X-NITRO-PASS:<password>

Note: In such cases, make sure that the request header DOES not include the following:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue>

`*Note:`*

Mandatory parameters are marked in redand placeholder content is marked in <green>.

get

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/sslpolicy_sslvserver_binding/name_value<String>

Query-parameters:

filter

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/sslpolicy_sslvserver_binding/name_value<String>?**filter=property-name1:property-value1,property-name2:property-value2**

Use this query-parameter to get the filtered set of sslpolicy_sslvserver_binding resources configured on Citrix ADC. Filtering can be done on any of the properties of the resource.

pagination

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/sslpolicy_sslvserver_binding/name_value<String>?**pagesize=#no;pageno=#no**

Use this query-parameter to get the sslpolicy_sslvserver_binding resources in chunks.

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "sslpolicy_sslvserver_binding": [ {

"boundto":<String_value>,

"name":<String_value>,

"priority":<Double_value>,

"labelname":<String_value>,

"gotopriorityexpression":<String_value>,

"labeltype":<String_value>,

"activepolicy":<Double_value>

}]}
```

get (all)

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/sslpolicy_sslvserver_binding

Query-parameters:

bulkbindings

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/sslpolicy_sslvserver_binding?**bulkbindings=yes**

NITRO allows you to fetch bindings in bulk.

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "sslpolicy_sslvserver_binding": [ {

"boundto":<String_value>,

"name":<String_value>,

"priority":<Double_value>,

"labelname":<String_value>,

"gotopriorityexpression":<String_value>,

"labeltype":<String_value>,

"activepolicy":<Double_value>

}]}
```

count

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/sslpolicy_sslvserver_binding/name_value<String>?**count=yes**

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{"sslpolicy_sslvserver_binding": [ { "__count": "#no"} ] }
```

sslprofile

Configuration for SSL profile resource.

Properties

(click to see Operations)

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
name	<String>	Read-write	Name for the SSL profile. Must begin with an ASCII alphanumeric or underscore (_) character, and must contain only ASCII alphanumeric, underscore, hash (#), period (.), space, colon (:), at (@), equals (=), and hyphen (-) characters. Cannot be changed after the profile is created. Minimum length = 1

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
			Maximum length = 127
sslprofiletype	<String>	Read-write	Type of profile. Front end profiles apply to the entity that receives requests from a client. Backend profiles apply to the entity that sends client requests to a server. Default value: FrontEnd Possible values = BackEnd, FrontEnd
ssllogprofile	<String>	Read-write	The name of the ssllogprofile. Minimum length = 1 Maximum length = 127
dhcount	<Double>	Read-write	Number of interactions, between the client and the Citrix ADC, after which the DH private-public pair is regenerated. A value of zero (0) specifies infinite use (no refresh). This parameter is not applicable when configuring a backend profile. Allowed DH count

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
			values are 0 and >= 500. Minimum value = 0 Maximum value = 65534
dh	<String>	Read-write	State of Diffie-Hellman (DH) key exchange. This parameter is not applicable when configuring a backend profile. Default value: DISABLED Possible values = ENABLED, DISABLED
dhfile	<String>	Read-write	The file name and path for the DH parameter. Minimum length = 1
ersa	<String>	Read-write	State of Ephemeral RSA (eRSA) key exchange. Ephemeral RSA allows clients that support only export ciphers to communicate with the secure server even if the server certificate does not support export clients. The

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
			ephemeral RSA key is automatically generated when you bind an export cipher to an SSL or TCP-based SSL virtual server or service. When you remove the export cipher, the eRSA key is not deleted. It is reused at a later date when another export cipher is bound to an SSL or TCP-based SSL virtual server or service. The eRSA key is deleted when the appliance restarts. This parameter is not applicable when configuring a backend profile. Default value: ENABLED Possible values = ENABLED, DISABLED
ersacount	<Double>	Read-write	The refresh count for the re-generation of RSA public-key and private-key pair. Minimum value = 0 Maximum value = 65534

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
sessreuse	<String>	Read-write	State of session reuse. Establishing the initial handshake requires CPU-intensive public key encryption operations. With the ENABLED setting, session key exchange is avoided for session resumption requests received from the client. Default value: ENABLED Possible values = ENABLED, DISABLED
sesstimeout	<Double>	Read-write	The Session timeout value in seconds. Minimum value = 0 Maximum value = 4294967294
cipherredirect	<String>	Read-write	State of Cipher Redirect. If this parameter is set to ENABLED, you can configure an SSL virtual server or service to display meaningful error messages if the SSL handshake fails because of a cipher mismatch between

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
			the virtual server or service and the client. This parameter is not applicable when configuring a backend profile. Default value: DISABLED Possible values = ENABLED, DISABLED
cipherurl	<String>	Read-write	The redirect URL to be used with the Cipher Redirect feature.
clientauth	<String>	Read-write	State of client authentication. In service-based SSL offload, the service terminates the SSL handshake if the SSL client does not provide a valid certificate. This parameter is not applicable when configuring a backend profile. Default value: DISABLED Possible values = ENABLED, DISABLED

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
clientcert	<String>	Read-write	The rule for client certificate requirement in client authentication. Possible values = Mandatory, Optional
dhkeyexpsizelimit	<String>	Read-write	This option enables the use of NIST recommended (NIST Special Publication 800-56A) bit size for private-key size. For example, for DH params of size 2048bit, the private-key size recommended is 224bits. This is rounded-up to 256bits. Default value: DISABLED Possible values = ENABLED, DISABLED
sslredirect	<String>	Read-write	State of HTTPS redirects for the SSL service. For an SSL session, if the client browser receives a redirect message, the browser tries to connect to the new location. However, the secure SSL

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
			session breaks if the object has moved from a secure site (https://) to an unsecure site (http://). Typically, a warning message appears on the screen, prompting the user to continue or disconnect. If SSL Redirect is ENABLED, the redirect message is automatically converted from http:// to https:// and the SSL session does not break. This parameter is not applicable when configuring a backend profile. Default value: DISABLED Possible values = ENABLED, DISABLED
redirectportrewrite	<String>	Read-write	State of the port rewrite while performing HTTPS redirect. If this parameter is set to

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
			ENABLED, and the URL from the server does not contain the standard port, the port is rewritten to the standard. Default value: DISABLED Possible values = ENABLED, DISABLED
ssl3	<String>	Read-write	State of SSLv3 protocol support for the SSL profile. Default value: DISABLED Possible values = ENABLED, DISABLED
tls1	<String>	Read-write	State of TLSv1.0 protocol support for the SSL profile. Default value: ENABLED Possible values = ENABLED, DISABLED
tls11	<String>	Read-write	State of TLSv1.1 protocol support for the SSL profile. Default value: ENABLED

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
			Possible values = ENABLED, DISABLED
tls12	<String>	Read-write	State of TLSv1.2 protocol support for the SSL profile. Default value: ENABLED Possible values = ENABLED, DISABLED
tls13	<String>	Read-write	State of TLSv1.3 protocol support for the SSL profile. Default value: DISABLED Possible values = ENABLED, DISABLED
snienable	<String>	Read-write	State of the Server Name Indication (SNI) feature on the virtual server and service-based offload. SNI helps to enable SSL encryption on multiple domains on a single virtual server or service if the domains are controlled by the same organization and share the same second-level domain

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
			name. For example, *.sports.net can be used to secure domains such as login.sports.net and help.sports.net. Default value: DISABLED Possible values = ENABLED, DISABLED
ocspstapling	<String>	Read-write	State of OCSP stapling support on the SSL virtual server. Supported only if the protocol used is higher than SSLv3. Possible values: ENABLED: The appliance sends a request to the OCSP responder to check the status of the server certificate and caches the response for the specified time. If the response is valid at the time of SSL handshake with the client, the OCSP-based server certificate status is sent to the client

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
			during the handshake. DISABLED: The appliance does not check the status of the server certificate. . Default value: DISABLED Possible values = ENABLED, DISABLED
serverauth	<String>	Read-write	State of server authentication support for the SSL Backend profile. Default value: DISABLED Possible values = ENABLED, DISABLED
commonname	<String>	Read-write	Name to be checked against the CommonName (CN) field in the server certificate bound to the SSL server. Minimum length = 1
pushenctrigger	<String>	Read-write	Trigger encryption on the basis of the PUSH flag value. Available settings function as follows:

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
			* ALWAYS - Any PUSH packet triggers encryption. * IGNORE - Ignore PUSH packet for triggering encryption. * MERGE - For a consecutive sequence of PUSH packets, the last PUSH packet triggers encryption. * TIMER - PUSH packet triggering encryption is delayed by the time defined in the set ssl parameter command or in the Change Advanced SSL Settings dialog box. Possible values = Always, Merge, Ignore, Timer
sendclosenotify	<String>	Read-write	Enable sending SSL Close-Notify at the end of a transaction. Default value: YES Possible values = YES, NO

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
cleartextport	<Integer>	Read-write	Port on which clear-text data is sent by the appliance to the server. Do not specify this parameter for SSL offloading with end-to-end encryption. Range 1 - 65535 * in CLI is represented as 65535 in NITRO API
insertionencoding	<String>	Read-write	Encoding method used to insert the subject or issuer's name in HTTP requests to servers. Default value: Unicode Possible values = Unicode, UTF-8
denysslreneg	<String>	Read-write	Deny renegotiation in specified circumstances. Available settings function as follows: * NO - Allow SSL renegotiation. * FRONTEND_CLIENT - Deny secure and nonsecure SSL renegotiation

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
			initiated by the client. * FRONTEND_CLIENTSERVER - Deny secure and nonsecure SSL renegotiation initiated by the client or the Citrix ADC during policy-based client authentication. * ALL - Deny all secure and nonsecure SSL renegotiation. * NONSECURE - Deny nonsecure SSL renegotiation. Allows only clients that support RFC 5746. Default value: ALL Possible values = NO, FRONTEND_CLIENT, FRONTEND_CLIENTSERVER, ALL, NONSECURE

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
quantumsize	<String>	Read-write	Amount of data to collect before the data is pushed to the crypto hardware for encryption. For large downloads, a larger quantum size better utilizes the crypto resources. Default value: 8192 Possible values = 4096, 8192, 16384
strictcachecks	<String>	Read-write	Enable strict CA certificate checks on the appliance. Default value: NO Possible values = YES, NO
encrypttriggerpktcount	<Double>	Read-write	Maximum number of queued packets after which encryption is triggered. Use this setting for SSL transactions that send small packets from server to Citrix ADC. Default value: 45 Minimum value = 10 Maximum value = 50
pushflag	<Double>	Read-write	Insert PUSH flag into decrypted, encrypted, or all

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
			records. If the PUSH flag is set to a value other than 0, the buffered records are forwarded on the basis of the value of the PUSH flag. Available settings function as follows: 0 - Auto (PUSH flag is not set.) 1 - Insert PUSH flag into every decrypted record. 2 -Insert PUSH flag into every encrypted record. 3 - Insert PUSH flag into every decrypted and encrypted record. Minimum value = 0 Maximum value = 3
dropreqwithnohostheader	<String>	Read-write	Host header check for SNI enabled sessions. If this check is enabled and the HTTP request does not contain the host header for SNI enabled sessions, the request is dropped. Also, please note that when the check is

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
			disabled, the request will not be dropped even if there is a mis-match between SNI and HTTP host-header. Default value: NO Possible values = YES, NO
pushenctriggertimeout	<Double>	Read-write	PUSH encryption trigger timeout value. The timeout value is applied only if you set the Push Encryption Trigger parameter to Timer in the SSL virtual server settings. Default value: 1 Minimum value = 1 Maximum value = 200
ssltriggertimeout	<Double>	Read-write	Time, in milliseconds, after which encryption is triggered for transactions that are not tracked on the Citrix ADC because their length is not known. There can be a delay of up to 10ms from the specified timeout value before the

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
			packet is pushed into the queue. Default value: 100 Minimum value = 1 Maximum value = 200
clientauthuseboundcachain	<String>	Read-write	Certficates bound on the VIP are used for validating the client cert. Certficates came along with client cert are not used for validating the client cert. Default value: DISABLED Possible values = ENABLED, DISABLED
sslinterception	<String>	Read-write	Enable or disable transparent interception of SSL sessions. Default value: DISABLED Possible values = ENABLED, DISABLED
sslireneg	<String>	Read-write	Enable or disable triggering the client renegotiation when renegotiation request

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
			is received from the origin server. Default value: ENABLED Possible values = ENABLED, DISABLED
ssliocspcheck	<String>	Read-write	Enable or disable OCSP check for origin server certificate. Default value: ENABLED Possible values = ENABLED, DISABLED
sslimaxsessperserver	<Double>	Read-write	Maximum ssl session to be cached per dynamic origin server. A unique ssl session is created for each SNI received from the client on ClientHello and the matching session is used for server session reuse. Default value: 10 Minimum value = 1 Maximum value = 1000
sessionticket	<String>	Read-write	This option enables the use of session

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
			tickets, as per the RFC 5077. Default value: DISABLED Possible values = ENABLED, DISABLED
sessionticketlifetime	<Double>	Read-write	This option sets the life time of session tickets issued by NS in secs. Default value: 300 Minimum value = 0 Maximum value = 172800
sessionticketkeyrefresh	<String>	Read-write	This option enables the use of session tickets, as per the RFC 5077. Default value: ENABLED Possible values = ENABLED, DISABLED
sessionticketkeydata	<String>	Read-write	Session ticket enc/dec key , admin can set it.
sessionkeylifetime	<Double>	Read-write	This option sets the life time of symm key used to generate session tickets

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
			issued by NS in secs. Default value: 3000 Minimum value = 600 Maximum value = 86400
prevsessionkeylifetime	<Double>	Read-write	This option sets the life time of symm key used to generate session tickets issued by NS in secs. Default value: 0 Minimum value = 0 Maximum value = 172800
hsts	<String>	Read-write	State of HSTS protocol support for the SSL profile. Using HSTS, a server can enforce the use of an HTTPS connection for all communication with a client. Default value: DISABLED Possible values = ENABLED, DISABLED
maxage	<Double>	Read-write	Set the maximum time, in seconds, in

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
			the strict transport security (STS) header during which the client must send only HTTPS requests to the server. Default value: 0 Minimum value = 0 Maximum value = 4294967294
includesubdomains	<String>	Read-write	Enable HSTS for subdomains. If set to Yes, a client must send only HTTPS requests for subdomains. Default value: NO Possible values = YES, NO
preload	<String>	Read-write	Flag indicates the consent of the site owner to have their domain preloaded. Default value: NO Possible values = YES, NO
skipclientcertpolicycheck	<String>	Read-write	This flag controls the processing of X509 certificate policies. If this option is Enabled, then the policy check in Client authentication will be

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
			skipped. This option can be used only when Client Authentication is Enabled and ClientCert is set to Mandatory. Default value: DISABLED Possible values = ENABLED, DISABLED
zerorttearlydata	<String>	Read-write	State of TLS 1.3 0-RTT early data support for the SSL Virtual Server. This setting only has an effect if resumption is enabled, as early data cannot be sent along with an initial handshake. Early application data has significantly different security properties - in particular there is no guarantee that the data cannot be replayed. Default value: DISABLED

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
			Possible values = ENABLED, DISABLED
tls13sessionticketsperauthcontext	<Double>	Read-write	Number of tickets the SSL Virtual Server will issue anytime TLS 1.3 is negotiated, ticket-based resumption is enabled, and either (1) a handshake completes or (2) post-handhsake client auth completes. This value can be increased to enable clients to open multiple parallel connections using a fresh ticket for each connection. No tickets are sent if resumption is disabled. Default value: 1 Minimum value = 1 Maximum value = 10
dhekeyexchangewithpsk	<String>	Read-write	Whether or not the SSL Virtual Server will require a DHE key exchange to occur when a PSK is

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
			accepted during a TLS 1.3 resumption handshake. A DHE key exchange ensures forward secrecy even in the event that ticket keys are compromised, at the expense of an additional round trip and resources required to carry out the DHE key exchange. If disabled, a DHE key exchange will be performed when a PSK is accepted but only if requested by the client. If enabled, the server will require a DHE key exchange when a PSK is accepted regardless of whether the client supports combined PSK-DHE key exchange. This setting only has an effect when resumption is enabled.

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
			Default value: NO Possible values = YES, NO
ciphername	<String>	Read-write	The cipher group/ alias/individual cipher configuration.
cipherpriority	<Double>	Read-write	cipher priority. Minimum value = 1
strictsigdigestcheck	<String>	Read-write	Parameter indicating to check whether peer entity certificate during TLS1.2 handshake is signed with one of signature-hash combination supported by Citrix ADC. Default value: DISABLED Possible values = ENABLED, DISABLED
nonfipsciphers	<String>	Read-only	State of usage of ciphers that are not FIPS approved. Valid only for an SSL service bound with a FIPS key and certificate. Default value: DISABLED

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
			Possible values = ENABLED, DISABLED
crlcheck	<String>	Read-only	The state of the CRL check parameter. (Mandatory/ Optional). Possible values = Mandatory, Optional
ocspcheck	<String>	Read-only	The state of the OCSP check parameter. (Mandatory/ Optional). Possible values = Mandatory, Optional
snicert	<Boolean>	Read-only	The name of the CertKey. Use this option to bind Certkey(s) which will be used in SNI processing.
skipcaname	<Boolean>	Read-only	The flag is used to indicate whether this particular CA certificate's CA_Name needs to be sent to the SSL client while requesting for client certificate in a SSL handshake.

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
invoke	<Boolean>	Read-only	Invoke flag. This attribute is relevant only for ADVANCED policies.
labeltype	<String>	Read-only	Type of policy label invocation. Possible values = vserver, service, policylabel
service	<Double>	Read-only	Service.
sslpfobjecttype	<Double>	Read-only	Internal flag to indicate what type of object binds this profile: monitor or service.
ssliverifyservercertforreuse	<String>	Read-only	Verify the origin server's certificate before reusing the front-end SSL session. Making it hidden since we always verify for now. Default value: ENABLED Possible values = ENABLED, DISABLED
__count	<Double>	Read-only	count parameter

Operations

(click to see Properties)

ADD| DELETE| UPDATE| UNSET| GET (ALL)| GET| COUNT

Some options that you can use for each operations:

- **Getting warnings in response:**NITRO allows you to get warnings in an operation by specifying the "warning" query parameter as "yes". For example, to get warnings while connecting to the Citrix ADC appliance, the URL is as follows:

  http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/login?warning=yes

  If any, the warnings are displayed in the response payload with the HTTP code "209 X-NITRO-WARNING".

- **Authenticated access for individual NITRO operations:**NITRO allows you to logon to the Citrix ADC appliance to perform individual operations. You can use this option instead of creating a NITRO session (using the login object) and then using that session to perform all operations,

  To do this, you must specify the username and password in the request header of the NITRO request as follows:

  X-NITRO-USER:<username>

  X-NITRO-PASS:<password>

  Note: In such cases, make sure that the request header DOES not include the following:

  Cookie:NITRO_AUTH_TOKEN=<tokenvalue>

`*Note:`*

Mandatory parameters are marked in redand placeholder content is marked in <green>.

add

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/sslprofile

HTTP Method:POST

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Content-Type:application/json

Request Payload:

```
{"sslprofile":{

<b>"name":<String_value>,

</b>"sslprofiletype":<String_value>,

"ssllogprofile":<String_value>,

"dhcount":<Double_value>,

"dh":<String_value>,

"dhfile":<String_value>,

"ersa":<String_value>,

"ersacount":<Double_value>,

"sessreuse":<String_value>,

"sesstimeout":<Double_value>,

"cipherredirect":<String_value>,

"cipherurl":<String_value>,

"clientauth":<String_value>,

"clientcert":<String_value>,

"dhkeyexpsizelimit":<String_value>,

"sslredirect":<String_value>,

"redirectportrewrite":<String_value>,

"ssl3":<String_value>,

"tls1":<String_value>,

"tls11":<String_value>,
```

```
"tls12":<String_value>,

"tls13":<String_value>,

"snienable":<String_value>,

"ocspstapling":<String_value>,

"serverauth":<String_value>,

"commonname":<String_value>,

"pushenctrigger":<String_value>,

"sendclosenotify":<String_value>,

"cleartextport":<Integer_value>,

"insertionencoding":<String_value>,

"denysslreneg":<String_value>,

"quantumsize":<String_value>,

"strictcachecks":<String_value>,

"encrypttriggerpktcount":<Double_value>,

"pushflag":<Double_value>,

"dropreqwithnohostheader":<String_value>,

"pushenctriggertimeout":<Double_value>,

"ssltriggertimeout":<Double_value>,

"clientauthuseboundcachain":<String_value>,

"sslinterception":<String_value>,

"sslireneg":<String_value>,

"ssliocspcheck":<String_value>,
```

```
"sslimaxsessperserver":<Double_value>,

"sessionticket":<String_value>,

"sessionticketlifetime":<Double_value>,

"sessionticketkeyrefresh":<String_value>,

"sessionticketkeydata":<String_value>,

"sessionkeylifetime":<Double_value>,

"prevsessionkeylifetime":<Double_value>,

"hsts":<String_value>,

"maxage":<Double_value>,

"includesubdomains":<String_value>,

"preload":<String_value>,

"skipclientcertpolicycheck":<String_value>,

"zerorttearlydata":<String_value>,

"tls13sessionticketsperauthcontext":<Double_value>,

"dhekeyexchangewithpsk":<String_value>

}}
```

Response:

HTTP Status Code on Success: 201 Created HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error

delete

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/sslprofile/name_value<String>

HTTP Method:DELETE

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue>

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error

update

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/sslprofile

HTTP Method:PUT

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Content-Type:application/json

Request Payload:

```
{"sslprofile":{

<b>"name":<String_value>,

</b>"ssllogprofile":<String_value>,

"dh":<String_value>,

"dhfile":<String_value>,

"dhcount":<Double_value>,

"dhkeyexpsizelimit":<String_value>,

"ersa":<String_value>,

"ersacount":<Double_value>,

"sessreuse":<String_value>,
```

```
"sesstimeout":<Double_value>,

"cipherredirect":<String_value>,

"cipherurl":<String_value>,

"clientauth":<String_value>,

"clientcert":<String_value>,

"sslredirect":<String_value>,

"redirectportrewrite":<String_value>,

"ssl3":<String_value>,

"tls1":<String_value>,

"tls11":<String_value>,

"tls12":<String_value>,

"tls13":<String_value>,

"snienable":<String_value>,

"ocspstapling":<String_value>,

"serverauth":<String_value>,

"commonname":<String_value>,

"pushenctrigger":<String_value>,

"sendclosenotify":<String_value>,

"cleartextport":<Integer_value>,

"insertionencoding":<String_value>,

"denysslreneg":<String_value>,

"quantumsize":<String_value>,
```

```
"strictcachecks":<String_value>,

"encrypttriggerpktcount":<Double_value>,

"pushflag":<Double_value>,

"dropreqwithnohostheader":<String_value>,

"pushenctriggertimeout":<Double_value>,

"ssltriggertimeout":<Double_value>,

"clientauthuseboundcachain":<String_value>,

"sslinterception":<String_value>,

"sslireneg":<String_value>,

"ssliocspcheck":<String_value>,

"sslimaxsessperserver":<Double_value>,

"hsts":<String_value>,

"maxage":<Double_value>,

"includesubdomains":<String_value>,

"preload":<String_value>,

"sessionticket":<String_value>,

"sessionticketlifetime":<Double_value>,

"sessionticketkeyrefresh":<String_value>,

"sessionticketkeydata":<String_value>,

"sessionkeylifetime":<Double_value>,

"prevsessionkeylifetime":<Double_value>,

"ciphername":<String_value>,
```

```
"cipherpriority":<Double_value>,

"strictsigdigestcheck":<String_value>,

"skipclientcertpolicycheck":<String_value>,

"zerorttearlydata":<String_value>,

"tls13sessionticketsperauthcontext":<Double_value>,

"dhekeyexchangewithpsk":<String_value>

}}
```

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error

unset

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/sslprofile?**action=unset**

HTTP Method:POST

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Content-Type:application/json

Request Payload:

```
{"sslprofile":{

<b>"name":<String_value>,

</b>"ssllogprofile":true,

"dh":true,
```

```
"dhfile":true,

"dhcount":true,

"dhkeyexpsizelimit":true,

"ersa":true,

"ersacount":true,

"sessreuse":true,

"sesstimeout":true,

"cipherredirect":true,

"cipherurl":true,

"clientauth":true,

"clientcert":true,

"sslredirect":true,

"redirectportrewrite":true,

"ssl3":true,

"tls1":true,

"tls11":true,

"tls12":true,

"tls13":true,

"snienable":true,

"ocspstapling":true,

"serverauth":true,

"commonname":true,
```

```
"pushenctrigger":true,

"sendclosenotify":true,

"cleartextport":true,

"insertionencoding":true,

"denysslreneg":true,

"quantumsize":true,

"strictcachecks":true,

"encrypttriggerpktcount":true,

"pushflag":true,

"dropreqwithnohostheader":true,

"pushenctriggertimeout":true,

"ssltriggertimeout":true,

"clientauthuseboundcachain":true,

"sslinterception":true,

"sslireneg":true,

"ssliocspcheck":true,

"sslimaxsessperserver":true,

"hsts":true,

"maxage":true,

"includesubdomains":true,

"preload":true,

"sessionticket":true,
```

```
"sessionticketlifetime":true,

"sessionticketkeyrefresh":true,

"sessionticketkeydata":true,

"sessionkeylifetime":true,

"prevsessionkeylifetime":true,

"ciphername":true,

"cipherpriority":true,

"strictsigdigestcheck":true,

"skipclientcertpolicycheck":true,

"zerorttearlydata":true,

"tls13sessionticketsperauthcontext":true,

"dhekeyexchangewithpsk":true

}}
```

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error

get (all)

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/sslprofile

Query-parameters:

attrs

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/sslprofile?**attrs=property-name1,property-name2**

Use this query parameter to specify the resource details that you want to retrieve.

filter

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/sslprofile?**filter=property-name1:property-val1,property-name2:property-val2**

Use this query-parameter to get the filtered set of sslprofile resources configured on Citrix ADC. Filtering can be done on any of the properties of the resource.

view

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/sslprofile?**view=summary**

Note: By default, the retrieved results are displayed in detail view (?view=detail).

pagination

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/sslprofile?**pagesize=#no;pageno=#no**

Use this query-parameter to get the sslprofile resources in chunks.

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "sslprofile": [ {

"name":<String_value>,

"dh":<String_value>,
```

```
"dhfile":<String_value>,

"dhcount":<Double_value>,

"dhkeyexpsizelimit":<String_value>,

"ersa":<String_value>,

"ersacount":<Double_value>,

"sessreuse":<String_value>,

"sesstimeout":<Double_value>,

"cipherredirect":<String_value>,

"cipherurl":<String_value>,

"clientauth":<String_value>,

"clientcert":<String_value>,

"sslredirect":<String_value>,

"redirectportrewrite":<String_value>,

"nonfipsciphers":<String_value>,

"ssl3":<String_value>,

"tls1":<String_value>,

"tls11":<String_value>,

"tls12":<String_value>,

"tls13":<String_value>,

"snienable":<String_value>,

"ocspstapling":<String_value>,

"serverauth":<String_value>,
```

```
"commonname":<String_value>,

"pushenctrigger":<String_value>,

"sendclosenotify":<String_value>,

"cleartextport":<Integer_value>,

"insertionencoding":<String_value>,

"denysslreneg":<String_value>,

"quantumsize":<String_value>,

"strictcachecks":<String_value>,

"encrypttriggerpktcount":<Double_value>,

"pushflag":<Double_value>,

"dropreqwithnohostheader":<String_value>,

"pushenctriggertimeout":<Double_value>,

"ssltriggertimeout":<Double_value>,

"cipherpriority":<Double_value>,

"ciphername":<String_value>,

"crlcheck":<String_value>,

"ocspcheck":<String_value>,

"snicert":<Boolean_value>,

"skipcaname":<Boolean_value>,

"invoke":<Boolean_value>,

"labeltype":<String_value>,

"sslprofiletype":<String_value>,
```

```
"clientauthuseboundcachain":<String_value>,

"ssllogprofile":<String_value>,

"service":<Double_value>,

"sslpfobjecttype":<Double_value>,

"sslinterception":<String_value>,

"ssliverifyservercertforreuse":<String_value>,

"sslireneg":<String_value>,

"ssliocspcheck":<String_value>,

"sslimaxsessperserver":<Double_value>,

"sessionticket":<String_value>,

"sessionticketlifetime":<Double_value>,

"sessionticketkeyrefresh":<String_value>,

"sessionticketkeydata":<String_value>,

"sessionkeylifetime":<Double_value>,

"prevsessionkeylifetime":<Double_value>,

"hsts":<String_value>,

"maxage":<Double_value>,

"includesubdomains":<String_value>,

"preload":<String_value>,

"strictsigdigestcheck":<String_value>,

"skipclientcertpolicycheck":<String_value>,

"zerorttearlydata":<String_value>,
```

```
"tls13sessionticketsperauthcontext":<Double_value>,

"dhekeyexchangewithpsk":<String_value>

}]}
```

get

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/sslprofile/name_value<String>

Query-parameters:

attrs

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/sslprofile/name_value<String>?**attrs=property-name1,property-name2**

Use this query parameter to specify the resource details that you want to retrieve.

view

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/sslprofile/name_value<String>?**view=summary**

Note: By default, the retrieved results are displayed in detail view (?view=detail).

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "sslprofile": [ {

"name":<String_value>,

"dh":<String_value>,

"dhfile":<String_value>,

"dhcount":<Double_value>,

"dhkeyexpsizelimit":<String_value>,

"ersa":<String_value>,

"ersacount":<Double_value>,

"sessreuse":<String_value>,

"sesstimeout":<Double_value>,

"cipherredirect":<String_value>,

"cipherurl":<String_value>,

"clientauth":<String_value>,

"clientcert":<String_value>,

"sslredirect":<String_value>,

"redirectportrewrite":<String_value>,

"nonfipsciphers":<String_value>,

"ssl3":<String_value>,

"tls1":<String_value>,

"tls11":<String_value>,

"tls12":<String_value>,
```

```
"tls13":<String_value>,

"snienable":<String_value>,

"ocspstapling":<String_value>,

"serverauth":<String_value>,

"commonname":<String_value>,

"pushenctrigger":<String_value>,

"sendclosenotify":<String_value>,

"cleartextport":<Integer_value>,

"insertionencoding":<String_value>,

"denysslreneg":<String_value>,

"quantumsize":<String_value>,

"strictcachecks":<String_value>,

"encrypttriggerpktcount":<Double_value>,

"pushflag":<Double_value>,

"dropreqwithnohostheader":<String_value>,

"pushenctriggertimeout":<Double_value>,

"ssltriggertimeout":<Double_value>,

"cipherpriority":<Double_value>,

"ciphername":<String_value>,

"crlcheck":<String_value>,

"ocspcheck":<String_value>,

"snicert":<Boolean_value>,
```

```
"skipcaname":<Boolean_value>,

"invoke":<Boolean_value>,

"labeltype":<String_value>,

"sslprofiletype":<String_value>,

"clientauthuseboundcachain":<String_value>,

"ssllogprofile":<String_value>,

"service":<Double_value>,

"sslpfobjecttype":<Double_value>,

"sslinterception":<String_value>,

"ssliverifyservercertforreuse":<String_value>,

"sslireneg":<String_value>,

"ssliocspcheck":<String_value>,

"sslimaxsessperserver":<Double_value>,

"sessionticket":<String_value>,

"sessionticketlifetime":<Double_value>,

"sessionticketkeyrefresh":<String_value>,

"sessionticketkeydata":<String_value>,

"sessionkeylifetime":<Double_value>,

"prevsessionkeylifetime":<Double_value>,

"hsts":<String_value>,

"maxage":<Double_value>,

"includesubdomains":<String_value>,
```

```
"preload":<String_value>,

"strictsigdigestcheck":<String_value>,

"skipclientcertpolicycheck":<String_value>,

"zerorttearlydata":<String_value>,

"tls13sessionticketsperauthcontext":<Double_value>,

"dhekeyexchangewithpsk":<String_value>

}]}
```

count

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/sslprofile?**count=yes**

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "sslprofile": [ { "__count": "#no"} ] }
```

sslprofile_binding

Binding object which returns the resources bound to sslprofile.

Properties

(click to see Operations)

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
name	<String>	Read-write	Name of the SSL profile for which to show detailed information. Minimum length = 1 Maximum length = 127
sslprofile_sslcipher_binding	<sslprofile_sslcipher_binding[]>	Read-only	sslcipher that can be bound to sslprofile.
sslprofile_ecccurve_binding	<sslprofile_ecccurve_binding[]>	Read-only	ecccurve that can be bound to sslprofile.
sslprofile_sslvserver_binding	<sslprofile_sslvserver_binding[]>	Read-only	sslvserver that can be bound to sslprofile.
sslprofile_sslciphersuite_binding	<sslprofile_sslciphersuite_binding[]>	Read-only	sslciphersuite that can be bound to sslprofile.
sslprofile_sslcertkey_binding	<sslprofile_sslcertkey_binding[]>	Read-only	sslcertkey that can be bound to sslprofile.

Operations

(click to see Properties)

GET| GET (ALL)

Some options that you can use for each operations:

- **Getting warnings in response:**NITRO allows you to get warnings in an operation by specifying the "warning" query parameter as "yes". For example, to get warnings while connecting to the Citrix ADC appliance, the URL is as follows:

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/login?warning=yes

If any, the warnings are displayed in the response payload with the HTTP code "209 X-NITRO-WARNING".

- **Authenticated access for individual NITRO operations:**NITRO allows you to logon to the Citrix ADC appliance to perform individual operations. You can use this option instead of creating a NITRO session (using the login object) and then using that session to perform all operations,

  To do this, you must specify the username and password in the request header of the NITRO request as follows:

  X-NITRO-USER:<username>

  X-NITRO-PASS:<password>

  Note: In such cases, make sure that the request header DOES not include the following:

  Cookie:NITRO_AUTH_TOKEN=<tokenvalue>

`*Note:`*

Mandatory parameters are marked in redand placeholder content is marked in <green>.

get

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/sslprofile_binding/name_value<String>

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "sslprofile_binding": [ {

"name":<String_value>,

"sslprofile_ecccurve_binding":<sslprofile_ecccurve_binding[]_value>,

"sslprofile_sslcertkey_binding":<sslprofile_sslcertkey_binding[]_value>,

"sslprofile_sslciphersuite_binding":<sslprofile_sslciphersuite_binding[]_value>,

"sslprofile_sslvserver_binding":<sslprofile_sslvserver_binding[]_value>,

"sslprofile_sslcipher_binding":<sslprofile_sslcipher_binding[]_value>

}]}
```

get (all)

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/sslprofile_binding

Query-parameters:

bulkbindings

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/sslprofile_binding?**bulkbindings=yes**

NITRO allows you to fetch bindings in bulk.

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "sslprofile_binding": [ {

"name":<String_value>,

"sslprofile_ecccurve_binding":<sslprofile_ecccurve_binding[]_value>,

"sslprofile_sslcertkey_binding":<sslprofile_sslcertkey_binding[]_value>,

"sslprofile_sslciphersuite_binding":<sslprofile_sslciphersuite_binding[]_value>,

"sslprofile_sslvserver_binding":<sslprofile_sslvserver_binding[]_value>,

"sslprofile_sslcipher_binding":<sslprofile_sslcipher_binding[]_value>

}]}
```

sslprofile_ecccurve_binding

Binding object showing the ecccurve that can be bound to sslprofile.

Properties

(click to see Operations)

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
cipherpriority	<Double>	Read-write	Priority of the cipher binding. Minimum value = 1 Maximum value = 1000
name	<String>	Read-write	Name of the SSL profile.

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
			Minimum length = 1 Maximum length = 127
ecccurvename	<String>	Read-write	Named ECC curve bound to vserver/ service. Possible values = ALL, P_224, P_256, P_384, P_521
__count	<Double>	Read-write	count parameter

Operations

(click to see Properties)

ADD:| DELETE:| GET| GET (ALL)| COUNT

Some options that you can use for each operations:

- **Getting warnings in response:**NITRO allows you to get warnings in an operation by specifying the "warning" query parameter as "yes". For example, to get warnings while connecting to the Citrix ADC appliance, the URL is as follows:

  http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/login?warning=yes

  If any, the warnings are displayed in the response payload with the HTTP code "209 X-NITRO-WARNING".

- **Authenticated access for individual NITRO operations:**NITRO allows you to logon to the Citrix ADC appliance to perform individual operations. You can use this option instead of creating a NITRO session (using the login object) and then using that session to perform all operations,

  To do this, you must specify the username and password in the request header of the NITRO request as follows:

  X-NITRO-USER:<username>

  X-NITRO-PASS:<password>

Note: In such cases, make sure that the request header DOES not include the following:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue>

`*Note:` *

Mandatory parameters are marked in redand placeholder content is marked in <green>.

add

URL:http://<Citrix-ADC-IP-address(NSIP)/nitro/v1/config/sslprofile_ecccurve_binding

HTTP Method:PUT

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Content-Type:application/json

Request Payload:

```
{

"sslprofile_ecccurve_binding":{

<b>"name":<String_value>,

</b>"cipherpriority":<Double_value>,

"ecccurvename":<String_value>

}}
```

Response:

HTTP Status Code on Success: 201 Created HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error

delete

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/sslprofile_ecccurve_binding/name_value<String>

Query-parameters:

args

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/sslprofile_ecccurve_binding/name_value<String>?**args=ecccurvename:<String_value>**

HTTP Method:DELETE

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue>

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error

get

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/sslprofile_ecccurve_binding/name_value<String>

Query-parameters:

filter

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/sslprofile_ecccurve_binding/name_value<String>?**filter=property-name1:property-value1,property-name2:property-value2**

Use this query-parameter to get the filtered set of sslprofile_ecccurve_binding resources configured on Citrix ADC. Filtering can be done on any of the properties of the resource.

pagination

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/sslprofile_ecccurve_binding/name_value<String>?**pagesize=#no;pageno=#no**

Use this query-parameter to get the sslprofile_ecccurve_binding resources in chunks.

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "sslprofile_ecccurve_binding": [ {

"cipherpriority":<Double_value>,

"name":<String_value>,

"ecccurvename":<String_value>

}]}
```

get (all)

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/sslprofile_ecccurve_binding

Query-parameters:

bulkbindings

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/sslprofile_ecccurve_binding?**bulkbindings=yes**

NITRO allows you to fetch bindings in bulk.

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "sslprofile_ecccurve_binding": [ {

"cipherpriority":<Double_value>,

"name":<String_value>,

"ecccurvename":<String_value>

}]}
```

count

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/sslprofile_ecccurve_binding/name_value<String>?**count=yes**

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{"sslprofile_ecccurve_binding": [ { "__count": "#no"} ] }
```

sslprofile_sslcertkey_binding

Binding object showing the sslcertkey that can be bound to sslprofile.

Properties

(click to see Operations)

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
cipherpriority	<Double>	Read-write	Priority of the cipher binding. Minimum value = 1 Maximum value = 1000
sslicacertkey	<String>	Read-write	The certkey (CA certificate + private key) to be used for SSL interception.
name	<String>	Read-write	Name of the SSL profile. Minimum length = 1 Maximum length = 127
__count	<Double>	Read-write	count parameter

Operations

(click to see Properties)

ADD:| DELETE:| GET| GET (ALL)| COUNT

Some options that you can use for each operations:

- **Getting warnings in response:**NITRO allows you to get warnings in an operation by specifying the "warning" query parameter as "yes". For example, to get warnings while connecting to the Citrix ADC appliance, the URL is as follows:

  http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/login?warning=yes

  If any, the warnings are displayed in the response payload with the HTTP code "209 X-NITRO-WARNING".

- **Authenticated access for individual NITRO operations:**NITRO allows you to logon to the Citrix ADC appliance to perform individual operations. You can use this option instead of creating a NITRO session (using the login object) and then using that session to perform all operations,

  To do this, you must specify the username and password in the request header of the NITRO request as follows:

  X-NITRO-USER:<username>

  X-NITRO-PASS:<password>

  Note: In such cases, make sure that the request header DOES not include the following:

  Cookie:NITRO_AUTH_TOKEN=<tokenvalue>

`*Note:`*

Mandatory parameters are marked in redand placeholder content is marked in <green>.

add

URL:http://<Citrix-ADC-IP-address(NSIP)/nitro/v1/config/sslprofile_sslcertkey_binding

HTTP Method:PUT

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Content-Type:application/json

Request Payload:

```
{
```

```
"sslprofile_sslcertkey_binding":{

<b>"name":<String_value>,

</b>"cipherpriority":<Double_value>,

"sslicacertkey":<String_value>

}}
```

Response:

HTTP Status Code on Success: 201 Created HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error

delete

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/sslprofile_sslcertkey_binding/name_value<String>

Query-parameters:

args

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/sslprofile_sslcertkey_binding/name_value<String>?**args=sslicacertkey:<String_value>**

HTTP Method:DELETE

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue>

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error

get

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/sslprofile_sslcertkey_binding/name_value<String>

Query-parameters:

filter

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/sslprofile_sslcertkey_binding/name_value<String>?**filter=property-name1:property-value1,property-name2:property-value2**

Use this query-parameter to get the filtered set of sslprofile_sslcertkey_binding resources configured on Citrix ADC. Filtering can be done on any of the properties of the resource.

pagination

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/sslprofile_sslcertkey_binding/name_value<String>?**pagesize=#no;pageno=#no**

Use this query-parameter to get the sslprofile_sslcertkey_binding resources in chunks.

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "sslprofile_sslcertkey_binding": [ {

"cipherpriority":<Double_value>,

"sslicacertkey":<String_value>,
```

```
"name":<String_value>

}]}
```

get (all)

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/sslprofile_sslcertkey_binding

Query-parameters:

bulkbindings

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/sslprofile_sslcertkey_binding?**bulkbindings=yes**

NITRO allows you to fetch bindings in bulk.

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "sslprofile_sslcertkey_binding": [ {

"cipherpriority":<Double_value>,

"sslicacertkey":<String_value>,

"name":<String_value>
```

```
}]}
```

count

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/sslprofile_sslcertkey_binding/name_value<String>?**count=yes**

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{"sslprofile_sslcertkey_binding": [ { "__count": "#no"} ] }
```

sslprofile_sslciphersuite_binding

Binding object showing the sslciphersuite that can be bound to sslprofile.

Properties

(click to see Operations)

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
ciphername	<String>	Read-write	The cipher group/alias/individual cipher configuration.

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
cipherpriority	<Double>	Read-write	cipher priority. Minimum value = 1
name	<String>	Read-write	Name of the SSL profile. Minimum length = 1 Maximum length = 127
description	<String>	Read-write	The cipher suite description.
__count	<Double>	Read-write	count parameter

Operations

(click to see Properties)

ADD:| DELETE:| GET| GET (ALL)| COUNT

Some options that you can use for each operations:

- **Getting warnings in response:**NITRO allows you to get warnings in an operation by specifying the "warning" query parameter as "yes". For example, to get warnings while connecting to the Citrix ADC appliance, the URL is as follows:

  http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/login?warning=yes

  If any, the warnings are displayed in the response payload with the HTTP code "209 X-NITRO-WARNING".

- **Authenticated access for individual NITRO operations:**NITRO allows you to logon to the Citrix ADC appliance to perform individual operations. You can use this option instead of creating a NITRO session (using the login object) and then using that session to perform all operations,

  To do this, you must specify the username and password in the request header of the NITRO request as follows:

  X-NITRO-USER:<username>

X-NITRO-PASS:<password>

Note: In such cases, make sure that the request header DOES not include the following:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue>

`*Note:`*

Mandatory parameters are marked in redand placeholder content is marked in <green>.

add

URL:http://<Citrix-ADC-IP-address(NSIP)/nitro/v1/config/sslprofile_sslciphersuite_binding

HTTP Method:PUT

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Content-Type:application/json

Request Payload:

```
{
"sslprofile_sslciphersuite_binding":{
<b>"name":<String_value>,
</b>"ciphername":<String_value>,
"cipherpriority":<Double_value>
}}
```

Response:

HTTP Status Code on Success: 201 Created HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error

delete

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/sslprofile_sslciphersuite_binding/name_value<String>

Query-parameters:

args

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/sslprofile_sslciphersuite_binding/name_value<String>?**args=ciphername:<String_value>**

HTTP Method:DELETE

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue>

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error

get

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/sslprofile_sslciphersuite_binding/name_value<String>

Query-parameters:

filter

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/sslprofile_sslciphersuite_binding/name_value<String>?**filter=property-name1:property-value1,property-name2:property-value2**

Use this query-parameter to get the filtered set of sslprofile_sslciphersuite_binding resources configured on Citrix ADC. Filtering can be done on any of the properties of the resource.

pagination

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/sslprofile_sslciphersuite_binding/name_value<String>?**pagesize=#no;pageno=#no**

Use this query-parameter to get the sslprofile_sslciphersuite_binding resources in chunks.

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "sslprofile_sslciphersuite_binding": [ {

"ciphername":<String_value>,

"cipherpriority":<Double_value>,

"name":<String_value>,

"description":<String_value>

}]}
```

get (all)

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/sslprofile_sslciphersuite_binding

Query-parameters:

bulkbindings

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/sslprofile_sslciphersuite_binding?**bulkbindings=yes**

NITRO allows you to fetch bindings in bulk.

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "sslprofile_sslciphersuite_binding": [ {

"ciphername":<String_value>,

"cipherpriority":<Double_value>,

"name":<String_value>,

"description":<String_value>

}]}
```

count

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/sslprofile_sslciphersuite_binding/name_value<String>?**count=yes**

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{"sslprofile_sslciphersuite_binding": [ { "__count": "#no"} ] }
```

sslprofile_sslcipher_binding

Binding object showing the sslcipher that can be bound to sslprofile.

Properties

(click to see Operations)

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
ciphername	<String>	Read-write	Name of the cipher. Minimum length = 1
cipheraliasname	<String>	Read-write	The name of the cipher group/alias/ individual cipheri bindings.
cipherpriority	<Double>	Read-write	cipher priority. Minimum value = 1
name	<String>	Read-write	Name of the SSL profile. Minimum length = 1 Maximum length = 127

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
description	<String>	Read-write	The cipher suite description.
__count	<Double>	Read-write	count parameter

Operations

(click to see Properties)

ADD:| DELETE:| GET| GET (ALL)| COUNT

Some options that you can use for each operations:

- **Getting warnings in response:**NITRO allows you to get warnings in an operation by specifying the "warning" query parameter as "yes". For example, to get warnings while connecting to the Citrix ADC appliance, the URL is as follows:

  http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/login?warning=yes

  If any, the warnings are displayed in the response payload with the HTTP code "209 X-NITRO-WARNING".

- **Authenticated access for individual NITRO operations:**NITRO allows you to logon to the Citrix ADC appliance to perform individual operations. You can use this option instead of creating a NITRO session (using the login object) and then using that session to perform all operations,

  To do this, you must specify the username and password in the request header of the NITRO request as follows:

  X-NITRO-USER:<username>

  X-NITRO-PASS:<password>

  Note: In such cases, make sure that the request header DOES not include the following:

  Cookie:NITRO_AUTH_TOKEN=<tokenvalue>

`*Note:`*

Mandatory parameters are marked in redand placeholder content is marked in <green>.

add

URL:http://<Citrix-ADC-IP-address(NSIP)/nitro/v1/config/sslprofile_sslcipher_binding

HTTP Method:PUT

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Content-Type:application/json

Request Payload:

```
{

"sslprofile_sslcipher_binding":{

<b>"name":<String_value>,

</b>"ciphername":<String_value>,

"cipherpriority":<Double_value>

}}
```

Response:

HTTP Status Code on Success: 201 Created HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error

delete

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/sslprofile_sslcipher_binding/name_value<String>

Query-parameters:

args

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/sslprofile_sslcipher_binding/name_value<String>?**args=ciphername:<String_value>**

HTTP Method:DELETE

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue>

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error

get

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/sslprofile_sslcipher_binding/name_value<String>

Query-parameters:

filter

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/sslprofile_sslcipher_binding/name_value<String>?**filter=property-name1:property-value1,property-name2:property-value2**

Use this query-parameter to get the filtered set of sslprofile_sslcipher_binding resources configured on Citrix ADC. Filtering can be done on any of the properties of the resource.

pagination

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/sslprofile_sslcipher_binding/name_value<String>?**pagesize=#no;pageno=#no**

Use this query-parameter to get the sslprofile_sslcipher_binding resources in chunks.

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "sslprofile_sslcipher_binding": [ {

"ciphername":<String_value>,

"cipheraliasname":<String_value>,

"cipherpriority":<Double_value>,

"name":<String_value>,

"description":<String_value>

}]}
```

get (all)

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/sslprofile_sslcipher_binding

Query-parameters:

bulkbindings

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/sslprofile_sslcipher_binding?**bulkbindings=yes**

NITRO allows you to fetch bindings in bulk.

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "sslprofile_sslcipher_binding": [ {

"ciphername":<String_value>,

"cipheraliasname":<String_value>,

"cipherpriority":<Double_value>,

"name":<String_value>,

"description":<String_value>

}]}
```

count

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/sslprofile_sslcipher_binding/name_value<String>?**count=yes**

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{"sslprofile_sslcipher_binding": [ { "__count": "#no"} ] }
```

sslprofile_sslvserver_binding

Binding object showing the sslvserver that can be bound to sslprofile.

Properties

(click to see Operations)

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
cipherpriority	<Double>	Read-write	Priority of the cipher binding. Minimum value = 1 Maximum value = 1000
name	<String>	Read-write	Name of the SSL profile. Minimum length = 1 Maximum length = 127
description	<String>	Read-write	The cipher suite description.
servicename	<String>	Read-write	Service name.
__count	<Double>	Read-write	count parameter

Operations

(click to see Properties)

GET| GET (ALL)| COUNT

Some options that you can use for each operations:

- **Getting warnings in response:**NITRO allows you to get warnings in an operation by specifying the "warning" query parameter as "yes". For example, to get warnings while connecting to the Citrix ADC appliance, the URL is as follows:

  http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/login?warning=yes

  If any, the warnings are displayed in the response payload with the HTTP code "209 X-NITRO-WARNING".

- **Authenticated access for individual NITRO operations:**NITRO allows you to logon to the Citrix ADC appliance to perform individual operations. You can use this option instead of creating a NITRO session (using the login object) and then using that session to perform all operations,

  To do this, you must specify the username and password in the request header of the NITRO request as follows:

  X-NITRO-USER:<username>

  X-NITRO-PASS:<password>

  Note: In such cases, make sure that the request header DOES not include the following:

  Cookie:NITRO_AUTH_TOKEN=<tokenvalue>

`*Note:`*

Mandatory parameters are marked in redand placeholder content is marked in <green>.

get

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/sslprofile_sslvserver_binding/name_value<String>

Query-parameters:

filter

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/sslprofile_sslvserver_binding/name_value<String>?**filter=property-name1:property-value1,property-name2:property-value2**

Use this query-parameter to get the filtered set of sslprofile_sslvserver_binding resources configured on Citrix ADC. Filtering can be done on any of the properties of the resource.

pagination

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/sslprofile_sslvserver_binding/name_value<String>?**pagesize=#no;pageno=#no**

Use this query-parameter to get the sslprofile_sslvserver_binding resources in chunks.

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "sslprofile_sslvserver_binding": [ {

"cipherpriority":<Double_value>,

"name":<String_value>,

"description":<String_value>,

"servicename":<String_value>

}]}
```

get (all)

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/sslprofile_sslvserver_binding

Query-parameters:

bulkbindings

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/sslprofile_sslvserver_binding?**bulkbindings=yes**

NITRO allows you to fetch bindings in bulk.

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "sslprofile_sslvserver_binding": [ {

"cipherpriority":<Double_value>,

"name":<String_value>,

"description":<String_value>,

"servicename":<String_value>

}]}
```

count

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/sslprofile_sslvserver_binding/name_value<String>?**count=yes**

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{"sslprofile_sslvserver_binding": [ { "__count": "#no"} ] }
```

sslrsakey

Configuration for RSA key resource.

Properties

(click to see Operations)

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
keyfile	<String>	Read-write	Name for and, optionally, path to the RSA key file. /nsconfig/ssl/ is the default path. Maximum length = 63
bits	<Double>	Read-write	Size, in bits, of the RSA key. Minimum value = 512 Maximum value = 4096

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
exponent	<String>	Read-write	Public exponent for the RSA key. The exponent is part of the cipher algorithm and is required for creating the RSA key. Default value: F4 Possible values = 3, F4
keyform	<String>	Read-write	Format in which the RSA key file is stored on the appliance. Default value: PEM Possible values = DER, PEM
des	<Boolean>	Read-write	Encrypt the generated RSA key by using the DES algorithm. On the command line, you are prompted to enter the pass phrase (password) that is used to encrypt the key.
des3	<Boolean>	Read-write	Encrypt the generated RSA key by using the Triple-DES algorithm. On the command line, you are prompted to enter the pass phrase (password)

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
			that is used to encrypt the key.
aes256	<Boolean>	Read-write	Encrypt the generated RSA key by using the AES algorithm.
password	<String>	Read-write	Pass phrase to use for encryption if DES or DES3 option is selected. Minimum length = 1 Maximum length = 31
pkcs8	<Boolean>	Read-write	Create the private key in PKCS#8 format.

Operations

(click to see Properties)

CREATE

Some options that you can use for each operations:

- **Getting warnings in response:**NITRO allows you to get warnings in an operation by specifying the "warning" query parameter as "yes". For example, to get warnings while connecting to the Citrix ADC appliance, the URL is as follows:

  http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/login?warning=yes

  If any, the warnings are displayed in the response payload with the HTTP code "209 X-NITRO-WARNING".

- **Authenticated access for individual NITRO operations:**NITRO allows you to logon to the Citrix ADC appliance to perform individual operations. You can use this option instead of

creating a NITRO session (using the login object) and then using that session to perform all operations,

To do this, you must specify the username and password in the request header of the NITRO request as follows:

X-NITRO-USER:<username>

X-NITRO-PASS:<password>

Note: In such cases, make sure that the request header DOES not include the following:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue>

`*Note:`*

Mandatory parameters are marked in redand placeholder content is marked in <green>.

create

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/sslrsakey?**action=create**

HTTP Method:POST

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Content-Type:application/json

Request Payload:

```
{"sslrsakey":{

<b>"keyfile":<String_value>,

</b><b>"bits":<Double_value>,

</b>"exponent":<String_value>,

"keyform":<String_value>,

"des":<Boolean_value>,

"des3":<Boolean_value>,
```

```
"aes256":<Boolean_value>,

"password":<String_value>,

"pkcs8":<Boolean_value>

}}
```

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error

sslservice

Configuration for SSL service resource.

Properties

(click to see Operations)

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
servicename	<String>	Read-write	Name of the SSL service. Minimum length = 1
dh	<String>	Read-write	State of Diffie-Hellman (DH) key exchange. This parameter is not applicable when configuring a backend service. Default value: DISABLED Possible values = ENABLED, DISABLED

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
dhfile	<String>	Read-write	Name for and, optionally, path to the PEM-format DH parameter file to be installed. /nsconfig/ssl/ is the default path. This parameter is not applicable when configuring a backend service. Minimum length = 1
dhcount	<Double>	Read-write	Number of interactions, between the client and the Citrix ADC, after which the DH private-public pair is regenerated. A value of zero (0) specifies infinite use (no refresh). This parameter is not applicable when configuring a backend service. Allowed DH count values are 0 and >= 500. Minimum value = 0 Maximum value = 65534
dhkeyexpsizelimit	<String>	Read-write	This option enables the use of NIST recommended (NIST Special Publication 800-56A) bit size for private-key size. For example, for DH

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
			params of size 2048bit, the private-key size recommended is 224bits. This is rounded-up to 256bits. Default value: DISABLED Possible values = ENABLED, DISABLED
ersa	<String>	Read-write	State of Ephemeral RSA (eRSA) key exchange. Ephemeral RSA allows clients that support only export ciphers to communicate with the secure server even if the server certificate does not support export clients. The ephemeral RSA key is automatically generated when you bind an export cipher to an SSL or TCP-based SSL virtual server or service. When you remove the export cipher, the eRSA key is not deleted. It is reused at a later date when another export cipher is bound to an SSL

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
			or TCP-based SSL virtual server or service. The eRSA key is deleted when the appliance restarts. This parameter is not applicable when configuring a backend service. Default value: DISABLED Possible values = ENABLED, DISABLED
ersacount	<Double>	Read-write	Refresh count for regeneration of RSA public-key and private-key pair. Zero (0) specifies infinite usage (no refresh). This parameter is not applicable when configuring a backend service. Minimum value = 0 Maximum value = 65534
sessreuse	<String>	Read-write	State of session reuse. Establishing the initial handshake requires CPU-intensive public key encryption

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
			operations. With the ENABLED setting, session key exchange is avoided for session resumption requests received from the client. Default value: ENABLED Possible values = ENABLED, DISABLED
sesstimeout	<Double>	Read-write	Time, in seconds, for which to keep the session active. Any session resumption request received after the timeout period will require a fresh SSL handshake and establishment of a new SSL session. Default value: 300 Minimum value = 0 Maximum value = 4294967294
cipherredirect	<String>	Read-write	State of Cipher Redirect. If this parameter is set to ENABLED, you can configure an SSL virtual server or service to display meaningful error

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
			messages if the SSL handshake fails because of a cipher mismatch between the virtual server or service and the client. This parameter is not applicable when configuring a backend service. Default value: DISABLED Possible values = ENABLED, DISABLED
cipherurl	<String>	Read-write	URL of the page to which to redirect the client in case of a cipher mismatch. Typically, this page has a clear explanation of the error or an alternative location that the transaction can continue from. This parameter is not applicable when configuring a backend service.
sslv2redirect	<String>	Read-write	State of SSLv2 Redirect. If this parameter is set to ENABLED, you can configure an SSL

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
			virtual server or service to display meaningful error messages if the SSL handshake fails because of a protocol version mismatch between the virtual server or service and the client. This parameter is not applicable when configuring a backend service. Default value: DISABLED Possible values = ENABLED, DISABLED
sslv2url	<String>	Read-write	URL of the page to which to redirect the client in case of a protocol version mismatch. Typically, this page has a clear explanation of the error or an alternative location that the transaction can continue from. This parameter is not applicable when configuring a backend service.

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
clientauth	<String>	Read-write	State of client authentication. In service-based SSL offload, the service terminates the SSL handshake if the SSL client does not provide a valid certificate. This parameter is not applicable when configuring a backend service. Default value: DISABLED Possible values = ENABLED, DISABLED
clientcert	<String>	Read-write	Type of client authentication. If this parameter is set to MANDATORY, the appliance terminates the SSL handshake if the SSL client does not provide a valid certificate. With the OPTIONAL setting, the appliance requests a certificate from the SSL clients but proceeds with the SSL transaction even if the client presents an invalid certificate.

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
			This parameter is not applicable when configuring a backend SSL service. Caution: Define proper access control policies before changing this setting to Optional. Possible values = Mandatory, Optional
sslredirect	<String>	Read-write	State of HTTPS redirects for the SSL service. For an SSL session, if the client browser receives a redirect message, the browser tries to connect to the new location. However, the secure SSL session breaks if the object has moved from a secure site (https://) to an unsecure site (http://). Typically, a warning message appears on the screen, prompting

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
			the user to continue or disconnect. If SSL Redirect is ENABLED, the redirect message is automatically converted from http:// to https:// and the SSL session does not break. This parameter is not applicable when configuring a backend service. Default value: DISABLED Possible values = ENABLED, DISABLED
redirectportrewrite	<String>	Read-write	State of the port rewrite while performing HTTPS redirect. If this parameter is set to ENABLED, and the URL from the server does not contain the standard port, the port is rewritten to the standard. Default value: DISABLED

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
			Possible values = ENABLED, DISABLED
ssl2	<String>	Read-write	State of SSLv2 protocol support for the SSL service. This parameter is not applicable when configuring a backend service. Default value: DISABLED Possible values = ENABLED, DISABLED
ssl3	<String>	Read-write	State of SSLv3 protocol support for the SSL service. Default value: ENABLED Possible values = ENABLED, DISABLED
tls1	<String>	Read-write	State of TLSv1.0 protocol support for the SSL service. Default value: ENABLED Possible values = ENABLED, DISABLED

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
tls11	<String>	Read-write	State of TLSv1.1 protocol support for the SSL service. Default value: ENABLED Possible values = ENABLED, DISABLED
tls12	<String>	Read-write	State of TLSv1.2 protocol support for the SSL service. Default value: ENABLED Possible values = ENABLED, DISABLED
tls13	<String>	Read-write	State of TLSv1.3 protocol support for the SSL service. Default value: DISABLED Possible values = ENABLED, DISABLED
snienable	<String>	Read-write	State of the Server Name Indication (SNI) feature on the virtual server and service-based offload. SNI helps to enable SSL encryption on multiple domains on a single virtual server

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
			or service if the domains are controlled by the same organization and share the same second-level domain name. For example, *.sports.net can be used to secure domains such as login.sports.net and help.sports.net. Default value: DISABLED Possible values = ENABLED, DISABLED
ocspstapling	<String>	Read-write	State of OCSP stapling support on the SSL virtual server. Supported only if the protocol used is higher than SSLv3. Possible values: ENABLED: The appliance sends a request to the OCSP responder to check the status of the server certificate and caches the response for the specified time. If the response is valid at the time of SSL handshake with the

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
			client, the OCSP-based server certificate status is sent to the client during the handshake. DISABLED: The appliance does not check the status of the server certificate. . Default value: DISABLED Possible values = ENABLED, DISABLED
serverauth	<String>	Read-write	State of server authentication support for the SSL service. Default value: DISABLED Possible values = ENABLED, DISABLED
commonname	<String>	Read-write	Name to be checked against the CommonName (CN) field in the server certificate bound to the SSL server. Minimum length = 1

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
pushenctrigger	<String>	Read-write	Trigger encryption on the basis of the PUSH flag value. Available settings function as follows: * ALWAYS - Any PUSH packet triggers encryption. * IGNORE - Ignore PUSH packet for triggering encryption. * MERGE - For a consecutive sequence of PUSH packets, the last PUSH packet triggers encryption. * TIMER - PUSH packet triggering encryption is delayed by the time defined in the set ssl parameter command or in the Change Advanced SSL Settings dialog box. Possible values = Always, Merge, Ignore, Timer

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
sendclosenotify	<String>	Read-write	Enable sending SSL Close-Notify at the end of a transaction. Default value: YES Possible values = YES, NO
dtlsprofilename	<String>	Read-write	Name of the DTLS profile that contains DTLS settings for the service. Minimum length = 1 Maximum length = 127
sslprofile	<String>	Read-write	Name of the SSL profile that contains SSL settings for the service. Minimum length = 1 Maximum length = 127
strictsigdigestcheck	<String>	Read-write	Parameter indicating to check whether peer's certificate during TLS1.2 handshake is signed with one of signature-hash combination supported by Citrix ADC. Default value: DISABLED

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
			Possible values = ENABLED, DISABLED
nonfipsciphers	<String>	Read-only	The state of usage of non FIPS approved ciphers. Default value: DISABLED Possible values = ENABLED, DISABLED
service	<Double>	Read-only	.
skipcaname	<Boolean>	Read-only	The flag is used to indicate whether this particular CA certificate's CA_Name needs to be sent to the SSL client while requesting for client certificate in a SSL handshake.
dtlsflag	<Boolean>	Read-only	The flag is used to indicate whether DTLS is set or not.
__count	<Double>	Read-only	count parameter

Operations

(click to see Properties)

UPDATE| UNSET| GET (ALL)| GET| COUNT

Some options that you can use for each operations:

- **Getting warnings in response:**NITRO allows you to get warnings in an operation by specifying the "warning" query parameter as "yes". For example, to get warnings while connecting to the Citrix ADC appliance, the URL is as follows:

  http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/login?warning=yes

  If any, the warnings are displayed in the response payload with the HTTP code "209 X-NITRO-WARNING".

- **Authenticated access for individual NITRO operations:**NITRO allows you to logon to the Citrix ADC appliance to perform individual operations. You can use this option instead of creating a NITRO session (using the login object) and then using that session to perform all operations,

  To do this, you must specify the username and password in the request header of the NITRO request as follows:

  X-NITRO-USER:<username>

  X-NITRO-PASS:<password>

  Note: In such cases, make sure that the request header DOES not include the following:

  Cookie:NITRO_AUTH_TOKEN=<tokenvalue>

`*Note:`*

Mandatory parameters are marked in redand placeholder content is marked in <green>.

update

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/sslservice

HTTP Method:PUT

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Content-Type:application/json

Request Payload:

```
{"sslservice":{
```

```
<b>"servicename":<String_value>,

</b>"dh":<String_value>,

"dhfile":<String_value>,

"dhcount":<Double_value>,

"dhkeyexpsizelimit":<String_value>,

"ersa":<String_value>,

"ersacount":<Double_value>,

"sessreuse":<String_value>,

"sesstimeout":<Double_value>,

"cipherredirect":<String_value>,

"cipherurl":<String_value>,

"sslv2redirect":<String_value>,

"sslv2url":<String_value>,

"clientauth":<String_value>,

"clientcert":<String_value>,

"sslredirect":<String_value>,

"redirectportrewrite":<String_value>,

"ssl2":<String_value>,

"ssl3":<String_value>,

"tls1":<String_value>,

"tls11":<String_value>,

"tls12":<String_value>,
```

```
"tls13":<String_value>,

"snienable":<String_value>,

"ocspstapling":<String_value>,

"serverauth":<String_value>,

"commonname":<String_value>,

"pushenctrigger":<String_value>,

"sendclosenotify":<String_value>,

"dtlsprofilename":<String_value>,

"sslprofile":<String_value>,

"strictsigdigestcheck":<String_value>

}}
```

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error

unset

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/sslservice?**action=unset**

HTTP Method:POST

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Content-Type:application/json

Request Payload:

```
{"sslservice":{
<b>"servicename":<String_value>,
</b>"dh":true,
"dhfile":true,
"dhcount":true,
"dhkeyexpsizelimit":true,
"ersa":true,
"ersacount":true,
"sessreuse":true,
"sesstimeout":true,
"cipherredirect":true,
"cipherurl":true,
"sslv2redirect":true,
"sslv2url":true,
"clientauth":true,
"clientcert":true,
"sslredirect":true,
"redirectportrewrite":true,
"ssl2":true,
"ssl3":true,
"tls1":true,
```

```
"tls11":true,

"tls12":true,

"tls13":true,

"snienable":true,

"ocspstapling":true,

"serverauth":true,

"commonname":true,

"sendclosenotify":true,

"dtlsprofilename":true,

"sslprofile":true,

"strictsigdigestcheck":true

}}
```

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error

get (all)

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/sslservice

Query-parameters:

attrs

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/sslservice?**attrs=property-name1,property-name2**

Use this query parameter to specify the resource details that you want to retrieve.

filter

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/sslservice?**filter=property-name1:property-val1,property-name2:property-val2**

Use this query-parameter to get the filtered set of sslservice resources configured on Citrix ADC. Filtering can be done on any of the properties of the resource.

view

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/sslservice?**view=summary**

Note: By default, the retrieved results are displayed in detail view (?view=detail).

pagination

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/sslservice?**pagesize=#no;pageno=#no**

Use this query-parameter to get the sslservice resources in chunks.

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "sslservice": [ {

"servicename":<String_value>,

"dh":<String_value>,

"dhfile":<String_value>,
```

```
"dhcount":<Double_value>,

"dhkeyexpsizelimit":<String_value>,

"ersa":<String_value>,

"ersacount":<Double_value>,

"sessreuse":<String_value>,

"sesstimeout":<Double_value>,

"cipherredirect":<String_value>,

"cipherurl":<String_value>,

"sslv2redirect":<String_value>,

"sslv2url":<String_value>,

"clientauth":<String_value>,

"clientcert":<String_value>,

"sslredirect":<String_value>,

"redirectportrewrite":<String_value>,

"nonfipsciphers":<String_value>,

"ssl2":<String_value>,

"ssl3":<String_value>,

"tls1":<String_value>,

"tls11":<String_value>,

"tls12":<String_value>,

"tls13":<String_value>,

"snienable":<String_value>,
```

```
"ocspstapling":<String_value>,

"serverauth":<String_value>,

"commonname":<String_value>,

"service":<Double_value>,

"pushenctrigger":<String_value>,

"skipcaname":<Boolean_value>,

"sendclosenotify":<String_value>,

"dtlsprofilename":<String_value>,

"dtlsflag":<Boolean_value>,

"sslprofile":<String_value>,

"strictsigdigestcheck":<String_value>

}]}
```

get

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/sslservice/servicename_value<String>

Query-parameters:

attrs

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/sslservice/servicename_value<String>?**attrs=property-name1,property-name2**

Use this query parameter to specify the resource details that you want to retrieve.

view

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/sslservice/servicename_value<String>?**view=summary**

Note: By default, the retrieved results are displayed in detail view (?view=detail).

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "sslservice": [ {

"servicename":<String_value>,

"dh":<String_value>,

"dhfile":<String_value>,

"dhcount":<Double_value>,

"dhkeyexpsizelimit":<String_value>,

"ersa":<String_value>,

"ersacount":<Double_value>,

"sessreuse":<String_value>,

"sesstimeout":<Double_value>,

"cipherredirect":<String_value>,

"cipherurl":<String_value>,

"sslv2redirect":<String_value>,
```

```
"sslv2url":<String_value>,

"clientauth":<String_value>,

"clientcert":<String_value>,

"sslredirect":<String_value>,

"redirectportrewrite":<String_value>,

"nonfipsciphers":<String_value>,

"ssl2":<String_value>,

"ssl3":<String_value>,

"tls1":<String_value>,

"tls11":<String_value>,

"tls12":<String_value>,

"tls13":<String_value>,

"snienable":<String_value>,

"ocspstapling":<String_value>,

"serverauth":<String_value>,

"commonname":<String_value>,

"service":<Double_value>,

"pushenctrigger":<String_value>,

"skipcaname":<Boolean_value>,

"sendclosenotify":<String_value>,

"dtlsprofilename":<String_value>,

"dtlsflag":<Boolean_value>,
```

```
"sslprofile":<String_value>,

"strictsigdigestcheck":<String_value>

}]}
```

count

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/sslservice?**count=yes**

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "sslservice": [ { "__count": "#no"} ] }
```

sslservicegroup

Configuration for SSL service group resource.

Properties

(click to see Operations)

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
servicegroupname	<String>	Read-write	Name of the SSL service group for which to set advanced configuration. Minimum length = 1
sslprofile	<String>	Read-write	Name of the SSL profile that contains SSL settings for the Service Group. Minimum length = 1 Maximum length = 127
sessreuse	<String>	Read-write	State of session reuse. Establishing the initial handshake requires CPU-intensive public key encryption operations. With the ENABLED setting, session key exchange is avoided for session resumption requests received from the client. Default value: ENABLED Possible values = ENABLED, DISABLED
sesstimeout	<Double>	Read-write	Time, in seconds, for which to keep the session active. Any

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
			session resumption request received after the timeout period will require a fresh SSL handshake and establishment of a new SSL session. Default value: 300 Minimum value = 0 Maximum value = 4294967294
ssl3	<String>	Read-write	State of SSLv3 protocol support for the SSL service group. Default value: ENABLED Possible values = ENABLED, DISABLED
tls1	<String>	Read-write	State of TLSv1.0 protocol support for the SSL service group. Default value: ENABLED Possible values = ENABLED, DISABLED
tls11	<String>	Read-write	State of TLSv1.1 protocol support for

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
			the SSL service group. Default value: ENABLED Possible values = ENABLED, DISABLED
tls12	<String>	Read-write	State of TLSv1.2 protocol support for the SSL service group. Default value: ENABLED Possible values = ENABLED, DISABLED
tls13	<String>	Read-write	State of TLSv1.3 protocol support for the SSL service group. Default value: DISABLED Possible values = ENABLED, DISABLED
snienable	<String>	Read-write	State of the Server Name Indication (SNI) feature on the service. SNI helps to enable SSL encryption on multiple domains on a single virtual server or service if the

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
			domains are controlled by the same organization and share the same second-level domain name. For example, *.sports.net can be used to secure domains such as login.sports.net and help.sports.net. Default value: DISABLED Possible values = ENABLED, DISABLED
ocspstapling	<String>	Read-write	State of OCSP stapling support on the SSL virtual server. Supported only if the protocol used is higher than SSLv3. Possible values: ENABLED: The appliance sends a request to the OCSP responder to check the status of the server certificate and caches the response for the specified time. If the response is valid at the time of SSL handshake with the client, the OCSP-

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
			based server certificate status is sent to the client during the handshake. DISABLED: The appliance does not check the status of the server certificate. Default value: DISABLED Possible values = ENABLED, DISABLED
serverauth	<String>	Read-write	State of server authentication support for the SSL service group. Default value: DISABLED Possible values = ENABLED, DISABLED
commonname	<String>	Read-write	Name to be checked against the CommonName (CN) field in the server certificate bound to the SSL server. Minimum length = 1

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
sendclosenotify	<String>	Read-write	Enable sending SSL Close-Notify at the end of a transaction. Default value: YES Possible values = YES, NO
strictsigdigestcheck	<String>	Read-write	Parameter indicating to check whether peer's certificate is signed with one of signature-hash combination supported by Citrix ADC. Default value: DISABLED Possible values = ENABLED, DISABLED
dh	<String>	Read-only	The state of DH key exchange support for the SSL service group. Possible values = ENABLED, DISABLED
dhfile	<String>	Read-only	The file name and path for the DH parameter.
dhcount	<Double>	Read-only	The refresh count for the re-generation of DH public-key and private-key from the DH parameter.

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
			Minimum value = 0 Maximum value = 65534
dhkeyexpsizelimit	<String>	Read-only	This option enables the use of NIST recommended (NIST Special Publication 800-56A) bit size for private-key size. For example, for DH params of size 2048bit, the private-key size recommended is 224bits. This is rounded-up to 256bits. Possible values = ENABLED, DISABLED
ersa	<String>	Read-only	The state of Ephemeral RSA key exchange support for the SSL service group.Ephemeral RSA is used for export ciphers. Possible values = ENABLED, DISABLED
ersacount	<Double>	Read-only	The refresh count for the re-generation of RSA public-key and private-key pair. Minimum value = 0

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
			Maximum value = 65534
cipherredirect	<String>	Read-only	The state of Cipher Redirect feature. Cipher Redirect feature can be used to provide more readable information to SSL clients about mismatch in ciphers between the client and the SSL vserver. Possible values = ENABLED, DISABLED
cipherurl	<String>	Read-only	The redirect URL to be used with the Cipher Redirect feature.
sslv2redirect	<String>	Read-only	The state of SSLv2 Redirect feature.SSLv2 Redirect feature can be used to provide more readable information to SSL client about non-support of SSLv2 protocol on the SSL vserver. Possible values = ENABLED, DISABLED
sslv2url	<String>	Read-only	The redirect URL to be used with SSLv2 Redirect feature.

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
clientauth	<String>	Read-only	The state of Client-Authentication support for the SSL service group. Possible values = ENABLED, DISABLED
clientcert	<String>	Read-only	The rule for client certificate requirement in client authentication. Possible values = Mandatory, Optional
sslredirect	<String>	Read-only	The state of HTTPS redirects for the SSL service group. This is required for the proper functioning of the redirect messages from the server. The redirect message from the server provides the new location for the moved object. This is contained in the HTTP header field: Location, e.g. Location: http://www.moved.org/here.html For the SSL session, if the client browser receives

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
			this message, the browser will try to connect to the new location. This will break the secure SSL session, as the object has moved from a secure site (https://) to an un-secure one (http://). Generally browsers flash a warning message on the screen and prompt the user, either to continue or disconnect. The above feature, when enabled will automatically convert all such http:// redirect message to https://. This will not break the client SSL session. Note: The set ssl service command can be used for configuring a front-end SSL service for service based SSL Off-Loading, or a backend SSL

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
			service for backend-encryption setup. Possible values = ENABLED, DISABLED
redirectportrewrite	<String>	Read-only	The state of port-rewrite feature. Possible values = ENABLED, DISABLED
nonfipsciphers	<String>	Read-only	The state of usage of non FIPS approved ciphers. Default value: DISABLED Possible values = ENABLED, DISABLED
ssl2	<String>	Read-only	The state of SSLv2 protocol support for the SSL service group. Possible values = ENABLED, DISABLED
ocspcheck	<String>	Read-only	The state of the OCSP check parameter. (Mandatory/ Optional). Possible values = Mandatory, Optional

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
crlcheck	<String>	Read-only	The state of the CRL check parameter. (Mandatory/ Optional). Possible values = Mandatory, Optional
cleartextport	<Integer>	Read-only	The port on the back-end web-servers where the clear-text data is sent by system. Use this setting for the wildcard IP based SSL Acceleration configuration (*:443). Minimum value = 0 Maximum value = 65534 Range 1 - 65535 * in CLI is represented as 65535 in NITRO API
servicename	<String>	Read-only	The service name.
ca	<Boolean>	Read-only	CA certificate.
snicert	<Boolean>	Read-only	The name of the CertKey. Use this option to bind Certkey(s) which will be used in SNI processing.
__count	<Double>	Read-only	count parameter

Operations

(click to see Properties)

UPDATE| UNSET| GET (ALL)| GET| COUNT

Some options that you can use for each operations:

- **Getting warnings in response:**NITRO allows you to get warnings in an operation by specifying the "warning" query parameter as "yes". For example, to get warnings while connecting to the Citrix ADC appliance, the URL is as follows:

  http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/login?warning=yes

  If any, the warnings are displayed in the response payload with the HTTP code "209 X-NITRO-WARNING".

- **Authenticated access for individual NITRO operations:**NITRO allows you to logon to the Citrix ADC appliance to perform individual operations. You can use this option instead of creating a NITRO session (using the login object) and then using that session to perform all operations,

  To do this, you must specify the username and password in the request header of the NITRO request as follows:

  X-NITRO-USER:<username>

  X-NITRO-PASS:<password>

  Note: In such cases, make sure that the request header DOES not include the following:

  Cookie:NITRO_AUTH_TOKEN=<tokenvalue>

`*Note:`*

Mandatory parameters are marked in redand placeholder content is marked in <green>.

update

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/sslservicegroup

HTTP Method:PUT

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Content-Type:application/json

Request Payload:

```
{"sslservicegroup":{

<b>"servicegroupname":<String_value>,

</b>"sslprofile":<String_value>,

"sessreuse":<String_value>,

"sesstimeout":<Double_value>,

"ssl3":<String_value>,

"tls1":<String_value>,

"tls11":<String_value>,

"tls12":<String_value>,

"tls13":<String_value>,

"snienable":<String_value>,

"ocspstapling":<String_value>,

"serverauth":<String_value>,

"commonname":<String_value>,

"sendclosenotify":<String_value>,

"strictsigdigestcheck":<String_value>

}}
```

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error

unset

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/sslservicegroup?**action=unset**

HTTP Method:POST

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Content-Type:application/json

Request Payload:

```
{"sslservicegroup":{

<b>"servicegroupname":<String_value>,

</b>"sslprofile":true,

"sessreuse":true,

"sesstimeout":true,

"ssl3":true,

"tls1":true,

"tls11":true,

"tls12":true,

"tls13":true,

"snienable":true,

"ocspstapling":true,

"serverauth":true,
```

```
"commonname":true,

"sendclosenotify":true,

"strictsigdigestcheck":true

}}
```

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error

get (all)

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/sslservicegroup

Query-parameters:

attrs

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/sslservicegroup?**attrs=property-name1,property-name2**

Use this query parameter to specify the resource details that you want to retrieve.

filter

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/sslservicegroup?**filter=property-name1:property-val1,property-name2:property-val2**

Use this query-parameter to get the filtered set of sslservicegroup resources configured on Citrix ADC. Filtering can be done on any of the properties of the resource.

view

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/sslservicegroup?**view=summary**

Note: By default, the retrieved results are displayed in detail view (?view=detail).

pagination

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/sslservicegroup?**pagesize=#no;pageno=#no**

Use this query-parameter to get the sslservicegroup resources in chunks.

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "sslservicegroup": [ {

"servicegroupname":<String_value>,

"dh":<String_value>,

"dhfile":<String_value>,

"dhcount":<Double_value>,

"dhkeyexpsizelimit":<String_value>,

"ersa":<String_value>,

"ersacount":<Double_value>,

"sessreuse":<String_value>,

"sesstimeout":<Double_value>,

"cipherredirect":<String_value>,
```

```
"cipherurl":<String_value>,

"sslv2redirect":<String_value>,

"sslv2url":<String_value>,

"clientauth":<String_value>,

"clientcert":<String_value>,

"sslredirect":<String_value>,

"redirectportrewrite":<String_value>,

"nonfipsciphers":<String_value>,

"ssl2":<String_value>,

"ssl3":<String_value>,

"tls1":<String_value>,

"tls11":<String_value>,

"tls12":<String_value>,

"tls13":<String_value>,

"snienable":<String_value>,

"ocspstapling":<String_value>,

"serverauth":<String_value>,

"commonname":<String_value>,

"ocspcheck":<String_value>,

"crlcheck":<String_value>,

"cleartextport":<Integer_value>,

"servicename":<String_value>,
```

```
"ca":<Boolean_value>,

"snicert":<Boolean_value>,

"sendclosenotify":<String_value>,

"sslprofile":<String_value>,

"strictsigdigestcheck":<String_value>

}]}
```

get

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/sslservicegroup/servicegroupname_value<String>

Query-parameters:

attrs

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/sslservicegroup/servicegroupname_value<String>?**attrs=property-name1,property-name2**

Use this query parameter to specify the resource details that you want to retrieve.

view

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/sslservicegroup/servicegroupname_value<String>?**view=summary**

Note: By default, the retrieved results are displayed in detail view (?view=detail).

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "sslservicegroup": [ {

"servicegroupname":<String_value>,

"dh":<String_value>,

"dhfile":<String_value>,

"dhcount":<Double_value>,

"dhkeyexpsizelimit":<String_value>,

"ersa":<String_value>,

"ersacount":<Double_value>,

"sessreuse":<String_value>,

"sesstimeout":<Double_value>,

"cipherredirect":<String_value>,

"cipherurl":<String_value>,

"sslv2redirect":<String_value>,

"sslv2url":<String_value>,

"clientauth":<String_value>,

"clientcert":<String_value>,

"sslredirect":<String_value>,
```

```
"redirectportrewrite":<String_value>,

"nonfipsciphers":<String_value>,

"ssl2":<String_value>,

"ssl3":<String_value>,

"tls1":<String_value>,

"tls11":<String_value>,

"tls12":<String_value>,

"tls13":<String_value>,

"snienable":<String_value>,

"ocspstapling":<String_value>,

"serverauth":<String_value>,

"commonname":<String_value>,

"ocspcheck":<String_value>,

"crlcheck":<String_value>,

"cleartextport":<Integer_value>,

"servicename":<String_value>,

"ca":<Boolean_value>,

"snicert":<Boolean_value>,

"sendclosenotify":<String_value>,

"sslprofile":<String_value>,

"strictsigdigestcheck":<String_value>

}]}
```

count

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/sslservicegroup?**count=yes**

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "sslservicegroup": [ { "__count": "#no"} ] }
```

sslservicegroup_binding

Binding object which returns the resources bound to sslservicegroup.

Properties

(click to see Operations)

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
servicegroupname	<String>	Read-write	Name of the SSL service group for which to show detailed information. Minimum length = 1

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
sslservicegroup_sslcipher_binding	<sslservicegroup_sslcipher_binding[]>	Read-only	sslcipher that can be bound to sslservicegroup.
sslservicegroup_eccurve_binding	<sslservicegroup_eccurve_binding[]>	Read-only	ecccurve that can be bound to sslservicegroup.
sslservicegroup_sslcertkey_binding	<sslservicegroup_sslcertkey_binding[]>	Read-only	sslcertkey that can be bound to sslservicegroup.
sslservicegroup_sslciphersuite_binding	<sslservicegroup_sslciphersuite_binding[]>	Read-only	sslciphersuite that can be bound to sslservicegroup.

Operations

(click to see Properties)

GET| GET (ALL)

Some options that you can use for each operations:

- **Getting warnings in response:**NITRO allows you to get warnings in an operation by specifying the "warning" query parameter as "yes". For example, to get warnings while connecting to the Citrix ADC appliance, the URL is as follows:

  http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/login?warning=yes

  If any, the warnings are displayed in the response payload with the HTTP code "209 X-NITRO-WARNING".

- **Authenticated access for individual NITRO operations:**NITRO allows you to logon to the Citrix ADC appliance to perform individual operations. You can use this option instead of creating a NITRO session (using the login object) and then using that session to perform all operations,

  To do this, you must specify the username and password in the request header of the NITRO request as follows:

  X-NITRO-USER:<username>

X-NITRO-PASS:<password>

Note: In such cases, make sure that the request header DOES not include the following:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue>

`*Note:`*

Mandatory parameters are marked in redand placeholder content is marked in <green>.

get

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/sslservicegroup_binding/servicegroupname_value<String>

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "sslservicegroup_binding": [ {

"servicegroupname":<String_value>,

"sslservicegroup_sslcipher_binding":<sslservicegroup_sslcipher_binding[]_value>,

"sslservicegroup_sslciphersuite_binding":<sslservicegroup_sslciphersuite_binding[]_value>,

"sslservicegroup_ecccurve_binding":<sslservicegroup_ecccurve_binding[]_value>,
```

```
"sslservicegroup_sslcertkey_binding":<sslservicegroup_sslcertkey_binding[]
_value>

}]}
```

get (all)

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/sslservicegroup_binding

Query-parameters:

bulkbindings

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/sslservicegroup_binding?**bulkbindings=yes**

NITRO allows you to fetch bindings in bulk.

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "sslservicegroup_binding": [ {

"servicegroupname":<String_value>,

"sslservicegroup_sslcipher_binding":<sslservicegroup_sslcipher_binding[]_v
alue>,

"sslservicegroup_sslciphersuite_binding":<sslservicegroup_sslciphersuite_b
```

```
inding[]_value>,

"sslservicegroup_ecccurve_binding":<sslservicegroup_ecccurve_binding[]_val
ue>,

"sslservicegroup_sslcertkey_binding":<sslservicegroup_sslcertkey_binding[]
_value>

}]}
```

sslservicegroup_ecccurve_binding

Binding object showing the ecccurve that can be bound to sslservicegroup.

Properties

(click to see Operations)

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
servicegroupname	<String>	Read-write	The name of the SSL service to which the SSL policy needs to be bound. Minimum length = 1
ecccurvename	<String>	Read-write	Named ECC curve bound to servicegroup. Possible values = ALL, P_224, P_256, P_384, P_521
__count	<Double>	Read-write	count parameter

Operations

(click to see Properties)

ADD:| DELETE:| GET| GET (ALL)| COUNT

Some options that you can use for each operations:

- **Getting warnings in response:**NITRO allows you to get warnings in an operation by specifying the "warning" query parameter as "yes". For example, to get warnings while connecting to the Citrix ADC appliance, the URL is as follows:

  http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/login?warning=yes

  If any, the warnings are displayed in the response payload with the HTTP code "209 X-NITRO-WARNING".

- **Authenticated access for individual NITRO operations:**NITRO allows you to logon to the Citrix ADC appliance to perform individual operations. You can use this option instead of creating a NITRO session (using the login object) and then using that session to perform all operations,

  To do this, you must specify the username and password in the request header of the NITRO request as follows:

  X-NITRO-USER:<username>

  X-NITRO-PASS:<password>

  Note: In such cases, make sure that the request header DOES not include the following:

  Cookie:NITRO_AUTH_TOKEN=<tokenvalue>

`*Note:`*

Mandatory parameters are marked in redand placeholder content is marked in <green>.

add

URL:http://<Citrix-ADC-IP-address(NSIP)/nitro/v1/config/sslservicegroup_ecccurve_binding

HTTP Method:PUT

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Content-Type:application/json

Request Payload:

```
{
```

```
"sslservicegroup_ecccurve_binding":{

<b>"servicegroupname":<String_value>,

</b>"ecccurvename":<String_value>

}}
```

Response:

HTTP Status Code on Success: 201 Created HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error

delete

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/sslservicegroup_ecccurve_binding/servicegroupname_value<String>

Query-parameters:

args

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/sslservicegroup_ecccurve_binding/servicegroupname_value<String>?**args=ecccurvename:<String_value>**

HTTP Method:DELETE

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue>

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error

get

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/sslservicegroup_ecccurve_binding/servicegroupname_value<String>

Query-parameters:

filter

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/sslservicegroup_ecccurve_binding/servicegroupname_value<String>?**filter=property-name1:property-value1,property-name2:property-value2**

Use this query-parameter to get the filtered set of sslservicegroup_ecccurve_binding resources configured on Citrix ADC. Filtering can be done on any of the properties of the resource.

pagination

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/sslservicegroup_ecccurve_binding/servicegroupname_value<String>?**pagesize=#no;pageno=#no**

Use this query-parameter to get the sslservicegroup_ecccurve_binding resources in chunks.

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "sslservicegroup_ecccurve_binding": [ {

"servicegroupname":<String_value>,

"ecccurvename":<String_value>

}]}
```

get (all)

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/sslservicegroup_ecccurve_binding

Query-parameters:

bulkbindings

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/sslservicegroup_ecccurve_binding?**bulkbindings=yes**

NITRO allows you to fetch bindings in bulk.

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "sslservicegroup_ecccurve_binding": [ {

"servicegroupname":<String_value>,

"ecccurvename":<String_value>

}]}
```

count

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/sslservicegroup_ecccurve_binding/servicegroupname_value<String>?**count=yes**

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{"sslservicegroup_ecccurve_binding": [ { "__count": "#no"} ] }
```

sslservicegroup_sslcertkey_binding

Binding object showing the sslcertkey that can be bound to sslservicegroup.

Properties

(click to see Operations)

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
servicegroupname	<String>	Read-write	The name of the SSL service to which the SSL policy needs to be bound. Minimum length = 1
ca	<Boolean>	Read-write	CA certificate.

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
crlcheck	<String>	Read-write	The state of the CRL check parameter. (Mandatory/ Optional). Possible values = Mandatory, Optional
certkeyname	<String>	Read-write	The name of the certificate bound to the SSL service group.
snicert	<Boolean>	Read-write	The name of the CertKey. Use this option to bind Certkey(s) which will be used in SNI processing.
ocspcheck	<String>	Read-write	The state of the OCSP check parameter. (Mandatory/ Optional). Possible values = Mandatory, Optional
__count	<Double>	Read-write	count parameter

Operations

(click to see Properties)

ADD:| DELETE:| GET| GET (ALL)| COUNT

Some options that you can use for each operations:

- **Getting warnings in response:**NITRO allows you to get warnings in an operation by specifying the "warning" query parameter as "yes". For example, to get warnings while connecting to the Citrix ADC appliance, the URL is as follows:

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/login?warning=yes

If any, the warnings are displayed in the response payload with the HTTP code "209 X-NITRO-WARNING".

- **Authenticated access for individual NITRO operations:**NITRO allows you to logon to the Citrix ADC appliance to perform individual operations. You can use this option instead of creating a NITRO session (using the login object) and then using that session to perform all operations,

  To do this, you must specify the username and password in the request header of the NITRO request as follows:

  X-NITRO-USER:<username>

  X-NITRO-PASS:<password>

  Note: In such cases, make sure that the request header DOES not include the following:

  Cookie:NITRO_AUTH_TOKEN=<tokenvalue>

`*Note:`*

Mandatory parameters are marked in redand placeholder content is marked in <green>.

add

URL:http://<Citrix-ADC-IP-address(NSIP)/nitro/v1/config/sslservicegroup_sslcertkey_binding

HTTP Method:PUT

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Content-Type:application/json

Request Payload:

```
{

"sslservicegroup_sslcertkey_binding":{

<b>"servicegroupname":<String_value>,

</b>"certkeyname":<String_value>,
```

```
"ca":<Boolean_value>,

"crlcheck":<String_value>,

"snicert":<Boolean_value>,

"ocspcheck":<String_value>

}}
```

Response:

HTTP Status Code on Success: 201 Created HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error

delete

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/sslservicegroup_sslcertkey_binding/servicegroupname_value<String>

Query-parameters:

args

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/sslservicegroup_sslcertkey_binding/servicegroupname_value<String>?
args=certkeyname:<String_value>,ca:<Boolean_value>,crlcheck:<String_value>,snicert:<Boolean_valu

HTTP Method:DELETE

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue>

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error

get

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/sslservicegroup_sslcertkey_binding/servicegroupname_value<String>

Query-parameters:

filter

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/sslservicegroup_sslcertkey_binding/servicegroupname_value<String>?**filter=property-name1:property-value1,property-name2:property-value2**

Use this query-parameter to get the filtered set of sslservicegroup_sslcertkey_binding resources configured on Citrix ADC. Filtering can be done on any of the properties of the resource.

pagination

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/sslservicegroup_sslcertkey_binding/servicegroupname_value<String>?**pagesize=#no;pageno=#no**

Use this query-parameter to get the sslservicegroup_sslcertkey_binding resources in chunks.

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "sslservicegroup_sslcertkey_binding": [ {

"servicegroupname":<String_value>,
```

```
"ca":<Boolean_value>,

"crlcheck":<String_value>,

"certkeyname":<String_value>,

"snicert":<Boolean_value>,

"ocspcheck":<String_value>

}]}
```

get (all)

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/sslservicegroup_sslcertkey_binding

Query-parameters:

bulkbindings

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/sslservicegroup_sslcertkey_binding?**bulkbindings=yes**

NITRO allows you to fetch bindings in bulk.

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "sslservicegroup_sslcertkey_binding": [ {

"servicegroupname":<String_value>,

"ca":<Boolean_value>,

"crlcheck":<String_value>,

"certkeyname":<String_value>,

"snicert":<Boolean_value>,

"ocspcheck":<String_value>

}]}
```

count

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/sslservicegroup_sslcertkey_binding/servicegroupname_value<String>?**count=yes**

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{"sslservicegroup_sslcertkey_binding": [ { "__count": "#no"} ] }
```

sslservicegroup_sslciphersuite_binding

Binding object showing the sslciphersuite that can be bound to sslservicegroup.

Properties

(click to see Operations)

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
ciphername	<String>	Read-write	The name of the cipher group/alias/ name configured for the SSL service group.
servicegroupname	<String>	Read-write	The name of the SSL service to which the SSL policy needs to be bound. Minimum length = 1
description	<String>	Read-write	The description of the cipher.
__count	<Double>	Read-write	count parameter

Operations

(click to see Properties)

ADD:| DELETE:| GET| GET (ALL)| COUNT

Some options that you can use for each operations:

- **Getting warnings in response:**NITRO allows you to get warnings in an operation by specifying the "warning" query parameter as "yes". For example, to get warnings while connecting to the Citrix ADC appliance, the URL is as follows:

  http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/login?warning=yes

If any, the warnings are displayed in the response payload with the HTTP code "209 X-NITRO-WARNING".

- **Authenticated access for individual NITRO operations:**NITRO allows you to logon to the Citrix ADC appliance to perform individual operations. You can use this option instead of creating a NITRO session (using the login object) and then using that session to perform all operations,

  To do this, you must specify the username and password in the request header of the NITRO request as follows:

  X-NITRO-USER:<username>

  X-NITRO-PASS:<password>

  Note: In such cases, make sure that the request header DOES not include the following:

  Cookie:NITRO_AUTH_TOKEN=<tokenvalue>

`*Note:`*

Mandatory parameters are marked in redand placeholder content is marked in <green>.

add

URL:http://<Citrix-ADC-IP-address(NSIP)/nitro/v1/config/sslservicegroup_sslciphersuite_binding

HTTP Method:PUT

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Content-Type:application/json

Request Payload:

```
{

"sslservicegroup_sslciphersuite_binding":{

<b>"servicegroupname":<String_value>,

</b>"ciphername":<String_value>
```

```
}}
```

Response:

HTTP Status Code on Success: 201 Created HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error

delete

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/sslservicegroup_sslciphersuite_binding/servicegroupname_value<String>

Query-parameters:

args

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/sslservicegroup_sslciphersuite_binding/servicegroupname_value<String>?**args=ciphername:<String_value>**

HTTP Method:DELETE

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue>

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error

get

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/sslservicegroup_sslciphersuite_binding/servicegroupname_value<String>

Query-parameters:

filter

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/sslservicegroup_sslciphersuite_binding/servicegroupname_value<String>?**filter=property-name1:property-value1,property-name2:property-value2**

Use this query-parameter to get the filtered set of sslservicegroup_sslciphersuite_binding resources configured on Citrix ADC. Filtering can be done on any of the properties of the resource.

pagination

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/sslservicegroup_sslciphersuite_binding/servicegroupname_value<String>?**pagesize=#no;pageno=#no**

Use this query-parameter to get the sslservicegroup_sslciphersuite_binding resources in chunks.

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "sslservicegroup_sslciphersuite_binding": [ {

"ciphername":<String_value>,

"servicegroupname":<String_value>,

"description":<String_value>

}]}
```

get (all)

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/sslservicegroup_sslciphersuite_binding

Query-parameters:

bulkbindings

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/sslservicegroup_sslciphersuite_binding?**bulkbindings=yes**

NITRO allows you to fetch bindings in bulk.

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "sslservicegroup_sslciphersuite_binding": [ {

"ciphername":<String_value>,

"servicegroupname":<String_value>,

"description":<String_value>

}]}
```

count

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/sslservicegroup_sslciphersuite_binding/servicegroupname_value<String>?**count=yes**

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{"sslservicegroup_sslciphersuite_binding": [ { "__count": "#no"} ] }
```

sslservicegroup_sslcipher_binding

Binding object showing the sslcipher that can be bound to sslservicegroup.

Properties

(click to see Operations)

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
ciphername	<String>	Read-write	A cipher-suite can consist of an individual cipher name, the system predefined cipher-alias name, or user defined cipher-group name.

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
cipheraliasname	<String>	Read-write	The name of the cipher group/alias/ name configured for the SSL service group.
servicegroupname	<String>	Read-write	The name of the SSL service to which the SSL policy needs to be bound. Minimum length = 1
description	<String>	Read-write	The description of the cipher.
__count	<Double>	Read-write	count parameter

Operations

(click to see Properties)

ADD:| DELETE:| GET| GET (ALL)| COUNT

Some options that you can use for each operations:

- **Getting warnings in response:**NITRO allows you to get warnings in an operation by specifying the "warning" query parameter as "yes". For example, to get warnings while connecting to the Citrix ADC appliance, the URL is as follows:

  http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/login?warning=yes

  If any, the warnings are displayed in the response payload with the HTTP code "209 X-NITRO-WARNING".

- **Authenticated access for individual NITRO operations:**NITRO allows you to logon to the Citrix ADC appliance to perform individual operations. You can use this option instead of creating a NITRO session (using the login object) and then using that session to perform all operations,

  To do this, you must specify the username and password in the request header of the NITRO request as follows:

X-NITRO-USER:<username>

X-NITRO-PASS:<password>

Note: In such cases, make sure that the request header DOES not include the following:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue>

`*Note:`*

Mandatory parameters are marked in redand placeholder content is marked in <green>.

add

URL:http://<Citrix-ADC-IP-address(NSIP)/nitro/v1/config/sslservicegroup_sslcipher_binding

HTTP Method:PUT

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Content-Type:application/json

Request Payload:

```
{

"sslservicegroup_sslcipher_binding":{

<b>"servicegroupname":<String_value>,

</b>"ciphername":<String_value>

}}
```

Response:

HTTP Status Code on Success: 201 Created HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error

delete

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/sslservicegroup_sslcipher_binding/servicegroupname_value<String>

Query-parameters:

args

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/sslservicegroup_sslcipher_binding/servicegroupname_value<String>?**args=ciphername:<String_value>**

HTTP Method:DELETE

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue>

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error

get

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/sslservicegroup_sslcipher_binding/servicegroupname_value<String>

Query-parameters:

filter

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/sslservicegroup_sslcipher_binding/servicegroupname_value<String>?**filter=property-name1:property-value1,property-name2:property-value2**

Use this query-parameter to get the filtered set of sslservicegroup_sslcipher_binding resources configured on Citrix ADC. Filtering can be done on any of the properties of the resource.

pagination

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/sslservicegroup_sslcipher_binding/servicegroupname_value<String>?**pagesize=#no;pageno=#no**

Use this query-parameter to get the sslservicegroup_sslcipher_binding resources in chunks.

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "sslservicegroup_sslcipher_binding": [ {

"ciphername":<String_value>,

"cipheraliasname":<String_value>,

"servicegroupname":<String_value>,

"description":<String_value>

}]}
```

get (all)

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/sslservicegroup_sslcipher_binding

Query-parameters:

bulkbindings

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/sslservicegroup_sslcipher_binding?**bulkbindings=yes**

NITRO allows you to fetch bindings in bulk.

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "sslservicegroup_sslcipher_binding": [ {

"ciphername":<String_value>,

"cipheraliasname":<String_value>,

"servicegroupname":<String_value>,

"description":<String_value>

}]}
```

count

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/sslservicegroup_sslcipher_binding/servicegroupname_value<String>?**count=yes**

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{"sslservicegroup_sslcipher_binding": [ { "__count": "#no"} ] }
```

sslservice_binding

Binding object which returns the resources bound to sslservice.

Properties

(click to see Operations)

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
servicename	<String>	Read-write	Name of the SSL service for which to show detailed information. Minimum length = 1
sslservice_ecccurve_binding	<sslservice_ecccurve_binding[]>	Read-only	ecccurve that can be bound to sslservice.
sslservice_sslpolicy_binding	<sslservice_sslpolicy_binding[]>	Read-only	sslpolicy that can be bound to sslservice.
sslservice_sslcipher_binding	<sslservice_sslcipher_binding[]>	Read-only	sslcipher that can be bound to sslservice.
sslservice_sslcertkey_binding	<sslservice_sslcertkey_binding[]>	Read-only	sslcertkey that can be bound to sslservice.

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
sslservice_sslciphersuite_binding	<sslservice_sslciphersuite_binding[]>	Read-only	sslciphersuite that can be bound to sslservice.

Operations

(click to see Properties)

GET| GET (ALL)

Some options that you can use for each operations:

- **Getting warnings in response:**NITRO allows you to get warnings in an operation by specifying the "warning" query parameter as "yes". For example, to get warnings while connecting to the Citrix ADC appliance, the URL is as follows:

  http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/login?warning=yes

  If any, the warnings are displayed in the response payload with the HTTP code "209 X-NITRO-WARNING".

- **Authenticated access for individual NITRO operations:**NITRO allows you to logon to the Citrix ADC appliance to perform individual operations. You can use this option instead of creating a NITRO session (using the login object) and then using that session to perform all operations,

  To do this, you must specify the username and password in the request header of the NITRO request as follows:

  X-NITRO-USER:<username>

  X-NITRO-PASS:<password>

  Note: In such cases, make sure that the request header DOES not include the following:

  Cookie:NITRO_AUTH_TOKEN=<tokenvalue>

`*Note:`*

Mandatory parameters are marked in redand placeholder content is marked in <green>.

get

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/sslservice_binding/servicename_value<String>

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "sslservice_binding": [ {

"servicename":<String_value>,

"sslservice_sslcipher_binding":<sslservice_sslcipher_binding[]_value>,

"sslservice_ecccurve_binding":<sslservice_ecccurve_binding[]_value>,

"sslservice_sslciphersuite_binding":<sslservice_sslciphersuite_binding[]_value>,

"sslservice_sslpolicy_binding":<sslservice_sslpolicy_binding[]_value>,

"sslservice_sslcertkey_binding":<sslservice_sslcertkey_binding[]_value>

}]}
```

get (all)

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/sslservice_binding

Query-parameters:

bulkbindings

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/sslservice_binding?**bulkbindings=yes**

NITRO allows you to fetch bindings in bulk.

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "sslservice_binding": [ {

"servicename":<String_value>,

"sslservice_sslcipher_binding":<sslservice_sslcipher_binding[]_value>,

"sslservice_ecccurve_binding":<sslservice_ecccurve_binding[]_value>,

"sslservice_sslciphersuite_binding":<sslservice_sslciphersuite_binding[]_v
alue>,

"sslservice_sslpolicy_binding":<sslservice_sslpolicy_binding[]_value>,

"sslservice_sslcertkey_binding":<sslservice_sslcertkey_binding[]_value>

}]}
```

sslservice_ecccurve_binding

Binding object showing the ecccurve that can be bound to sslservice.

Properties

(click to see Operations)

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
ecccurvename	<String>	Read-write	Named ECC curve bound to service/ vserver. Possible values = ALL, P_224, P_256, P_384, P_521
servicename	<String>	Read-write	Name of the SSL service for which to set advanced configuration. Minimum length = 1
__count	<Double>	Read-write	count parameter

Operations

(click to see Properties)

ADD:| DELETE:| GET| GET (ALL)| COUNT

Some options that you can use for each operations:

- **Getting warnings in response:**NITRO allows you to get warnings in an operation by specifying the "warning" query parameter as "yes". For example, to get warnings while connecting to the Citrix ADC appliance, the URL is as follows:

  http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/login?warning=yes

  If any, the warnings are displayed in the response payload with the HTTP code "209 X-NITRO-WARNING".

- **Authenticated access for individual NITRO operations:**NITRO allows you to logon to the Citrix ADC appliance to perform individual operations. You can use this option instead of

creating a NITRO session (using the login object) and then using that session to perform all operations,

To do this, you must specify the username and password in the request header of the NITRO request as follows:

X-NITRO-USER:<username>

X-NITRO-PASS:<password>

Note: In such cases, make sure that the request header DOES not include the following:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue>

`*Note:`*

Mandatory parameters are marked in redand placeholder content is marked in <green>.

add

URL:http://<Citrix-ADC-IP-address(NSIP)/nitro/v1/config/sslservice_ecccurve_binding

HTTP Method:PUT

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Content-Type:application/json

Request Payload:

```
{

"sslservice_ecccurve_binding":{

<b>"servicename":<String_value>,

</b>"ecccurvename":<String_value>

}}
```

Response:

HTTP Status Code on Success: 201 Created HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error

delete

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/sslservice_ecccurve_binding/servicename_value<String>

Query-parameters:

args

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/sslservice_ecccurve_binding/servicename_value<String>?**args=ecccurvename:<String_value>**

HTTP Method:DELETE

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue>

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error

get

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/sslservice_ecccurve_binding/servicename_value<String>

Query-parameters:

filter

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/sslservice_ecccurve_binding/servicename_value<String>?**filter=property-name1:property-value1,property-name2:property-value2**

Use this query-parameter to get the filtered set of sslservice_ecccurve_binding resources configured on Citrix ADC. Filtering can be done on any of the properties of the resource.

pagination

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/sslservice_ecccurve_binding/servicename_value<String>?**pagesize=#no;pageno=#no**

Use this query-parameter to get the sslservice_ecccurve_binding resources in chunks.

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "sslservice_ecccurve_binding": [ {

"ecccurvename":<String_value>,

"servicename":<String_value>

}]}
```

get (all)

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/sslservice_ecccurve_binding

Query-parameters:

bulkbindings

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/sslservice_ecccurve_binding?**bulkbindings=yes**

NITRO allows you to fetch bindings in bulk.

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "sslservice_ecccurve_binding": [ {

"ecccurvename":<String_value>,

"servicename":<String_value>

}]}
```

count

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/sslservice_ecccurve_binding/servicename_value<String>?**count=yes**

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{"sslservice_ecccurve_binding": [ { "__count": "#no"} ] }
```

sslservice_sslcertkey_binding

Binding object showing the sslcertkey that can be bound to sslservice.

Properties

(click to see Operations)

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
ca	<Boolean>	Read-write	CA certificate.
crlcheck	<String>	Read-write	The state of the CRL check parameter. (Mandatory/ Optional). Possible values = Mandatory, Optional
servicename	<String>	Read-write	Name of the SSL service for which to set advanced configuration. Minimum length = 1
certkeyname	<String>	Read-write	The certificate key pair binding.
skipcaname	<Boolean>	Read-write	The flag is used to indicate whether this particular CA certificate's CA_Name needs to be sent to the SSL

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
			client while requesting for client certificate in a SSL handshake.
snicert	<Boolean>	Read-write	The name of the CertKey. Use this option to bind Certkey(s) which will be used in SNI processing.
ocspcheck	<String>	Read-write	Rule to use for the OCSP responder associated with the CA certificate during client authentication. If MANDATORY is specified, deny all SSL clients if the OCSP check fails because of connectivity issues with the remote OCSP server, or any other reason that prevents the OCSP check. With the OPTIONAL setting, allow SSL clients even if the OCSP check fails except when the client certificate is revoked. Possible values = Mandatory, Optional
cleartextport	<Integer>	Read-only	The clearTextPort settings. Minimum value = 0

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
			Maximum value = 65534 Range 1 - 65535 * in CLI is represented as 65535 in NITRO API
__count	<Double>	Read-only	count parameter

Operations

(click to see Properties)

ADD:| DELETE:| GET| GET (ALL)| COUNT

Some options that you can use for each operations:

- **Getting warnings in response:**NITRO allows you to get warnings in an operation by specifying the "warning" query parameter as "yes". For example, to get warnings while connecting to the Citrix ADC appliance, the URL is as follows:

  http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/login?warning=yes

  If any, the warnings are displayed in the response payload with the HTTP code "209 X-NITRO-WARNING".

- **Authenticated access for individual NITRO operations:**NITRO allows you to logon to the Citrix ADC appliance to perform individual operations. You can use this option instead of creating a NITRO session (using the login object) and then using that session to perform all operations,

  To do this, you must specify the username and password in the request header of the NITRO request as follows:

  X-NITRO-USER:<username>

  X-NITRO-PASS:<password>

  Note: In such cases, make sure that the request header DOES not include the following:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue>

`*Note:` *

Mandatory parameters are marked in redand placeholder content is marked in <green>.

add

URL:http://<Citrix-ADC-IP-address(NSIP)/nitro/v1/config/sslservice_sslcertkey_binding

HTTP Method:PUT

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Content-Type:application/json

Request Payload:

```
{
"sslservice_sslcertkey_binding":{
<b>"servicename":<String_value>,
</b>"certkeyname":<String_value>,
"ca":<Boolean_value>,
"crlcheck":<String_value>,
"skipcaname":<Boolean_value>,
"snicert":<Boolean_value>,
"ocspcheck":<String_value>
}}
```

Response:

HTTP Status Code on Success: 201 Created HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error

delete

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/sslservice_sslcertkey_binding/servicename_value<String>

Query-parameters:

args

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/sslservice_sslcertkey_binding/servicename_value<String>?**args=certkeyname:<String_value>,ca:<Boolean_value>,crlcheck:<String_value>,snicert:<Boolean_valu**

HTTP Method:DELETE

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue>

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error

get

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/sslservice_sslcertkey_binding/servicename_value<String>

Query-parameters:

filter

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/sslservice_sslcertkey_binding/servicename_value<String>?**filter=property-name1:property-value1,property-name2:property-value2**

Use this query-parameter to get the filtered set of sslservice_sslcertkey_binding resources configured on Citrix ADC. Filtering can be done on any of the properties of the resource.

pagination

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/sslservice_sslcertkey_binding/servicename_value<String>?**pagesize=#no;pageno=#no**

Use this query-parameter to get the sslservice_sslcertkey_binding resources in chunks.

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "sslservice_sslcertkey_binding": [ {

"ca":<Boolean_value>,

"crlcheck":<String_value>,

"servicename":<String_value>,

"certkeyname":<String_value>,

"skipcaname":<Boolean_value>,

"snicert":<Boolean_value>,

"ocspcheck":<String_value>,

"cleartextport":<Integer_value>

}]}
```

get (all)

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/sslservice_sslcertkey_binding

Query-parameters:

bulkbindings

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/sslservice_sslcertkey_binding?**bulkbindings=yes**

NITRO allows you to fetch bindings in bulk.

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "sslservice_sslcertkey_binding": [ {

"ca":<Boolean_value>,

"crlcheck":<String_value>,

"servicename":<String_value>,

"certkeyname":<String_value>,

"skipcaname":<Boolean_value>,
```

```
"snicert":<Boolean_value>,

"ocspcheck":<String_value>,

"cleartextport":<Integer_value>

}]}
```

count

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/sslservice_sslcertkey_binding/servicename_value<String>?**count=yes**

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{"sslservice_sslcertkey_binding": [ { "__count": "#no"} ] }
```

sslservice_sslciphersuite_binding

Binding object showing the sslciphersuite that can be bound to sslservice.

Properties

(click to see Operations)

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
ciphername	<String>	Read-write	The cipher group/ alias/individual cipher configuration.
servicename	<String>	Read-write	Name of the SSL service for which to set advanced configuration. Minimum length = 1
description	<String>	Read-write	The cipher suite description.
__count	<Double>	Read-write	count parameter

Operations

(click to see Properties)

ADD:| DELETE:| GET| GET (ALL)| COUNT

Some options that you can use for each operations:

- **Getting warnings in response:**NITRO allows you to get warnings in an operation by specifying the "warning" query parameter as "yes". For example, to get warnings while connecting to the Citrix ADC appliance, the URL is as follows:

  http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/login?warning=yes

  If any, the warnings are displayed in the response payload with the HTTP code "209 X-NITRO-WARNING".

- **Authenticated access for individual NITRO operations:**NITRO allows you to logon to the Citrix ADC appliance to perform individual operations. You can use this option instead of creating a NITRO session (using the login object) and then using that session to perform all operations,

  To do this, you must specify the username and password in the request header of the NITRO request as follows:

  X-NITRO-USER:<username>

X-NITRO-PASS:<password>

Note: In such cases, make sure that the request header DOES not include the following:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue>

`*Note:`*

Mandatory parameters are marked in redand placeholder content is marked in <green>.

add

URL:http://<Citrix-ADC-IP-address(NSIP)/nitro/v1/config/sslservice_sslciphersuite_binding

HTTP Method:PUT

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Content-Type:application/json

Request Payload:

```
{

"sslservice_sslciphersuite_binding":{

<b>"servicename":<String_value>,

</b>"ciphername":<String_value>

}}
```

Response:

HTTP Status Code on Success: 201 Created HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error

delete

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/sslservice_sslciphersuite_binding/servicename_value<String>

Query-parameters:

args

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/sslservice_sslciphersuite_binding/servicename_value<String>?**args=ciphername:<String_value>**

HTTP Method:DELETE

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue>

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error

get

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/sslservice_sslciphersuite_binding/servicename_value<String>

Query-parameters:

filter

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/sslservice_sslciphersuite_binding/servicename_value<String>?**filter=property-name1:property-value1,property-name2:property-value2**

Use this query-parameter to get the filtered set of sslservice_sslciphersuite_binding resources configured on Citrix ADC. Filtering can be done on any of the properties of the resource.

pagination

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/sslservice_sslciphersuite_binding/servicename_value<String>?**pagesize=#no;pageno=#no**

Use this query-parameter to get the sslservice_sslciphersuite_binding resources in chunks.

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "sslservice_sslciphersuite_binding": [ {

"ciphername":<String_value>,

"servicename":<String_value>,

"description":<String_value>

}]}
```

get (all)

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/sslservice_sslciphersuite_binding

Query-parameters:

bulkbindings

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/sslservice_sslciphersuite_binding?**bulkbindings=yes**

NITRO allows you to fetch bindings in bulk.

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "sslservice_sslciphersuite_binding": [ {

"ciphername":<String_value>,

"servicename":<String_value>,

"description":<String_value>

}]}
```

count

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/sslservice_sslciphersuite_binding/servicename_value<String>?**count=yes**

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{"sslservice_sslciphersuite_binding": [ { "__count": "#no"} ] }
```

sslservice_sslcipher_binding

Binding object showing the sslcipher that can be bound to sslservice.

Properties

(click to see Operations)

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
ciphername	<String>	Read-write	Name of the individual cipher, user-defined cipher group, or predefined (built-in) cipher alias.
cipheraliasname	<String>	Read-write	The cipher group/ alias/individual cipher configuration.
servicename	<String>	Read-write	Name of the SSL service for which to set advanced configuration. Minimum length = 1
description	<String>	Read-write	The cipher suite description.
__count	<Double>	Read-write	count parameter

Operations

(click to see Properties)

ADD:| DELETE:| GET| GET (ALL)| COUNT

Some options that you can use for each operations:

- **Getting warnings in response:**NITRO allows you to get warnings in an operation by specifying the "warning" query parameter as "yes". For example, to get warnings while connecting to the Citrix ADC appliance, the URL is as follows:

  http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/login?warning=yes

  If any, the warnings are displayed in the response payload with the HTTP code "209 X-NITRO-WARNING".

- **Authenticated access for individual NITRO operations:**NITRO allows you to logon to the Citrix ADC appliance to perform individual operations. You can use this option instead of creating a NITRO session (using the login object) and then using that session to perform all operations,

  To do this, you must specify the username and password in the request header of the NITRO request as follows:

  X-NITRO-USER:<username>

  X-NITRO-PASS:<password>

  Note: In such cases, make sure that the request header DOES not include the following:

  Cookie:NITRO_AUTH_TOKEN=<tokenvalue>

`*Note:`*

Mandatory parameters are marked in redand placeholder content is marked in <green>.

add

URL:http://<Citrix-ADC-IP-address(NSIP)/nitro/v1/config/sslservice_sslcipher_binding

HTTP Method:PUT

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Content-Type:application/json

Request Payload:

```
{
```

```
"sslservice_sslcipher_binding":{

<b>"servicename":<String_value>,

</b>"ciphername":<String_value>

}}
```

Response:

HTTP Status Code on Success: 201 Created HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error

delete

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/sslservice_sslcipher_binding/servicename_value<String>

Query-parameters:

args

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/sslservice_sslcipher_binding/servicename_value<String>?**args=ciphername:<String_value>**

HTTP Method:DELETE

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue>

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error

get

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/sslservice_sslcipher_binding/servicename_value<String>

Query-parameters:

filter

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/sslservice_sslcipher_binding/servicename_value<String>?**filter=property-name1:property-value1,property-name2:property-value2**

Use this query-parameter to get the filtered set of sslservice_sslcipher_binding resources configured on Citrix ADC. Filtering can be done on any of the properties of the resource.

pagination

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/sslservice_sslcipher_binding/servicename_value<String>?**pagesize=#no;pageno=#no**

Use this query-parameter to get the sslservice_sslcipher_binding resources in chunks.

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "sslservice_sslcipher_binding": [ {

"ciphername":<String_value>,

"cipheraliasname":<String_value>,

"servicename":<String_value>,
```

```
"description":<String_value>

}]}
```

get (all)

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/sslservice_sslcipher_binding

Query-parameters:

bulkbindings

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/sslservice_sslcipher_binding?**bulkbindings=yes**

NITRO allows you to fetch bindings in bulk.

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "sslservice_sslcipher_binding": [ {

"ciphername":<String_value>,

"cipheraliasname":<String_value>,

"servicename":<String_value>,
```

```
"description":<String_value>

}]}
```

count

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/sslservice_sslcipher_binding/servicename_value<String>?**count=yes**

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{"sslservice_sslcipher_binding": [ { "__count": "#no"} ] }
```

sslservice_sslpolicy_binding

Binding object showing the sslpolicy that can be bound to sslservice.

Properties

(click to see Operations)

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
priority	<Double>	Read-write	The priority of the policies bound to this SSL service. Minimum value = 0 Maximum value = 65534
policyname	<String>	Read-write	The SSL policy binding.
labelname	<String>	Read-write	Name of the label to invoke if the current policy rule evaluates to TRUE.
servicename	<String>	Read-write	Name of the SSL service for which to set advanced configuration. Minimum length = 1
gotopriorityexpression	<String>	Read-write	Expression specifying the priority of the next policy which will get evaluated if the current policy rule evaluates to TRUE.
invoke	<Boolean>	Read-write	Invoke flag. This attribute is relevant only for ADVANCED policies.
labeltype	<String>	Read-write	Type of policy label invocation. Possible values = vserver, service, policylabel

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
polinherit	<Double>	Read-only	Whether the bound policy is a inherited policy or not. Minimum value = 0 Maximum value = 254
__count	<Double>	Read-only	count parameter

Operations

(click to see Properties)

ADD:| DELETE:| GET| GET (ALL)| COUNT

Some options that you can use for each operations:

- **Getting warnings in response:**NITRO allows you to get warnings in an operation by specifying the "warning" query parameter as "yes". For example, to get warnings while connecting to the Citrix ADC appliance, the URL is as follows:

  http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/login?warning=yes

  If any, the warnings are displayed in the response payload with the HTTP code "209 X-NITRO-WARNING".
- **Authenticated access for individual NITRO operations:**NITRO allows you to logon to the Citrix ADC appliance to perform individual operations. You can use this option instead of creating a NITRO session (using the login object) and then using that session to perform all operations,

  To do this, you must specify the username and password in the request header of the NITRO request as follows:

  X-NITRO-USER:<username>

  X-NITRO-PASS:<password>

  Note: In such cases, make sure that the request header DOES not include the following:

  Cookie:NITRO_AUTH_TOKEN=<tokenvalue>

`*Note:` *

Mandatory parameters are marked in redand placeholder content is marked in <green>.

add

URL:http://<Citrix-ADC-IP-address(NSIP)/nitro/v1/config/sslservice_sslpolicy_binding

HTTP Method:PUT

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Content-Type:application/json

Request Payload:

```
{

"sslservice_sslpolicy_binding":{

<b>"servicename":<String_value>,

</b>"policyname":<String_value>,

"priority":<Double_value>,

"gotopriorityexpression":<String_value>,

"invoke":<Boolean_value>,

"labeltype":<String_value>,

"labelname":<String_value>

}}
```

Response:

HTTP Status Code on Success: 201 Created HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error

delete

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/sslservice_sslpolicy_binding/servicename_value<String>

Query-parameters:

args

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/sslservice_sslpolicy_binding/servicename_value<String>?**args=policyname:<String_value>,priority:<Double_value>**

HTTP Method:DELETE

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue>

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error

get

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/sslservice_sslpolicy_binding/servicename_value<String>

Query-parameters:

filter

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/sslservice_sslpolicy_binding/servicename_value<String>?**filter=property-name1:property-value1,property-name2:property-value2**

Use this query-parameter to get the filtered set of sslservice_sslpolicy_binding resources configured on Citrix ADC. Filtering can be done on any of the properties of the resource.

pagination

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/sslservice_sslpolicy_binding/servicename_value<String>?**pagesize=#no;pageno=#no**

Use this query-parameter to get the sslservice_sslpolicy_binding resources in chunks.

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "sslservice_sslpolicy_binding": [ {

"priority":<Double_value>,

"policyname":<String_value>,

"labelname":<String_value>,

"servicename":<String_value>,

"gotopriorityexpression":<String_value>,

"invoke":<Boolean_value>,

"labeltype":<String_value>,

"polinherit":<Double_value>

}]}
```

get (all)

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/sslservice_sslpolicy_binding

Query-parameters:

bulkbindings

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/sslservice_sslpolicy_binding?**bulkbindings=yes**

NITRO allows you to fetch bindings in bulk.

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "sslservice_sslpolicy_binding": [ {

"priority":<Double_value>,

"policyname":<String_value>,

"labelname":<String_value>,

"servicename":<String_value>,

"gotopriorityexpression":<String_value>,

"invoke":<Boolean_value>,

"labeltype":<String_value>,

"polinherit":<Double_value>

}]}
```

count

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/sslservice_sslpolicy_binding/servicename_value<String>?**count=yes**

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{"sslservice_sslpolicy_binding": [ { "__count": "#no"} ] }
```

sslvserver

Configuration for SSL virtual server resource.

Properties

(click to see Operations)

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
vservername	<String>	Read-write	Name of the SSL virtual server for which to set advanced configuration.

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
			Minimum length = 1
cleartextport	<Integer>	Read-write	Port on which clear-text data is sent by the appliance to the server. Do not specify this parameter for SSL offloading with end-to-end encryption. Default value: 0 Minimum value = 0 Maximum value = 65534
dh	<String>	Read-write	State of Diffie-Hellman (DH) key exchange. Default value: DISABLED Possible values = ENABLED, DISABLED
dhfile	<String>	Read-write	Name of and, optionally, path to the DH parameter file, in PEM format, to be installed. /nsconfig/ssl/ is the default path. Minimum length = 1
dhcount	<Double>	Read-write	Number of interactions, between the client and the Citrix ADC, after which the DH

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
			private-public pair is regenerated. A value of zero (0) specifies infinite use (no refresh). Minimum value = 0 Maximum value = 65534
dhkeyexpsizelimit	<String>	Read-write	This option enables the use of NIST recommended (NIST Special Publication 800-56A) bit size for private-key size. For example, for DH params of size 2048bit, the private-key size recommended is 224bits. This is rounded-up to 256bits. Default value: DISABLED Possible values = ENABLED, DISABLED
ersa	<String>	Read-write	State of Ephemeral RSA (eRSA) key exchange. Ephemeral RSA allows clients that support only export ciphers to communicate with the secure server even if the server

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
			certificate does not support export clients. The ephemeral RSA key is automatically generated when you bind an export cipher to an SSL or TCP-based SSL virtual server or service. When you remove the export cipher, the eRSA key is not deleted. It is reused at a later date when another export cipher is bound to an SSL or TCP-based SSL virtual server or service. The eRSA key is deleted when the appliance restarts. Default value: ENABLED Possible values = ENABLED, DISABLED
ersacount	<Double>	Read-write	Refresh count for regeneration of the RSA public-key and private-key pair. Zero (0) specifies infinite usage (no refresh). Minimum value = 0 Maximum value = 65534

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
sessreuse	<String>	Read-write	State of session reuse. Establishing the initial handshake requires CPU-intensive public key encryption operations. With the ENABLED setting, session key exchange is avoided for session resumption requests received from the client. Default value: ENABLED Possible values = ENABLED, DISABLED
sesstimeout	<Double>	Read-write	Time, in seconds, for which to keep the session active. Any session resumption request received after the timeout period will require a fresh SSL handshake and establishment of a new SSL session. Default value: 120 Minimum value = 0 Maximum value = 4294967294

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
cipherredirect	<String>	Read-write	State of Cipher Redirect. If cipher redirect is enabled, you can configure an SSL virtual server or service to display meaningful error messages if the SSL handshake fails because of a cipher mismatch between the virtual server or service and the client. Default value: DISABLED Possible values = ENABLED, DISABLED
cipherurl	<String>	Read-write	The redirect URL to be used with the Cipher Redirect feature.
sslv2redirect	<String>	Read-write	State of SSLv2 Redirect. If SSLv2 redirect is enabled, you can configure an SSL virtual server or service to display meaningful error messages if the SSL handshake fails because of a protocol version mismatch between the virtual server or

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
			service and the client. Default value: DISABLED Possible values = ENABLED, DISABLED
sslv2url	<String>	Read-write	URL of the page to which to redirect the client in case of a protocol version mismatch. Typically, this page has a clear explanation of the error or an alternative location that the transaction can continue from.
clientauth	<String>	Read-write	State of client authentication. If client authentication is enabled, the virtual server terminates the SSL handshake if the SSL client does not provide a valid certificate. Default value: DISABLED Possible values = ENABLED, DISABLED
clientcert	<String>	Read-write	Type of client authentication. If this parameter is set to MANDATORY, the

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
			appliance terminates the SSL handshake if the SSL client does not provide a valid certificate. With the OPTIONAL setting, the appliance requests a certificate from the SSL clients but proceeds with the SSL transaction even if the client presents an invalid certificate. Caution: Define proper access control policies before changing this setting to Optional. Possible values = Mandatory, Optional
sslredirect	<String>	Read-write	State of HTTPS redirects for the SSL virtual server. For an SSL session, if the client browser receives a redirect message, the browser tries to connect to the new location. However, the secure SSL session breaks if the object has moved from a secure site (https://) to an unsecure site

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
			(http://). Typically, a warning message appears on the screen, prompting the user to continue or disconnect. If SSL Redirect is ENABLED, the redirect message is automatically converted from http:// to https:// and the SSL session does not break. Default value: DISABLED Possible values = ENABLED, DISABLED
redirectportrewrite	<String>	Read-write	State of the port rewrite while performing HTTPS redirect. If this parameter is ENABLED and the URL from the server does not contain the standard port, the port is rewritten to the standard. Default value: DISABLED

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
			Possible values = ENABLED, DISABLED
ssl2	<String>	Read-write	State of SSLv2 protocol support for the SSL Virtual Server. Default value: DISABLED Possible values = ENABLED, DISABLED
ssl3	<String>	Read-write	State of SSLv3 protocol support for the SSL Virtual Server. Default value: ENABLED Possible values = ENABLED, DISABLED
tls1	<String>	Read-write	State of TLSv1.0 protocol support for the SSL Virtual Server. Default value: ENABLED Possible values = ENABLED, DISABLED
tls11	<String>	Read-write	State of TLSv1.1 protocol support for

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
			the SSL Virtual Server. Default value: ENABLED Possible values = ENABLED, DISABLED
tls12	<String>	Read-write	State of TLSv1.2 protocol support for the SSL Virtual Server. Default value: ENABLED Possible values = ENABLED, DISABLED
tls13	<String>	Read-write	State of TLSv1.3 protocol support for the SSL Virtual Server. Default value: DISABLED Possible values = ENABLED, DISABLED
snienable	<String>	Read-write	State of the Server Name Indication (SNI) feature on the virtual server and service-based offload. SNI helps to enable SSL encryption on multiple domains on

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
			a single virtual server or service if the domains are controlled by the same organization and share the same second-level domain name. For example, *.sports.net can be used to secure domains such as login.sports.net and help.sports.net. Default value: DISABLED Possible values = ENABLED, DISABLED
ocspstapling	<String>	Read-write	State of OCSP stapling support on the SSL virtual server. Supported only if the protocol used is higher than SSLv3. Possible values: ENABLED: The appliance sends a request to the OCSP responder to check the status of the server certificate and caches the response for the specified time. If the response is valid at the time of SSL

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
			handshake with the client, the OCSP-based server certificate status is sent to the client during the handshake. DISABLED: The appliance does not check the status of the server certificate. . Default value: DISABLED Possible values = ENABLED, DISABLED
pushenctrigger	<String>	Read-write	Trigger encryption on the basis of the PUSH flag value. Available settings function as follows: * ALWAYS - Any PUSH packet triggers encryption. * IGNORE - Ignore PUSH packet for triggering encryption. * MERGE - For a consecutive sequence of PUSH packets, the last

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
			PUSH packet triggers encryption. * TIMER - PUSH packet triggering encryption is delayed by the time defined in the set ssl parameter command or in the Change Advanced SSL Settings dialog box. Possible values = Always, Merge, Ignore, Timer
sendclosenotify	<String>	Read-write	Enable sending SSL Close-Notify at the end of a transaction. Default value: YES Possible values = YES, NO
dtlsprofilename	<String>	Read-write	Name of the DTLS profile whose settings are to be applied to the virtual server. Minimum length = 1 Maximum length = 127
sslprofile	<String>	Read-write	Name of the SSL profile that contains

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
			SSL settings for the virtual server. Minimum length = 1 Maximum length = 127
hsts	<String>	Read-write	State of HSTS protocol support for the SSL Virtual Server. Using HSTS, a server can enforce the use of an HTTPS connection for all communication with a client. Default value: DISABLED Possible values = ENABLED, DISABLED
maxage	<Double>	Read-write	Set the maximum time, in seconds, in the strict transport security (STS) header during which the client must send only HTTPS requests to the server. Default value: 0 Minimum value = 0 Maximum value = 4294967294
includesubdomains	<String>	Read-write	Enable HSTS for subdomains. If set to

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
			Yes, a client must send only HTTPS requests for subdomains. Default value: NO Possible values = YES, NO
preload	<String>	Read-write	Flag indicates the consent of the site owner to have their domain preloaded. Default value: NO Possible values = YES, NO
strictsigdigestcheck	<String>	Read-write	Parameter indicating to check whether peer entity certificate during TLS1.2 handshake is signed with one of signature-hash combination supported by Citrix ADC. Default value: DISABLED Possible values = ENABLED, DISABLED
zerorttearlydata	<String>	Read-write	State of TLS 1.3 0-RTT early data support for the SSL Virtual Server. This setting only has an effect if resumption is

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
			enabled, as early data cannot be sent along with an initial handshake. Early application data has significantly different security properties - in particular there is no guarantee that the data cannot be replayed. Default value: DISABLED Possible values = ENABLED, DISABLED
tls13sessionticketsperauthcontext	<Double>	Read-write	Number of tickets the SSL Virtual Server will issue anytime TLS 1.3 is negotiated, ticket-based resumption is enabled, and either (1) a handshake completes or (2) post-handhsake client auth completes. This value can be increased to enable clients to open multiple parallel connections using a

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
			fresh ticket for each connection. No tickets are sent if resumption is disabled. Default value: 1 Minimum value = 1 Maximum value = 10
dhekeyexchangewithpsk	<String>	Read-write	Whether or not the SSL Virtual Server will require a DHE key exchange to occur when a PSK is accepted during a TLS 1.3 resumption handshake. A DHE key exchange ensures forward secrecy even in the event that ticket keys are compromised, at the expense of an additional round trip and resources required to carry out the DHE key exchange. If disabled, a DHE key exchange will be performed when a PSK is accepted

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
			but only if requested by the client. If enabled, the server will require a DHE key exchange when a PSK is accepted regardless of whether the client supports combined PSK-DHE key exchange. This setting only has an effect when resumption is enabled. Default value: NO Possible values = YES, NO
crlcheck	<String>	Read-only	The state of the CRL check parameter. (Mandatory/ Optional). Possible values = Mandatory, Optional
nonfipsciphers	<String>	Read-only	The state of usage of non FIPS approved ciphers. Default value: DISABLED Possible values = ENABLED, DISABLED

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
service	<Double>	Read-only	Service. Minimum value = 0 Maximum value = 65534
ocspcheck	<String>	Read-only	The state of the OCSP check parameter. (Mandatory/ Optional). Possible values = Mandatory, Optional
ca	<Boolean>	Read-only	CA certificate.
snicert	<Boolean>	Read-only	The name of the CertKey. Use this option to bind Certkey(s) which will be used in SNI processing.
skipcaname	<Boolean>	Read-only	The flag is used to indicate whether this particular CA certificate's CA_Name needs to be sent to the SSL client while requesting for client certificate in a SSL handshake.
dtlsflag	<Boolean>	Read-only	The flag is used to indicate whether DTLS is set or not.
__count	<Double>	Read-only	count parameter

Operations

(click to see Properties)

UPDATE| UNSET| GET (ALL)| GET| COUNT

Some options that you can use for each operations:

- **Getting warnings in response:**NITRO allows you to get warnings in an operation by specifying the "warning" query parameter as "yes". For example, to get warnings while connecting to the Citrix ADC appliance, the URL is as follows:

  http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/login?warning=yes

  If any, the warnings are displayed in the response payload with the HTTP code "209 X-NITRO-WARNING".

- **Authenticated access for individual NITRO operations:**NITRO allows you to logon to the Citrix ADC appliance to perform individual operations. You can use this option instead of creating a NITRO session (using the login object) and then using that session to perform all operations,

  To do this, you must specify the username and password in the request header of the NITRO request as follows:

  X-NITRO-USER:<username>

  X-NITRO-PASS:<password>

  Note: In such cases, make sure that the request header DOES not include the following:

  Cookie:NITRO_AUTH_TOKEN=<tokenvalue>

`*Note:`*

Mandatory parameters are marked in redand placeholder content is marked in <green>.

update

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/sslvserver

HTTP Method:PUT

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Content-Type:application/json

Request Payload:

```
{"sslvserver":{

<b>"vservername":<String_value>,

</b>"cleartextport":<Integer_value>,

"dh":<String_value>,

"dhfile":<String_value>,

"dhcount":<Double_value>,

"dhkeyexpsizelimit":<String_value>,

"ersa":<String_value>,

"ersacount":<Double_value>,

"sessreuse":<String_value>,

"sesstimeout":<Double_value>,

"cipherredirect":<String_value>,

"cipherurl":<String_value>,

"sslv2redirect":<String_value>,

"sslv2url":<String_value>,

"clientauth":<String_value>,

"clientcert":<String_value>,

"sslredirect":<String_value>,

"redirectportrewrite":<String_value>,

"ssl2":<String_value>,
```

```
"ssl3":<String_value>,

"tls1":<String_value>,

"tls11":<String_value>,

"tls12":<String_value>,

"tls13":<String_value>,

"snienable":<String_value>,

"ocspstapling":<String_value>,

"pushenctrigger":<String_value>,

"sendclosenotify":<String_value>,

"dtlsprofilename":<String_value>,

"sslprofile":<String_value>,

"hsts":<String_value>,

"maxage":<Double_value>,

"includesubdomains":<String_value>,

"preload":<String_value>,

"strictsigdigestcheck":<String_value>,

"zerorttearlydata":<String_value>,

"tls13sessionticketsperauthcontext":<Double_value>,

"dhekeyexchangewithpsk":<String_value>

}}
```

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error

unset

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/sslvserver?**action=unset**

HTTP Method:POST

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Content-Type:application/json

Request Payload:

```
{"sslvserver":{

<b>"vservername":<String_value>,

</b>"cleartextport":true,

"dh":true,

"dhfile":true,

"dhcount":true,

"dhkeyexpsizelimit":true,

"ersa":true,

"ersacount":true,

"sessreuse":true,

"sesstimeout":true,

"cipherredirect":true,

"cipherurl":true,
```

```
"sslv2redirect":true,

"sslv2url":true,

"clientauth":true,

"clientcert":true,

"sslredirect":true,

"redirectportrewrite":true,

"ssl2":true,

"ssl3":true,

"tls1":true,

"tls11":true,

"tls12":true,

"tls13":true,

"snienable":true,

"ocspstapling":true,

"sendclosenotify":true,

"dtlsprofilename":true,

"sslprofile":true,

"hsts":true,

"maxage":true,

"includesubdomains":true,

"preload":true,

"strictsigdigestcheck":true,
```

```
"zerorttearlydata":true,

"tls13sessionticketsperauthcontext":true,

"dhekeyexchangewithpsk":true

}}
```

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error

get (all)

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/sslvserver

Query-parameters:

attrs

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/sslvserver?**attrs=property-name1,property-name2**

Use this query parameter to specify the resource details that you want to retrieve.

filter

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/sslvserver?**filter=property-name1:property-val1,property-name2:property-val2**

Use this query-parameter to get the filtered set of sslvserver resources configured on Citrix ADC. Filtering can be done on any of the properties of the resource.

view

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/sslvserver?**view=summary**

Note: By default, the retrieved results are displayed in detail view (?view=detail).

pagination

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/sslvserver?**pagesize=#no;pageno=#no**

Use this query-parameter to get the sslvserver resources in chunks.

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "sslvserver": [ {

"vservername":<String_value>,

"cleartextport":<Integer_value>,

"dh":<String_value>,

"dhfile":<String_value>,

"dhcount":<Double_value>,

"dhkeyexpsizelimit":<String_value>,

"ersa":<String_value>,

"ersacount":<Double_value>,

"sessreuse":<String_value>,

"sesstimeout":<Double_value>,
```

```
"cipherredirect":<String_value>,

"crlcheck":<String_value>,

"cipherurl":<String_value>,

"sslv2redirect":<String_value>,

"sslv2url":<String_value>,

"clientauth":<String_value>,

"clientcert":<String_value>,

"sslredirect":<String_value>,

"redirectportrewrite":<String_value>,

"nonfipsciphers":<String_value>,

"ssl2":<String_value>,

"ssl3":<String_value>,

"tls1":<String_value>,

"tls11":<String_value>,

"tls12":<String_value>,

"tls13":<String_value>,

"snienable":<String_value>,

"ocspstapling":<String_value>,

"service":<Double_value>,

"servicename":<String_value>,

"ocspcheck":<String_value>,

"pushenctrigger":<String_value>,
```

```
"ca":<Boolean_value>,

"snicert":<Boolean_value>,

"skipcaname":<Boolean_value>,

"sendclosenotify":<String_value>,

"dtlsprofilename":<String_value>,

"dtlsflag":<Boolean_value>,

"sslprofile":<String_value>,

"hsts":<String_value>,

"maxage":<Double_value>,

"includesubdomains":<String_value>,

"preload":<String_value>,

"strictsigdigestcheck":<String_value>,

"zerorttearlydata":<String_value>,

"tls13sessionticketsperauthcontext":<Double_value>,

"dhekeyexchangewithpsk":<String_value>

}]}
```

get

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/sslvserver/vservername_value<String>

Query-parameters:

attrs

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/sslvserver/vservername_value<String>?**attrs=property-name1,property-name2**

Use this query parameter to specify the resource details that you want to retrieve.

view

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/sslvserver/vservername_value<String>?**view=summary**

Note: By default, the retrieved results are displayed in detail view (?view=detail).

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "sslvserver": [ {

"vservername":<String_value>,

"cleartextport":<Integer_value>,

"dh":<String_value>,

"dhfile":<String_value>,

"dhcount":<Double_value>,

"dhkeyexpsizelimit":<String_value>,

"ersa":<String_value>,

"ersacount":<Double_value>,
```

```
"sessreuse":<String_value>,

"sesstimeout":<Double_value>,

"cipherredirect":<String_value>,

"crlcheck":<String_value>,

"cipherurl":<String_value>,

"sslv2redirect":<String_value>,

"sslv2url":<String_value>,

"clientauth":<String_value>,

"clientcert":<String_value>,

"sslredirect":<String_value>,

"redirectportrewrite":<String_value>,

"nonfipsciphers":<String_value>,

"ssl2":<String_value>,

"ssl3":<String_value>,

"tls1":<String_value>,

"tls11":<String_value>,

"tls12":<String_value>,

"tls13":<String_value>,

"snienable":<String_value>,

"ocspstapling":<String_value>,

"service":<Double_value>,

"servicename":<String_value>,
```

```
"ocspcheck":<String_value>,

"pushenctrigger":<String_value>,

"ca":<Boolean_value>,

"snicert":<Boolean_value>,

"skipcaname":<Boolean_value>,

"sendclosenotify":<String_value>,

"dtlsprofilename":<String_value>,

"dtlsflag":<Boolean_value>,

"sslprofile":<String_value>,

"hsts":<String_value>,

"maxage":<Double_value>,

"includesubdomains":<String_value>,

"preload":<String_value>,

"strictsigdigestcheck":<String_value>,

"zerorttearlydata":<String_value>,

"tls13sessionticketsperauthcontext":<Double_value>,

"dhekeyexchangewithpsk":<String_value>

}]}
```

count

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/sslvserver?**count=yes**

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "sslvserver": [ { "__count": "#no"} ] }
```

sslvserver_binding

Binding object which returns the resources bound to sslvserver.

Properties

(click to see Operations)

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
vservername	<String>	Read-write	Name of the SSL virtual server for which to show detailed information. Minimum length = 1
sslvserver_sslcipher_binding	<sslvserver_sslcipher_binding[]>	Read-only	sslcipher that can be bound to sslvserver.
sslvserver_ecccurve_binding	<sslvserver_ecccurve_binding[]>	Read-only	ecccurve that can be bound to sslvserver.
sslvserver_sslpolicy_binding	<sslvserver_sslpolicy_binding[]>	Read-only	sslpolicy that can be bound to sslvserver.

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
sslvserver_sslcertkey_binding	<sslvserver_sslcertkey_binding[]>	Read-only	sslcertkey that can be bound to sslvserver.
sslvserver_sslciphersuite_binding	<sslvserver_sslciphersuite_binding[]>	Read-only	sslciphersuite that can be bound to sslvserver.

Operations

(click to see Properties)

GET| GET (ALL)

Some options that you can use for each operations:

- **Getting warnings in response:**NITRO allows you to get warnings in an operation by specifying the "warning" query parameter as "yes". For example, to get warnings while connecting to the Citrix ADC appliance, the URL is as follows:

  http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/login?warning=yes

  If any, the warnings are displayed in the response payload with the HTTP code "209 X-NITRO-WARNING".

- **Authenticated access for individual NITRO operations:**NITRO allows you to logon to the Citrix ADC appliance to perform individual operations. You can use this option instead of creating a NITRO session (using the login object) and then using that session to perform all operations,

  To do this, you must specify the username and password in the request header of the NITRO request as follows:

  X-NITRO-USER:<username>

  X-NITRO-PASS:<password>

  Note: In such cases, make sure that the request header DOES not include the following:

  Cookie:NITRO_AUTH_TOKEN=<tokenvalue>

`*Note:`*

Mandatory parameters are marked in redand placeholder content is marked in <green>.

get

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/sslvserver_binding/vservername_value<String>

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "sslvserver_binding": [ {

"vservername":<String_value>,

"sslvserver_sslciphersuite_binding":<sslvserver_sslciphersuite_binding[]_value>,

"sslvserver_ecccurve_binding":<sslvserver_ecccurve_binding[]_value>,

"sslvserver_sslpolicy_binding":<sslvserver_sslpolicy_binding[]_value>,

"sslvserver_sslcertkey_binding":<sslvserver_sslcertkey_binding[]_value>,

"sslvserver_sslcipher_binding":<sslvserver_sslcipher_binding[]_value>

}]}
```

get (all)

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/sslvserver_binding

Query-parameters:

bulkbindings

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/sslvserver_binding?**bulkbindings=yes**

NITRO allows you to fetch bindings in bulk.

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "sslvserver_binding": [ {

"vservername":<String_value>,

"sslvserver_sslciphersuite_binding":<sslvserver_sslciphersuite_binding[]_value>,

"sslvserver_ecccurve_binding":<sslvserver_ecccurve_binding[]_value>,

"sslvserver_sslpolicy_binding":<sslvserver_sslpolicy_binding[]_value>,

"sslvserver_sslcertkey_binding":<sslvserver_sslcertkey_binding[]_value>,

"sslvserver_sslcipher_binding":<sslvserver_sslcipher_binding[]_value>
```

```
}]}
```

sslvserver_ecccurve_binding

Binding object showing the ecccurve that can be bound to sslvserver.

Properties

(click to see Operations)

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
ecccurvename	<String>	Read-write	Named ECC curve bound to vserver/ service. Possible values = ALL, P_224, P_256, P_384, P_521
vservername	<String>	Read-write	Name of the SSL virtual server. Minimum length = 1
__count	<Double>	Read-write	count parameter

Operations

(click to see Properties)

ADD:| DELETE:| GET| GET (ALL)| COUNT

Some options that you can use for each operations:

- **Getting warnings in response:**NITRO allows you to get warnings in an operation by specifying the "warning" query parameter as "yes". For example, to get warnings while connecting to the Citrix ADC appliance, the URL is as follows:

  http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/login?warning=yes

  If any, the warnings are displayed in the response payload with the HTTP code "209 X-NITRO-WARNING".

- **Authenticated access for individual NITRO operations:**NITRO allows you to logon to the Citrix ADC appliance to perform individual operations. You can use this option instead of creating a NITRO session (using the login object) and then using that session to perform all operations,

  To do this, you must specify the username and password in the request header of the NITRO request as follows:

  X-NITRO-USER:<username>

  X-NITRO-PASS:<password>

  Note: In such cases, make sure that the request header DOES not include the following:

  Cookie:NITRO_AUTH_TOKEN=<tokenvalue>

`*Note:`*

Mandatory parameters are marked in redand placeholder content is marked in <green>.

add

URL:http://<Citrix-ADC-IP-address(NSIP)/nitro/v1/config/sslvserver_ecccurve_binding

HTTP Method:PUT

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Content-Type:application/json

Request Payload:

```
{

"sslvserver_ecccurve_binding":{

<b>"vservername":<String_value>,

</b>"ecccurvename":<String_value>

}}
```

Response:

HTTP Status Code on Success: 201 Created HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error

delete

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/sslvserver_ecccurve_binding/vservername_value<String>

Query-parameters:

args

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/sslvserver_ecccurve_binding/vservername_value<String>?**args=ecccurvename:<String_value>**

HTTP Method:DELETE

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue>

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error

get

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/sslvserver_ecccurve_binding/vservername_value<String>

Query-parameters:

filter

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/sslvserver_ecccurve_binding/vservername_value<String>?**filter=property-name1:property-value1,property-name2:property-value2**

Use this query-parameter to get the filtered set of sslvserver_ecccurve_binding resources configured on Citrix ADC. Filtering can be done on any of the properties of the resource.

pagination

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/sslvserver_ecccurve_binding/vservername_value<String>?**pagesize=#no;pageno=#no**

Use this query-parameter to get the sslvserver_ecccurve_binding resources in chunks.

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "sslvserver_ecccurve_binding": [ {

"ecccurvename":<String_value>,

"vservername":<String_value>

}]}
```

get (all)

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/sslvserver_ecccurve_binding

Query-parameters:

bulkbindings

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/sslvserver_ecccurve_binding?**bulkbindings=yes**

NITRO allows you to fetch bindings in bulk.

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "sslvserver_ecccurve_binding": [ {

"ecccurvename":<String_value>,

"vservername":<String_value>

}]}
```

count

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/sslvserver_ecccurve_binding/vservername_value<String>?**count=yes**

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{"sslvserver_ecccurve_binding": [ { "__count": "#no"} ] }
```

sslvserver_sslcertkey_binding

Binding object showing the sslcertkey that can be bound to sslvserver.

Properties

(click to see Operations)

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
ca	<Boolean>	Read-write	CA certificate.
crlcheck	<String>	Read-write	The state of the CRL check parameter. (Mandatory/ Optional). Possible values = Mandatory, Optional
vservername	<String>	Read-write	Name of the SSL virtual server. Minimum length = 1
certkeyname	<String>	Read-write	The name of the certificate key pair binding.
skipcaname	<Boolean>	Read-write	The flag is used to indicate whether this particular CA

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
			certificate's CA_Name needs to be sent to the SSL client while requesting for client certificate in a SSL handshake.
snicert	<Boolean>	Read-write	The name of the CertKey. Use this option to bind Certkey(s) which will be used in SNI processing.
ocspcheck	<String>	Read-write	The state of the OCSP check parameter. (Mandatory/ Optional). Possible values = Mandatory, Optional
cleartextport	<Integer>	Read-only	Port on which clear-text data is sent by the appliance to the server. Do not specify this parameter for SSL offloading with end-to-end encryption. Default value: 0 Minimum value = 0 Maximum value = 65534
__count	<Double>	Read-only	count parameter

Operations

(click to see Properties)

ADD:| DELETE:| GET| GET (ALL)| COUNT

Some options that you can use for each operations:

- **Getting warnings in response:**NITRO allows you to get warnings in an operation by specifying the "warning" query parameter as "yes". For example, to get warnings while connecting to the Citrix ADC appliance, the URL is as follows:

  http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/login?warning=yes

  If any, the warnings are displayed in the response payload with the HTTP code "209 X-NITRO-WARNING".

- **Authenticated access for individual NITRO operations:**NITRO allows you to logon to the Citrix ADC appliance to perform individual operations. You can use this option instead of creating a NITRO session (using the login object) and then using that session to perform all operations,

  To do this, you must specify the username and password in the request header of the NITRO request as follows:

  X-NITRO-USER:<username>

  X-NITRO-PASS:<password>

  Note: In such cases, make sure that the request header DOES not include the following:

  Cookie:NITRO_AUTH_TOKEN=<tokenvalue>

`*Note:`*

Mandatory parameters are marked in redand placeholder content is marked in <green>.

add

URL:http://<Citrix-ADC-IP-address(NSIP)/nitro/v1/config/sslvserver_sslcertkey_binding

HTTP Method:PUT

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Content-Type:application/json

Request Payload:

```
{
"sslvserver_sslcertkey_binding":{
<b>"vservername":<String_value>,
</b>"certkeyname":<String_value>,
"ca":<Boolean_value>,
"crlcheck":<String_value>,
"skipcaname":<Boolean_value>,
"snicert":<Boolean_value>,
"ocspcheck":<String_value>
}}
```

Response:

HTTP Status Code on Success: 201 Created HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error

delete

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/sslvserver_sslcertkey_binding/vservername_value<String>

Query-parameters:

args

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/sslvserver_sslcertkey_binding/vservername_value<String>?
args=certkeyname:<String_value>,ca:<Boolean_value>,crlcheck:<String_value>,snicert:<Boolean_valu

HTTP Method:DELETE

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue>

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error

get

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/sslvserver_sslcertkey_binding/vservername_value<String>

Query-parameters:

filter

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/sslvserver_sslcertkey_binding/vservername_value<String>?**filter=property-name1:property-value1,property-name2:property-value2**

Use this query-parameter to get the filtered set of sslvserver_sslcertkey_binding resources configured on Citrix ADC. Filtering can be done on any of the properties of the resource.

pagination

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/sslvserver_sslcertkey_binding/vservername_value<String>?**pagesize=#no;pageno=#no**

Use this query-parameter to get the sslvserver_sslcertkey_binding resources in chunks.

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "sslvserver_sslcertkey_binding": [ {

"ca":<Boolean_value>,

"crlcheck":<String_value>,

"vservername":<String_value>,

"certkeyname":<String_value>,

"skipcaname":<Boolean_value>,

"snicert":<Boolean_value>,

"ocspcheck":<String_value>,

"cleartextport":<Integer_value>

}]}
```

get (all)

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/sslvserver_sslcertkey_binding

Query-parameters:

bulkbindings

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/sslvserver_sslcertkey_binding?**bulkbindings=yes**

NITRO allows you to fetch bindings in bulk.

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "sslvserver_sslcertkey_binding": [ {

"ca":<Boolean_value>,

"crlcheck":<String_value>,

"vservername":<String_value>,

"certkeyname":<String_value>,

"skipcaname":<Boolean_value>,

"snicert":<Boolean_value>,

"ocspcheck":<String_value>,

"cleartextport":<Integer_value>

}]}
```

count

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/sslvserver_sslcertkey_binding/vservername_value<String>?**count=yes**

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{"sslvserver_sslcertkey_binding": [ { "__count": "#no"} ] }
```

sslvserver_sslciphersuite_binding

Binding object showing the sslciphersuite that can be bound to sslvserver.

Properties

(click to see Operations)

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
ciphername	<String>	Read-write	The cipher group/ alias/individual cipher configuration.
description	<String>	Read-write	The cipher suite description.
vservername	<String>	Read-write	Name of the SSL virtual server. Minimum length = 1
__count	<Double>	Read-write	count parameter

Operations

(click to see Properties)

ADD:| DELETE:| GET| GET (ALL)| COUNT

Some options that you can use for each operations:

- **Getting warnings in response:**NITRO allows you to get warnings in an operation by specifying the "warning" query parameter as "yes". For example, to get warnings while connecting to the Citrix ADC appliance, the URL is as follows:

  http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/login?warning=yes

  If any, the warnings are displayed in the response payload with the HTTP code "209 X-NITRO-WARNING".

- **Authenticated access for individual NITRO operations:**NITRO allows you to logon to the Citrix ADC appliance to perform individual operations. You can use this option instead of creating a NITRO session (using the login object) and then using that session to perform all operations,

  To do this, you must specify the username and password in the request header of the NITRO request as follows:

  X-NITRO-USER:<username>

  X-NITRO-PASS:<password>

  Note: In such cases, make sure that the request header DOES not include the following:

  Cookie:NITRO_AUTH_TOKEN=<tokenvalue>

`*Note:`*

Mandatory parameters are marked in redand placeholder content is marked in <green>.

add

URL:http://<Citrix-ADC-IP-address(NSIP)/nitro/v1/config/sslvserver_sslciphersuite_binding

HTTP Method:PUT

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Content-Type:application/json

Request Payload:

```
{

"sslvserver_sslciphersuite_binding":{

<b>"vservername":<String_value>,

</b>"ciphername":<String_value>

}}
```

Response:

HTTP Status Code on Success: 201 Created HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error

delete

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/sslvserver_sslciphersuite_binding/vservername_value<String>

Query-parameters:

args

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/sslvserver_sslciphersuite_binding/vservername_value<String>?**args=ciphername:<String_value>**

HTTP Method:DELETE

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue>

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error

get

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/sslvserver_sslciphersuite_binding/vservername_value<String>

Query-parameters:

filter

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/sslvserver_sslciphersuite_binding/vservername_value<String>?**filter=property-name1:property-value1,property-name2:property-value2**

Use this query-parameter to get the filtered set of sslvserver_sslciphersuite_binding resources configured on Citrix ADC. Filtering can be done on any of the properties of the resource.

pagination

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/sslvserver_sslciphersuite_binding/vservername_value<String>?**pagesize=#no;pageno=#no**

Use this query-parameter to get the sslvserver_sslciphersuite_binding resources in chunks.

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "sslvserver_sslciphersuite_binding": [ {

"ciphername":<String_value>,
```

```
"description":<String_value>,

"vservername":<String_value>

}]}
```

get (all)

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/sslvserver_sslciphersuite_binding

Query-parameters:

bulkbindings

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/sslvserver_sslciphersuite_binding?**bulkbindings=yes**

NITRO allows you to fetch bindings in bulk.

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "sslvserver_sslciphersuite_binding": [ {

"ciphername":<String_value>,

"description":<String_value>,
```

```
"vservername":<String_value>

}]}
```

count

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/sslvserver_sslciphersuite_binding/vservername_value<String>?**count=yes**

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{"sslvserver_sslciphersuite_binding": [ { "__count": "#no"} ] }
```

sslvserver_sslcipher_binding

Binding object showing the sslcipher that can be bound to sslvserver.

Properties

(click to see Operations)

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
ciphername	<String>	Read-write	Name of the individual cipher,

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
			user-defined cipher group, or predefined (built-in) cipher alias.
cipheraliasname	<String>	Read-write	The name of the cipher group/alias/ individual cipheri bindings.
description	<String>	Read-write	The cipher suite description.
vservername	<String>	Read-write	Name of the SSL virtual server. Minimum length = 1
__count	<Double>	Read-write	count parameter

Operations

(click to see Properties)

ADD:| DELETE:| GET| GET (ALL)| COUNT

Some options that you can use for each operations:

- **Getting warnings in response:**NITRO allows you to get warnings in an operation by specifying the "warning" query parameter as "yes". For example, to get warnings while connecting to the Citrix ADC appliance, the URL is as follows:

  http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/login?warning=yes

  If any, the warnings are displayed in the response payload with the HTTP code "209 X-NITRO-WARNING".

- **Authenticated access for individual NITRO operations:**NITRO allows you to logon to the Citrix ADC appliance to perform individual operations. You can use this option instead of creating a NITRO session (using the login object) and then using that session to perform all operations,

  To do this, you must specify the username and password in the request header of the NITRO request as follows:

X-NITRO-USER:<username>

X-NITRO-PASS:<password>

Note: In such cases, make sure that the request header DOES not include the following:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue>

`*Note:`*

Mandatory parameters are marked in redand placeholder content is marked in <green>.

add

URL:http://<Citrix-ADC-IP-address(NSIP)/nitro/v1/config/sslvserver_sslcipher_binding

HTTP Method:PUT

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Content-Type:application/json

Request Payload:

```
{
"sslvserver_sslcipher_binding":{
<b>"vservername":<String_value>,
</b>"ciphername":<String_value>
}}
```

Response:

HTTP Status Code on Success: 201 Created HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error

delete

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/sslvserver_sslcipher_binding/vservername_value<String>

Query-parameters:

args

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/sslvserver_sslcipher_binding/vservername_value<String>?**args=ciphername:<String_value>**

HTTP Method:DELETE

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue>

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error

get

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/sslvserver_sslcipher_binding/vservername_value<String>

Query-parameters:

filter

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/sslvserver_sslcipher_binding/vservername_value<String>?**filter=property-name1:property-value1,property-name2:property-value2**

Use this query-parameter to get the filtered set of sslvserver_sslcipher_binding resources configured on Citrix ADC. Filtering can be done on any of the properties of the resource.

pagination

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/sslvserver_sslcipher_binding/vservername_value<String>?**pagesize=#no;pageno=#no**

Use this query-parameter to get the sslvserver_sslcipher_binding resources in chunks.

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "sslvserver_sslcipher_binding": [ {

"ciphername":<String_value>,

"cipheraliasname":<String_value>,

"description":<String_value>,

"vservername":<String_value>

}]}
```

get (all)

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/sslvserver_sslcipher_binding

Query-parameters:

bulkbindings

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/sslvserver_sslcipher_binding?**bulkbindings=yes**

NITRO allows you to fetch bindings in bulk.

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "sslvserver_sslcipher_binding": [ {

"ciphername":<String_value>,

"cipheraliasname":<String_value>,

"description":<String_value>,

"vservername":<String_value>

}]}
```

count

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/sslvserver_sslcipher_binding/vservername_value<String>?**count=yes**

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{"sslvserver_sslcipher_binding": [ { "__count": "#no"} ] }
```

sslvserver_sslpolicy_binding

Binding object showing the sslpolicy that can be bound to sslvserver.

Properties

(click to see Operations)

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
priority	<Double>	Read-write	The priority of the policies bound to this SSL service. Minimum value = 0 Maximum value = 65534
policyname	<String>	Read-write	The name of the SSL policy binding.
labelname	<String>	Read-write	Name of the label to invoke if the current policy rule evaluates to TRUE.
vservername	<String>	Read-write	Name of the SSL virtual server. Minimum length = 1

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
gotopriorityexpression	<String>	Read-write	Expression specifying the priority of the next policy which will get evaluated if the current policy rule evaluates to TRUE.
invoke	<Boolean>	Read-write	Invoke flag. This attribute is relevant only for ADVANCED policies.
type	<String>	Read-write	Bind point to which to bind the policy. Possible Values: REQUEST, INTERCEPT_REQ and CLIENTHELLO_REQ. These bindpoints mean: 1. REQUEST: Policy evaluation will be done at appplication above SSL. This bindpoint is default and is used for actions based on clientauth and client cert. 2. INTERCEPT_REQ: Policy evaluation will be done during SSL handshake to decide whether to intercept or not. Actions allowed with this type are: INTERCEPT, BYPASS and RESET. 3. CLIENTHELLO_RE

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
			Q: Policy evaluation will be done during handling of Client Hello Request. Action allowed with this type is: RESET, FORWARD and PICKCACERTGRP. Default value: REQUEST Possible values = INTERCEPT_REQ, REQUEST, CLIENTHELLO_REQ
labeltype	<String>	Read-write	Type of policy label invocation. Possible values = vserver, service, policylabel
polinherit	<Double>	Read-only	Whether the bound policy is a inherited policy or not. Minimum value = 0 Maximum value = 254
__count	<Double>	Read-only	count parameter

Operations

(click to see Properties)

ADD:| DELETE:| GET| GET (ALL)| COUNT

Some options that you can use for each operations:

- **Getting warnings in response:**NITRO allows you to get warnings in an operation by specifying the "warning" query parameter as "yes". For example, to get warnings while connecting to the Citrix ADC appliance, the URL is as follows:

  http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/login?warning=yes

  If any, the warnings are displayed in the response payload with the HTTP code "209 X-NITRO-WARNING".

- **Authenticated access for individual NITRO operations:**NITRO allows you to logon to the Citrix ADC appliance to perform individual operations. You can use this option instead of creating a NITRO session (using the login object) and then using that session to perform all operations,

  To do this, you must specify the username and password in the request header of the NITRO request as follows:

  X-NITRO-USER:<username>

  X-NITRO-PASS:<password>

  Note: In such cases, make sure that the request header DOES not include the following:

  Cookie:NITRO_AUTH_TOKEN=<tokenvalue>

`*Note:`*

Mandatory parameters are marked in redand placeholder content is marked in <green>.

add

URL:http://<Citrix-ADC-IP-address(NSIP)/nitro/v1/config/sslvserver_sslpolicy_binding

HTTP Method:PUT

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Content-Type:application/json

Request Payload:

```
{
```

```
"sslvserver_sslpolicy_binding":{

<b>"vservername":<String_value>,

</b>"policyname":<String_value>,

"priority":<Double_value>,

"gotopriorityexpression":<String_value>,

"invoke":<Boolean_value>,

"labeltype":<String_value>,

"labelname":<String_value>,

"type":<String_value>

}}
```

Response:

HTTP Status Code on Success: 201 Created HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error

delete

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/sslvserver_sslpolicy_binding/vservername_value<String>

Query-parameters:

args

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/sslvserver_sslpolicy_binding/vservername_value<String>?**args=policyname:<String_value>,priority:<Double_value>,type:<String_value>**

HTTP Method:DELETE

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue>

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error

get

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/sslvserver_sslpolicy_binding/vservername_value<String>

Query-parameters:

filter

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/sslvserver_sslpolicy_binding/vservername_value<String>?**filter=property-name1:property-value1,property-name2:property-value2**

Use this query-parameter to get the filtered set of sslvserver_sslpolicy_binding resources configured on Citrix ADC. Filtering can be done on any of the properties of the resource.

pagination

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/sslvserver_sslpolicy_binding/vservername_value<String>?**pagesize=#no;pageno=#no**

Use this query-parameter to get the sslvserver_sslpolicy_binding resources in chunks.

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "sslvserver_sslpolicy_binding": [ {

"priority":<Double_value>,

"policyname":<String_value>,

"labelname":<String_value>,

"vservername":<String_value>,

"gotopriorityexpression":<String_value>,

"invoke":<Boolean_value>,

"type":<String_value>,

"labeltype":<String_value>,

"polinherit":<Double_value>

}]}
```

get (all)

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/sslvserver_sslpolicy_binding

Query-parameters:

bulkbindings

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/sslvserver_sslpolicy_binding?**bulkbindings=yes**

NITRO allows you to fetch bindings in bulk.

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "sslvserver_sslpolicy_binding": [ {

"priority":<Double_value>,

"policyname":<String_value>,

"labelname":<String_value>,

"vservername":<String_value>,

"gotopriorityexpression":<String_value>,

"invoke":<Boolean_value>,

"type":<String_value>,

"labeltype":<String_value>,

"polinherit":<Double_value>

}]}
```

count

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/sslvserver_sslpolicy_binding/vservername_value<String>?**count=yes**

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{"sslvserver_sslpolicy_binding": [ { "__count": "#no"} ] }
```

sslwrapkey

Configuration for WRAP key resource.

Properties

(click to see Operations)

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
wrapkeyname	<String>	Read-write	Name for the wrap key. Must begin with an ASCII alphanumeric or underscore (_) character, and must contain only ASCII alphanumeric, underscore, hash (#), period (.), space, colon (:), at (@), equals (=), and hyphen (-) characters. Cannot

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
			be changed after the wrap key is created. The following requirement applies only to the Citrix ADC CLI: If the name includes one or more spaces, enclose the name in double or single quotation marks (for example, "my key" or 'my key'). Minimum length = 1
password	<String>	Read-write	Password string for the wrap key. Minimum length = 1
salt	<String>	Read-write	Salt string for the wrap key. Minimum length = 1
__count	<Double>	Read-only	count parameter

Operations

(click to see Properties)

CREATE| DELETE| GET (ALL)| COUNT

Some options that you can use for each operations:

- **Getting warnings in response:**NITRO allows you to get warnings in an operation by specifying the "warning" query parameter as "yes". For example, to get warnings while connecting to the Citrix ADC appliance, the URL is as follows:

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/login?warning=yes

If any, the warnings are displayed in the response payload with the HTTP code "209 X-NITRO-WARNING".

- **Authenticated access for individual NITRO operations:**NITRO allows you to logon to the Citrix ADC appliance to perform individual operations. You can use this option instead of creating a NITRO session (using the login object) and then using that session to perform all operations,

  To do this, you must specify the username and password in the request header of the NITRO request as follows:

  X-NITRO-USER:<username>

  X-NITRO-PASS:<password>

  Note: In such cases, make sure that the request header DOES not include the following:

  Cookie:NITRO_AUTH_TOKEN=<tokenvalue>

`*Note:`*

Mandatory parameters are marked in redand placeholder content is marked in <green>.

create

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/sslwrapkey?**action=create**

HTTP Method:POST

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Content-Type:application/json

Request Payload:

```
{"sslwrapkey":{

<b>"wrapkeyname":<String_value>,

</b><b>"password":<String_value>,

</b><b>"salt":<String_value>
```

```
</b>}}
```

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error

delete

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/sslwrapkey/wrapkeyname_value<String>

HTTP Method:DELETE

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue>

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error

get (all)

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/sslwrapkey

Query-parameters:

attrs

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/sslwrapkey?**attrs=property-name1,property-name2**

Use this query parameter to specify the resource details that you want to retrieve.

filter

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/sslwrapkey?**filter=property-name1:property-val1,property-name2:property-val2**

Use this query-parameter to get the filtered set of sslwrapkey resources configured on Citrix ADC. Filtering can be done on any of the properties of the resource.

view

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/sslwrapkey?**view=summary**

Note: By default, the retrieved results are displayed in detail view (?view=detail).

pagination

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/sslwrapkey?**pagesize=#no;pageno=#no**

Use this query-parameter to get the sslwrapkey resources in chunks.

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "sslwrapkey": [ {

"wrapkeyname":<String_value>

}]}
```

count

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/sslwrapkey?**count=yes**

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "sslwrapkey": [ { "__count": "#no"} ] }
```

sslcacertgroup

Configuration for Group of CA certificate-key pairs resource.

Properties

(click to see Operations)

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
cacertgroupname	<String>	Read-write	Name given to the CA certificate group. The name will be used to add the CA certificates to the group. Must begin with an ASCII alphanumeric or underscore (_) character, and must contain only ASCII alphanumeric, underscore, hash (#), period (.), space,

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
			colon (:), at (@), equals (=), and hyphen (-) characters. The following requirement applies only to the Citrix ADC CLI: If the name includes one or more spaces, enclose the name in double or single quotation marks (for example, "my file" or 'my file'). Minimum length = 1 Maximum length = 31
cacertgroupreferences	<Double>	Read-only	Count for ssl actions referring to this ca certificate group.
ocspcheck	<String>	Read-only	The state of the OCSP check parameter. (Mandatory/Optional). Possible values = Mandatory, Optional
crlcheck	<String>	Read-only	The state of the CRL check parameter. (Mandatory/Optional). Possible values = Mandatory, Optional
__count	<Double>	Read-only	count parameter

Operations

(click to see Properties)

ADD| DELETE| GET (ALL)| GET| COUNT

Some options that you can use for each operations:

- **Getting warnings in response:**NITRO allows you to get warnings in an operation by specifying the "warning" query parameter as "yes". For example, to get warnings while connecting to the Citrix ADC appliance, the URL is as follows:

  http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/login?warning=yes

  If any, the warnings are displayed in the response payload with the HTTP code "209 X-NITRO-WARNING".

- **Authenticated access for individual NITRO operations:**NITRO allows you to logon to the Citrix ADC appliance to perform individual operations. You can use this option instead of creating a NITRO session (using the login object) and then using that session to perform all operations,

  To do this, you must specify the username and password in the request header of the NITRO request as follows:

  X-NITRO-USER:<username>

  X-NITRO-PASS:<password>

  Note: In such cases, make sure that the request header DOES not include the following:

  Cookie:NITRO_AUTH_TOKEN=<tokenvalue>

`*Note:`*

Mandatory parameters are marked in redand placeholder content is marked in <green>.

add

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/sslcacertgroup

HTTP Method:POST

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Content-Type:application/json

Request Payload:

```
{"sslcacertgroup":{

<b>"cacertgroupname":<String_value>

</b>}}
```

Response:

HTTP Status Code on Success: 201 Created HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error

delete

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/sslcacertgroup/cacertgroupname_value<String>

HTTP Method:DELETE

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue>

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error

get (all)

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/sslcacertgroup

Query-parameters:

attrs

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/sslcacertgroup?**attrs=property-name1,property-name2**

Use this query parameter to specify the resource details that you want to retrieve.

filter

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/sslcacertgroup?**filter=property-name1:property-val1,property-name2:property-val2**

Use this query-parameter to get the filtered set of sslcacertgroup resources configured on Citrix ADC. Filtering can be done on any of the properties of the resource.

view

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/sslcacertgroup?**view=summary**

Note: By default, the retrieved results are displayed in detail view (?view=detail).

pagination

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/sslcacertgroup?**pagesize=#no;pageno=#no**

Use this query-parameter to get the sslcacertgroup resources in chunks.

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "sslcacertgroup": [ {

"cacertgroupname":<String_value>,

"cacertgroupreferences":<Double_value>,
```

```
"ocspcheck":<String_value>,

"crlcheck":<String_value>

}]}
```

get

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/sslcacertgroup/cacertgroupname_value<String>

Query-parameters:

attrs

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/sslcacertgroup/cacertgroupname_value<String>?**attrs=property-name1,property-name2**

Use this query parameter to specify the resource details that you want to retrieve.

view

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/sslcacertgroup/cacertgroupname_value<String>?**view=summary**

Note: By default, the retrieved results are displayed in detail view (?view=detail).

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "sslcacertgroup": [ {

"cacertgroupname":<String_value>,

"cacertgroupreferences":<Double_value>,

"ocspcheck":<String_value>,

"crlcheck":<String_value>

}]}
```

count

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/sslcacertgroup?**count=yes**

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "sslcacertgroup": [ { "__count": "#no"} ] }
```

sslcacertgroup_binding

Binding object which returns the resources bound to sslcacertgroup.

Properties

(click to see Operations)

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
cacertgroupname	<String>	Read-write	Name of the CA certificate group for which to show detailed information. Minimum length = 1 Maximum length = 31
sslcacertgroup_sslcertkey_binding	<sslcacertgroup_sslcertkey_binding[]>	Read-only	sslcertkey that can be bound to sslcacertgroup.

Operations

(click to see Properties)

GET| GET (ALL)

Some options that you can use for each operations:

- **Getting warnings in response:**NITRO allows you to get warnings in an operation by specifying the "warning" query parameter as "yes". For example, to get warnings while connecting to the Citrix ADC appliance, the URL is as follows:

  http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/login?warning=yes

  If any, the warnings are displayed in the response payload with the HTTP code "209 X-NITRO-WARNING".

- **Authenticated access for individual NITRO operations:**NITRO allows you to logon to the Citrix ADC appliance to perform individual operations. You can use this option instead of creating a NITRO session (using the login object) and then using that session to perform all operations,

  To do this, you must specify the username and password in the request header of the NITRO request as follows:

X-NITRO-USER:<username>

X-NITRO-PASS:<password>

Note: In such cases, make sure that the request header DOES not include the following:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue>

`*Note:`*

Mandatory parameters are marked in redand placeholder content is marked in <green>.

get

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/sslcacertgroup_binding/cacertgroupname_value<String>

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "sslcacertgroup_binding": [ {

"cacertgroupname":<String_value>,

"sslcacertgroup_sslcertkey_binding":<sslcacertgroup_sslcertkey_binding[]_value>

}]}
```

get (all)

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/sslcacertgroup_binding

Query-parameters:

bulkbindings

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/sslcacertgroup_binding?**bulkbindings=yes**

NITRO allows you to fetch bindings in bulk.

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "sslcacertgroup_binding": [ {

"cacertgroupname":<String_value>,

"sslcacertgroup_sslcertkey_binding":<sslcacertgroup_sslcertkey_binding[]_value>

}]}
```

sslcacertgroup_sslcertkey_binding

Binding object showing the sslcertkey that can be bound to sslcacertgroup.

Properties

(click to see Operations)

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
cacertgroupname	<String>	Read-write	Name given to the CA certificate group. The name will be used to add the CA certificates to the group. Must begin with an ASCII alphanumeric or underscore (_) character, and must contain only ASCII alphanumeric, underscore, hash (#), period (.), space, colon (:), at (@), equals (=), and hyphen (-) characters. The following requirement applies only to the Citrix ADC CLI: If the name includes one or more spaces, enclose the name in double or single quotation marks (for example, "my file" or 'my file'). Minimum length = 1 Maximum length = 31
certkeyname	<String>	Read-write	Name for the certkey added to the Citrix ADC. Must begin

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
			with an ASCII alphanumeric or underscore (_) character, and must contain only ASCII alphanumeric, underscore, hash (#), period (.), space, colon (:), at (@), equals (=), and hyphen (-) characters. Cannot be changed after the certificate-key pair is created.The following requirement applies only to the Citrix ADC CLI: If the name includes one or more spaces, enclose the name in double or single quotation marks (for example, "my cert" or 'my cert'). Minimum length = 1 Maximum length = 31
crlcheck	<String>	Read-write	The state of the CRL check parameter. (Mandatory/ Optional). Possible values = Mandatory, Optional
ocspcheck	<String>	Read-write	The state of the OCSP check parameter.

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
			(Mandatory/ Optional). Possible values = Mandatory, Optional
__count	<Double>	Read-write	count parameter

Operations

(click to see Properties)

ADD:| DELETE:| GET| GET (ALL)| COUNT

Some options that you can use for each operations:

- **Getting warnings in response:**NITRO allows you to get warnings in an operation by specifying the "warning" query parameter as "yes". For example, to get warnings while connecting to the Citrix ADC appliance, the URL is as follows:

  http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/login?warning=yes

  If any, the warnings are displayed in the response payload with the HTTP code "209 X-NITRO-WARNING".

- **Authenticated access for individual NITRO operations:**NITRO allows you to logon to the Citrix ADC appliance to perform individual operations. You can use this option instead of creating a NITRO session (using the login object) and then using that session to perform all operations,

  To do this, you must specify the username and password in the request header of the NITRO request as follows:

  X-NITRO-USER:<username>

  X-NITRO-PASS:<password>

  Note: In such cases, make sure that the request header DOES not include the following:

  Cookie:NITRO_AUTH_TOKEN=<tokenvalue>

`*Note:`*

Mandatory parameters are marked in redand placeholder content is marked in <green>.

add

URL:http://<Citrix-ADC-IP-address(NSIP)/nitro/v1/config/sslcacertgroup_sslcertkey_binding

HTTP Method:PUT

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Content-Type:application/json

Request Payload:

```
{

"sslcacertgroup_sslcertkey_binding":{

<b>"cacertgroupname":<String_value>,

</b><b>"certkeyname":<String_value>,

</b>"crlcheck":<String_value>,

"ocspcheck":<String_value>

}}
```

Response:

HTTP Status Code on Success: 201 Created HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error

delete

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/sslcacertgroup_sslcertkey_binding/cacertgroupname_value<String>

Query-parameters:

args

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/sslcacertgroup_sslcertkey_binding/cacertgroupname_value<String>?**args=certkeyname:<String_value>**

HTTP Method:DELETE

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue>

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error

get

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/sslcacertgroup_sslcertkey_binding/cacertgroupname_value<String>

Query-parameters:

filter

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/sslcacertgroup_sslcertkey_binding/cacertgroupname_value<String>?**filter=property-name1:property-value1,property-name2:property-value2**

Use this query-parameter to get the filtered set of sslcacertgroup_sslcertkey_binding resources configured on Citrix ADC. Filtering can be done on any of the properties of the resource.

pagination

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/sslcacertgroup_sslcertkey_binding/cacertgroupname_value<String>?**pagesize=#no;pageno=#no**

Use this query-parameter to get the sslcacertgroup_sslcertkey_binding resources in chunks.

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "sslcacertgroup_sslcertkey_binding": [ {

"cacertgroupname":<String_value>,

"certkeyname":<String_value>,

"crlcheck":<String_value>,

"ocspcheck":<String_value>

}]}
```

get (all)

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/sslcacertgroup_sslcertkey_binding

Query-parameters:

bulkbindings

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/sslcacertgroup_sslcertkey_binding?**bulkbindings=yes**

NITRO allows you to fetch bindings in bulk.

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "sslcacertgroup_sslcertkey_binding": [ {

"cacertgroupname":<String_value>,

"certkeyname":<String_value>,

"crlcheck":<String_value>,

"ocspcheck":<String_value>

}]}
```

count

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/sslcacertgroup_sslcertkey_binding/cacertgroupname_value<String>?**count=yes**

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{"sslcacertgroup_sslcertkey_binding": [ { "__count": "#no"} ] }
```

sslecdsakey

Configuration for ecdsa key resource.

Properties

(click to see Operations)

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
keyfile	<String>	Read-write	Name for and, optionally, path to the ECDSA key file. /nsconfig/ssl/ is the default path. Maximum length = 63
curve	<String>	Read-write	Curve id to generate ECDSA key. Only P_256 and P_384 are supported. Default value: FIPSEXP_F4 Possible values = P_256, P_384
keyform	<String>	Read-write	Format in which the ECDSA key file is stored on the appliance. Default value: PEM Possible values = DER, PEM

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
des	<Boolean>	Read-write	Encrypt the generated ECDSA key by using the DES algorithm. On the command line, you are prompted to enter the pass phrase (password) that is used to encrypt the key.
des3	<Boolean>	Read-write	Encrypt the generated ECDSA key by using the Triple-DES algorithm. On the command line, you are prompted to enter the pass phrase (password) that is used to encrypt the key.
aes256	<Boolean>	Read-write	Encrypt the generated ECDSA key by using the AES algorithm.
password	<String>	Read-write	Pass phrase to use for encryption if DES or DES3 option is selected. Minimum length = 1 Maximum length = 31
pkcs8	<Boolean>	Read-write	Create the private key in PKCS#8 format.

Operations

(click to see Properties)

CREATE

Some options that you can use for each operations:

- **Getting warnings in response:**NITRO allows you to get warnings in an operation by specifying the "warning" query parameter as "yes". For example, to get warnings while connecting to the Citrix ADC appliance, the URL is as follows:

  http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/login?warning=yes

  If any, the warnings are displayed in the response payload with the HTTP code "209 X-NITRO-WARNING".

- **Authenticated access for individual NITRO operations:**NITRO allows you to logon to the Citrix ADC appliance to perform individual operations. You can use this option instead of creating a NITRO session (using the login object) and then using that session to perform all operations,

  To do this, you must specify the username and password in the request header of the NITRO request as follows:

  X-NITRO-USER:<username>

  X-NITRO-PASS:<password>

  Note: In such cases, make sure that the request header DOES not include the following:

  Cookie:NITRO_AUTH_TOKEN=<tokenvalue>

`*Note:`*

Mandatory parameters are marked in redand placeholder content is marked in <green>.

create

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/sslecdsakey?**action=create**

HTTP Method:POST

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Content-Type:application/json

Request Payload:

```
{"sslecdsakey":{

<b>"keyfile":<String_value>,

</b><b>"curve":<String_value>,

</b>"keyform":<String_value>,

"des":<Boolean_value>,

"des3":<Boolean_value>,

"aes256":<Boolean_value>,

"password":<String_value>,

"pkcs8":<Boolean_value>

}}
```

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error

ssllogprofile

Configuration for SSL logging Profile resource.

Properties

(click to see Operations)

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
name	<String>	Read-write	The name of the ssllogprofile.

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
			Minimum length = 1 Maximum length = 127
ssllogclauth	<String>	Read-write	log all SSL ClAuth events. Default value: DISABLED Possible values = ENABLED, DISABLED
ssllogclauthfailures	<String>	Read-write	log all SSL ClAuth error events. Default value: DISABLED Possible values = ENABLED, DISABLED
ssllogghs	<String>	Read-write	log all SSL HS events. Default value: DISABLED Possible values = ENABLED, DISABLED
ssllogghsfailures	<String>	Read-write	log all SSL HS error events. Default value: DISABLED

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
			Possible values = ENABLED, DISABLED
__count	<Double>	Read-only	count parameter

Operations

(click to see Properties)

ADD| UPDATE| UNSET| DELETE| GET (ALL)| GET| COUNT

Some options that you can use for each operations:

- **Getting warnings in response:**NITRO allows you to get warnings in an operation by specifying the "warning" query parameter as "yes". For example, to get warnings while connecting to the Citrix ADC appliance, the URL is as follows:

  http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/login?warning=yes

  If any, the warnings are displayed in the response payload with the HTTP code "209 X-NITRO-WARNING".

- **Authenticated access for individual NITRO operations:**NITRO allows you to logon to the Citrix ADC appliance to perform individual operations. You can use this option instead of creating a NITRO session (using the login object) and then using that session to perform all operations,

  To do this, you must specify the username and password in the request header of the NITRO request as follows:

  X-NITRO-USER:<username>

  X-NITRO-PASS:<password>

  Note: In such cases, make sure that the request header DOES not include the following:

  Cookie:NITRO_AUTH_TOKEN=<tokenvalue>

`*Note:`*

Mandatory parameters are marked in redand placeholder content is marked in <green>.

add

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/ssllogprofile

HTTP Method:POST

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Content-Type:application/json

Request Payload:

```
{"ssllogprofile":{

<b>"name":<String_value>,

</b>"ssllogclauth":<String_value>,

"ssllogclauthfailures":<String_value>,

"ssllogs":<String_value>,

"ssllogsfailures":<String_value>

}}
```

Response:

HTTP Status Code on Success: 201 Created HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error

update

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/ssllogprofile

HTTP Method:PUT

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Content-Type:application/json

Request Payload:

```
{"ssllogprofile":{

<b>"name":<String_value>,

</b>"ssllogclauth":<String_value>,

"ssllogclauthfailures":<String_value>,

"sslloghs":<String_value>,

"sslloghsfailures":<String_value>

}}
```

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error

unset

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/ssllogprofile?**action=unset**

HTTP Method:POST

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Content-Type:application/json

Request Payload:

```
{"ssllogprofile":{

<b>"name":<String_value>,

</b>"ssllogclauth":true,
```

```
"ssllogclauthfailures":true,

"sslloghs":true,

"sslloghsfailures":true

}}
```

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error

delete

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/ssllogprofile/name_value<String>

HTTP Method:DELETE

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue>

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error

get (all)

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/ssllogprofile

Query-parameters:

attrs

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/ssllogprofile?**attrs=property-name1,property-name2**

Use this query parameter to specify the resource details that you want to retrieve.

filter

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/ssllogprofile?**filter=property-name1:property-val1,property-name2:property-val2**

Use this query-parameter to get the filtered set of ssllogprofile resources configured on Citrix ADC. Filtering can be done on any of the properties of the resource.

view

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/ssllogprofile?**view=summary**

Note: By default, the retrieved results are displayed in detail view (?view=detail).

pagination

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/ssllogprofile?**pagesize=#no;pageno=#no**

Use this query-parameter to get the ssllogprofile resources in chunks.

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "ssllogprofile": [ {

"name":<String_value>,

"ssllogclauth":<String_value>,

"ssllogclauthfailures":<String_value>,

"sslloghs":<String_value>,
```

```
"ssllloghsfailures":<String_value>

}]}
```

get

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/ssllogprofile/name_value<String>

Query-parameters:

attrs

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/ssllogprofile/name_value<String>?**attrs=property-name1,property-name2**

Use this query parameter to specify the resource details that you want to retrieve.

view

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/ssllogprofile/name_value<String>?**view=summary**

Note: By default, the retrieved results are displayed in detail view (?view=detail).

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "ssllogprofile": [ {
```

```
"name":<String_value>,

"ssllogclauth":<String_value>,

"ssllogclauthfailures":<String_value>,

"sslloghs":<String_value>,

"sslloghsfailures":<String_value>

}]}
```

count

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/ssllogprofile?**count=yes**

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "ssllogprofile": [ { "__count": "#no"} ] }
```

SSL VPN

SSL VPN

Virtual Private Network configuration.

NAME	DESCRIPTION
vpnalwaysonprofile	Configuration for AlwyasON profile
vpnclientlessaccesspolicy	Configuration for Clientless VPN rewrite policy
vpnclientlessaccesspolicy_binding	Binding object showing the resources that can be bound to vpnclientlessaccesspolicy
vpnclientlessaccesspolicy_vpnglobal_binding	Binding object showing the vpnglobal that can be bound to vpnclientlessaccesspolicy
vpnclientlessaccesspolicy_vpnvserver_binding	Binding object showing the vpnvserver that can be bound to vpnclientlessaccesspolicy
vpnclientlessaccessprofile	Configuration for Clientless VPN rewrite profile
vpnepaprofile	Configuration for Epa profile
vpneula	Configuration for EULA for vservers
vpnformssoaction	Configuration for Form sso action
vpnglobal_appcontroller_binding	Binding object showing the appcontroller that can be bound to vpnglobal
vpnglobal_auditnslogpolicy_binding	Binding object showing the auditnslogpolicy that can be bound to vpnglobal
vpnglobal_auditsyslogpolicy_binding	Binding object showing the auditsyslogpolicy that can be bound to vpnglobal
vpnglobal_authenticationcertpolicy_binding	Binding object showing the authenticationcertpolicy that can be bound to vpnglobal

NAME	DESCRIPTION
vpnglobal_authenticationldappolicy_binding	Binding object showing the authenticationldappolicy that can be bound to vpnglobal
vpnglobal_authenticationlocalpolicy_binding	Binding object showing the authenticationlocalpolicy that can be bound to vpnglobal
vpnglobal_authenticationnegotiatepolicy_binding	Binding object showing the authenticationnegotiatepolicy that can be bound to vpnglobal
vpnglobal_authenticationpolicy_binding	Binding object showing the authenticationpolicy that can be bound to vpnglobal
vpnglobal_authenticationradiuspolicy_binding	Binding object showing the authenticationradiuspolicy that can be bound to vpnglobal
vpnglobal_authenticationsamlpolicy_binding	Binding object showing the authenticationsamlpolicy that can be bound to vpnglobal
vpnglobal_authenticationtacacspolicy_binding	Binding object showing the authenticationtacacspolicy that can be bound to vpnglobal
vpnglobal_binding	Binding object showing the resources that can be bound to vpnglobal
vpnglobal_domain_binding	Binding object showing the domain that can be bound to vpnglobal
vpnglobal_intranetip6_binding	Binding object showing the intranetip6 that can be bound to vpnglobal
vpnglobal_intranetip_binding	Binding object showing the intranetip that can be bound to vpnglobal

NAME	DESCRIPTION
vpnglobal_sharefileserver_binding	Binding object showing the sharefileserver that can be bound to vpnglobal
vpnglobal_sslcertkey_binding	Binding object showing the sslcertkey that can be bound to vpnglobal
vpnglobal_staserver_binding	Binding object showing the staserver that can be bound to vpnglobal
vpnglobal_vpnclientlessaccesspolicy_binding	Binding object showing the vpnclientlessaccesspolicy that can be bound to vpnglobal
vpnglobal_vpneula_binding	Binding object showing the vpneula that can be bound to vpnglobal
vpnglobal_vpnintranetapplication_binding	Binding object showing the vpnintranetapplication that can be bound to vpnglobal
vpnglobal_vpnnexthopserver_binding	Binding object showing the vpnnexthopserver that can be bound to vpnglobal
vpnglobal_vpnportaltheme_binding	Binding object showing the vpnportaltheme that can be bound to vpnglobal
vpnglobal_vpnsessionpolicy_binding	Binding object showing the vpnsessionpolicy that can be bound to vpnglobal
vpnglobal_vpntrafficpolicy_binding	Binding object showing the vpntrafficpolicy that can be bound to vpnglobal
vpnglobal_vpnurl_binding	Binding object showing the vpnurl that can be bound to vpnglobal
vpnglobal_vpnurlpolicy_binding	Binding object showing the vpnurlpolicy that can be bound to vpnglobal
vpnicaconnection	Configuration for active ica connections
vpnicadtlsconnection	Configuration for active ica connections

NAME	DESCRIPTION
vpnintranetapplication	Configuration for SSLVPN intranet application
vpnnexthopserver	Configuration for Next Hop Server
vpnparameter	Configuration for VPN parameter
vpnpcoipconnection	Configuration for PCoIP connection
vpnpcoipprofile	Configuration for PCoIP session profile
vpnpcoipvserverprofile	Configuration for PCoIP vserver profile
vpnportaltheme	Configuration for portaltheme
vpnsamlssoprofile	Configuration for SAML sso action
vpnsessionaction	Configuration for VPN session action
vpnsessionpolicy	Configuration for VPN session policy
vpnsessionpolicy_aaagroup_binding	Binding object showing the aaagroup that can be bound to vpnsessionpolicy
vpnsessionpolicy_aaauser_binding	Binding object showing the aaauser that can be bound to vpnsessionpolicy
vpnsessionpolicy_binding	Binding object showing the resources that can be bound to vpnsessionpolicy
vpnsessionpolicy_vpnglobal_binding	Binding object showing the vpnglobal that can be bound to vpnsessionpolicy
vpnsessionpolicy_vpnvserver_binding	Binding object showing the vpnvserver that can be bound to vpnsessionpolicy
vpnsfconfig	Configuration for Create a StoreFront configuration file
vpnstoreinfo	Configuration for Store information for a URL
vpntrafficaction	Configuration for VPN traffic action

NAME	DESCRIPTION
vpntrafficpolicy	Configuration for VPN traffic policy
vpntrafficpolicy_aaagroup_binding	Binding object showing the aaagroup that can be bound to vpntrafficpolicy
vpntrafficpolicy_aaauser_binding	Binding object showing the aaauser that can be bound to vpntrafficpolicy
vpntrafficpolicy_binding	Binding object showing the resources that can be bound to vpntrafficpolicy
vpntrafficpolicy_vpnglobal_binding	Binding object showing the vpnglobal that can be bound to vpntrafficpolicy
vpntrafficpolicy_vpnvserver_binding	Binding object showing the vpnvserver that can be bound to vpntrafficpolicy
vpnurl	Configuration for VPN URL
vpnurlaction	Configuration for VPN url action
vpnurlpolicy	Configuration for VPN url policy
vpnurlpolicy_aaagroup_binding	Binding object showing the aaagroup that can be bound to vpnurlpolicy
vpnurlpolicy_aaauser_binding	Binding object showing the aaauser that can be bound to vpnurlpolicy
vpnurlpolicy_binding	Binding object showing the resources that can be bound to vpnurlpolicy
vpnurlpolicy_vpnglobal_binding	Binding object showing the vpnglobal that can be bound to vpnurlpolicy
vpnurlpolicy_vpnvserver_binding	Binding object showing the vpnvserver that can be bound to vpnurlpolicy
vpnvserver	Configuration for VPN virtual server

NAME	DESCRIPTION
vpnvserver_aaapreauthenticationpolicy_binding	Binding object showing the aaapreauthenticationpolicy that can be bound to vpnvserver
vpnvserver_appcontroller_binding	Binding object showing the appcontroller that can be bound to vpnvserver
vpnvserver_appflowpolicy_binding	Binding object showing the appflowpolicy that can be bound to vpnvserver
vpnvserver_auditnslogpolicy_binding	Binding object showing the auditnslogpolicy that can be bound to vpnvserver
vpnvserver_auditsyslogpolicy_binding	Binding object showing the auditsyslogpolicy that can be bound to vpnvserver
vpnvserver_authenticationcertpolicy_binding	Binding object showing the authenticationcertpolicy that can be bound to vpnvserver
vpnvserver_authenticationdfapolicy_binding	Binding object showing the authenticationdfapolicy that can be bound to vpnvserver
vpnvserver_authenticationldappolicy_binding	Binding object showing the authenticationldappolicy that can be bound to vpnvserver
vpnvserver_authenticationlocalpolicy_binding	Binding object showing the authenticationlocalpolicy that can be bound to vpnvserver
vpnvserver_authenticationloginschemapolicy_binding	Binding object showing the authenticationloginschemapolicy that can be bound to vpnvserver
vpnvserver_authenticationnegotiatepolicy_binding	Binding object showing the authenticationnegotiatepolicy that can be bound to vpnvserver

NAME	DESCRIPTION
vpnvserver_authenticationoauthidppolicy_binding	Binding object showing the authenticationoauthidppolicy that can be bound to vpnvserver
vpnvserver_authenticationpolicy_binding	Binding object showing the authenticationpolicy that can be bound to vpnvserver
vpnvserver_authenticationradiuspolicy_binding	Binding object showing the authenticationradiuspolicy that can be bound to vpnvserver
vpnvserver_authenticationsamlidppolicy_binding	Binding object showing the authenticationsamlidppolicy that can be bound to vpnvserver
vpnvserver_authenticationsamlpolicy_binding	Binding object showing the authenticationsamlpolicy that can be bound to vpnvserver
vpnvserver_authenticationtacacspolicy_binding	Binding object showing the authenticationtacacspolicy that can be bound to vpnvserver
vpnvserver_authenticationwebauthpolicy_binding	Binding object showing the authenticationwebauthpolicy that can be bound to vpnvserver
vpnvserver_binding	Binding object showing the resources that can be bound to vpnvserver
vpnvserver_cachepolicy_binding	Binding object showing the cachepolicy that can be bound to vpnvserver
vpnvserver_cspolicy_binding	Binding object showing the cspolicy that can be bound to vpnvserver
vpnvserver_feopolicy_binding	Binding object showing the feopolicy that can be bound to vpnvserver

NAME	DESCRIPTION
vpnvserver_icapolicy_binding	Binding object showing the icapolicy that can be bound to vpnvserver
vpnvserver_intranetip6_binding	Binding object showing the intranetip6 that can be bound to vpnvserver
vpnvserver_intranetip_binding	Binding object showing the intranetip that can be bound to vpnvserver
vpnvserver_responderpolicy_binding	Binding object showing the responderpolicy that can be bound to vpnvserver
vpnvserver_rewritepolicy_binding	Binding object showing the rewritepolicy that can be bound to vpnvserver
vpnvserver_sharefileserver_binding	Binding object showing the sharefileserver that can be bound to vpnvserver
vpnvserver_staserver_binding	Binding object showing the staserver that can be bound to vpnvserver
vpnvserver_vpnclientlessaccesspolicy_binding	Binding object showing the vpnclientlessaccesspolicy that can be bound to vpnvserver
vpnvserver_vpnepaprofile_binding	Binding object showing the vpnepaprofile that can be bound to vpnvserver
vpnvserver_vpneula_binding	Binding object showing the vpneula that can be bound to vpnvserver
vpnvserver_vpnintranetapplication_binding	Binding object showing the vpnintranetapplication that can be bound to vpnvserver
vpnvserver_vpnnexthopserver_binding	Binding object showing the vpnnexthopserver that can be bound to vpnvserver
vpnvserver_vpnportaltheme_binding	Binding object showing the vpnportaltheme that can be bound to vpnvserver

NAME	DESCRIPTION
vpnvserver_vpnsessionpolicy_binding	Binding object showing the vpnsessionpolicy that can be bound to vpnvserver
vpnvserver_vpntrafficpolicy_binding	Binding object showing the vpntrafficpolicy that can be bound to vpnvserver
vpnvserver_vpnurl_binding	Binding object showing the vpnurl that can be bound to vpnvserver
vpnvserver_vpnurlpolicy_binding	Binding object showing the vpnurlpolicy that can be bound to vpnvserver

vpnalwaysonprofile

Configuration for AlwyasON profile resource.

Properties

(click to see Operations)

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
name	<String>	Read-write	name of AlwaysON profile. Minimum length = 1
networkaccessonvpnfailure	<String>	Read-write	Option to block network traffic when tunnel is not established(and the config requires that tunnel be established). When set to onlyToGateway, the network traffic to and from the client (except Gateway IP) is blocked. When set to fullAccess, the

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
			network traffic is not blocked. Default value: fullAccess, Possible values = onlyToGateway, fullAccess
clientcontrol	<String>	Read-write	Allow/Deny user to log off and connect to another Gateway. Default value: DENY Possible values = ALLOW, DENY
locationbasedvpn	<String>	Read-write	Option to decide if tunnel should be established when in enterprise network. When locationBasedVPN is remote, client tries to detect if it is located in enterprise network or not and establishes the tunnel if not in enterprise network. Dns suffixes configured using -add dns suffix- are used to decide if the client is in the enterprise network or not. If the resolution of the DNS suffix results in private IP, client is said to be in

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
			enterprise network. When set to EveryWhere, the client skips the check to detect if it is on the enterprise network and tries to establish the tunnel. Default value: Remote Possible values = Remote, Everywhere
__count	<Double>	Read-only	count parameter

Operations

(click to see Properties)

ADD| DELETE| UPDATE| UNSET| GET (ALL)| GET| COUNT

Some options that you can use for each operations:

- **Getting warnings in response:**NITRO allows you to get warnings in an operation by specifying the "warning" query parameter as "yes". For example, to get warnings while connecting to the Citrix ADC appliance, the URL is as follows:

  http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/login?warning=yes

  If any, the warnings are displayed in the response payload with the HTTP code "209 X-NITRO-WARNING".
- **Authenticated access for individual NITRO operations:**NITRO allows you to logon to the Citrix ADC appliance to perform individual operations. You can use this option instead of creating a NITRO session (using the login object) and then using that session to perform all operations,

  To do this, you must specify the username and password in the request header of the NITRO request as follows:

X-NITRO-USER:<username>

X-NITRO-PASS:<password>

Note: In such cases, make sure that the request header DOES not include the following:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue>

`*Note:` *

Mandatory parameters are marked in redand placeholder content is marked in <green>.

add

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/vpnalwaysonprofile

HTTP Method:POST

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Content-Type:application/json

Request Payload:

```
{"vpnalwaysonprofile":{

<b>"name":<String_value>,

</b>"networkaccessonvpnfailure":<String_value>,

"clientcontrol":<String_value>,

"locationbasedvpn":<String_value>

}}
```

Response:

HTTP Status Code on Success: 201 Created HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error

delete

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/vpnalwaysonprofile/name_value<String>

HTTP Method:DELETE

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue>

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error

update

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/vpnalwaysonprofile

HTTP Method:PUT

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Content-Type:application/json

Request Payload:

```
{"vpnalwaysonprofile":{

<b>"name":<String_value>,

</b>"networkaccessonvpnfailure":<String_value>,

"clientcontrol":<String_value>,

"locationbasedvpn":<String_value>

}}
```

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error

unset

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/vpnalwaysonprofile?**action=unset**

HTTP Method:POST

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Content-Type:application/json

Request Payload:

```
{"vpnalwaysonprofile":{

<b>"name":<String_value>,

</b>"networkaccessonvpnfailure":true,

"clientcontrol":true,

"locationbasedvpn":true

}}
```

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error

get (all)

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/vpnalwaysonprofile

Query-parameters:

attrs

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/vpnalwaysonprofile?**attrs=property-name1,property-name2**

Use this query parameter to specify the resource details that you want to retrieve.

filter

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/vpnalwaysonprofile?**filter=property-name1:property-val1,property-name2:property-val2**

Use this query-parameter to get the filtered set of vpnalwaysonprofile resources configured on Citrix ADC. Filtering can be done on any of the properties of the resource.

view

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/vpnalwaysonprofile?**view=summary**

Note: By default, the retrieved results are displayed in detail view (?view=detail).

pagination

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/vpnalwaysonprofile?**pagesize=#no;pageno=#no**

Use this query-parameter to get the vpnalwaysonprofile resources in chunks.

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "vpnalwaysonprofile": [ {
```

```
"name":<String_value>,

"networkaccessonvpnfailure":<String_value>,

"clientcontrol":<String_value>,

"locationbasedvpn":<String_value>

}]}
```

get

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/vpnalwaysonprofile/name_value<String>

Query-parameters:

attrs

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/vpnalwaysonprofile/name_value<String>?**attrs=property-name1,property-name2**

Use this query parameter to specify the resource details that you want to retrieve.

view

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/vpnalwaysonprofile/name_value<String>?**view=summary**

Note: By default, the retrieved results are displayed in detail view (?view=detail).

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "vpnalwaysonprofile": [ {

"name":<String_value>,

"networkaccessonvpnfailure":<String_value>,

"clientcontrol":<String_value>,

"locationbasedvpn":<String_value>

}]}
```

count

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/vpnalwaysonprofile?**count=yes**

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "vpnalwaysonprofile": [ { "__count": "#no"} ] }
```

vpnclientlessaccesspolicy

Configuration for Clientless VPN rewrite policy resource.

Properties

(click to see Operations)

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
name	<String>	Read-write	Name of the new clientless access policy. Minimum length = 1
rule	<String>	Read-write	Expression, or name of a named expression, specifying the traffic that matches the policy. The following requirements apply only to the Citrix ADC CLI: * If the expression includes one or more spaces, enclose the entire expression in double quotation marks. * If the expression itself includes double quotation marks, escape the quotations by using the \ character.

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
			* Alternatively, you can use single quotation marks to enclose the rule, in which case you do not have to escape the double quotation marks.
profilename	<String>	Read-write	Name of the profile to invoke for the clientless access.
undefaction	<String>	Read-only	The UNDEF action.
hits	<Double>	Read-only	The number of times the policy evaluated to true.
undefhits	<Double>	Read-only	The number of times the policy evaluation resulted in undefined processing.
description	<String>	Read-only	Description of the clientless access policy.
isdefault	<Boolean>	Read-only	A value of true is returned if it is a default vpnclientlessrwpolicy .
__count	<Double>	Read-only	count parameter

Operations

(click to see Properties)

ADD| DELETE| UPDATE| GET (ALL)| GET| COUNT

Some options that you can use for each operations:

- **Getting warnings in response:**NITRO allows you to get warnings in an operation by specifying the "warning" query parameter as "yes". For example, to get warnings while connecting to the Citrix ADC appliance, the URL is as follows:

  http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/login?warning=yes

  If any, the warnings are displayed in the response payload with the HTTP code "209 X-NITRO-WARNING".

- **Authenticated access for individual NITRO operations:**NITRO allows you to logon to the Citrix ADC appliance to perform individual operations. You can use this option instead of creating a NITRO session (using the login object) and then using that session to perform all operations,

  To do this, you must specify the username and password in the request header of the NITRO request as follows:

  X-NITRO-USER:<username>

  X-NITRO-PASS:<password>

  Note: In such cases, make sure that the request header DOES not include the following:

  Cookie:NITRO_AUTH_TOKEN=<tokenvalue>

`*Note:`*

Mandatory parameters are marked in redand placeholder content is marked in <green>.

add

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/vpnclientlessaccesspolicy

HTTP Method:POST

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Content-Type:application/json

Request Payload:

```
{"vpnclientlessaccesspolicy":{

<b>"name":<String_value>,

</b><b>"rule":<String_value>,

</b><b>"profilename":<String_value>

</b>}}
```

Response:

HTTP Status Code on Success: 201 Created HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error

delete

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/vpnclientlessaccesspolicy/name_value<String>

HTTP Method:DELETE

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue>

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error

update

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/vpnclientlessaccesspolicy

HTTP Method:PUT

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Content-Type:application/json

Request Payload:

```
{"vpnclientlessaccesspolicy":{

<b>"name":<String_value>,

</b>"rule":<String_value>,

"profilename":<String_value>

}}
```

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error

get (all)

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/vpnclientlessaccesspolicy

Query-parameters:

attrs

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/vpnclientlessaccesspolicy?**attrs=property-name1,property-name2**

Use this query parameter to specify the resource details that you want to retrieve.

filter

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/vpnclientlessaccesspolicy?**filter=property-name1:property-val1,property-name2:property-val2**

Use this query-parameter to get the filtered set of vpnclientlessaccesspolicy resources configured on Citrix ADC. Filtering can be done on any of the properties of the resource.

view

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/vpnclientlessaccesspolicy?**view=summary**

Note: By default, the retrieved results are displayed in detail view (?view=detail).

pagination

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/vpnclientlessaccesspolicy?**pagesize=#no;pageno=#no**

Use this query-parameter to get the vpnclientlessaccesspolicy resources in chunks.

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "vpnclientlessaccesspolicy": [ {

"name":<String_value>,

"rule":<String_value>,

"profilename":<String_value>,

"undefaction":<String_value>,

"hits":<Double_value>,

"undefhits":<Double_value>,

"description":<String_value>,
```

```
"isdefault":<Boolean_value>

}]}
```

get

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/vpnclientlessaccesspolicy/name_value<String>

Query-parameters:

attrs

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/vpnclientlessaccesspolicy/name_value<String>?**attrs=property-name1,property-name2**

Use this query parameter to specify the resource details that you want to retrieve.

view

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/vpnclientlessaccesspolicy/name_value<String>?**view=summary**

Note: By default, the retrieved results are displayed in detail view (?view=detail).

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "vpnclientlessaccesspolicy": [ {

"name":<String_value>,

"rule":<String_value>,

"profilename":<String_value>,

"undefaction":<String_value>,

"hits":<Double_value>,

"undefhits":<Double_value>,

"description":<String_value>,

"isdefault":<Boolean_value>

}]}
```

count

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/vpnclientlessaccesspolicy?**count=yes**

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "vpnclientlessaccesspolicy": [ { "__count": "#no"} ] }
```

vpnclientlessaccesspolicy_binding

Binding object which returns the resources bound to vpnclientlessaccesspolicy.

Properties

(click to see Operations)

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
name	<String>	Read-write	Name of the clientless access policy to display. Minimum length = 1
vpnclientlessaccesspolicy_vpnvserver_binding	<vpnclientlessaccesspolicy_vpnvserver_binding[]>	Read-only	vpnvserver that can be bound to vpnclientlessaccesspolicy.
vpnclientlessaccesspolicy_vpnglobal_binding	<vpnclientlessaccesspolicy_vpnglobal_binding[]>	Read-only	vpnglobal that can be bound to vpnclientlessaccesspolicy.

Operations

(click to see Properties)

GET| GET (ALL)

Some options that you can use for each operations:

- **Getting warnings in response:**NITRO allows you to get warnings in an operation by specifying the "warning" query parameter as "yes". For example, to get warnings while connecting to the Citrix ADC appliance, the URL is as follows:

  http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/login?warning=yes

If any, the warnings are displayed in the response payload with the HTTP code "209 X-NITRO-WARNING".

- **Authenticated access for individual NITRO operations:**NITRO allows you to logon to the Citrix ADC appliance to perform individual operations. You can use this option instead of creating a NITRO session (using the login object) and then using that session to perform all operations,

  To do this, you must specify the username and password in the request header of the NITRO request as follows:

  X-NITRO-USER:<username>

  X-NITRO-PASS:<password>

  Note: In such cases, make sure that the request header DOES not include the following:

  Cookie:NITRO_AUTH_TOKEN=<tokenvalue>

`*Note:`*

Mandatory parameters are marked in redand placeholder content is marked in <green>.

get

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/vpnclientlessaccesspolicy_binding/name_value<String>

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "vpnclientlessaccesspolicy_binding": [ {

"name":<String_value>,

"vpnclientlessaccesspolicy_vpnglobal_binding":<vpnclientlessaccesspolicy_vpnglobal_binding[]_value>,

"vpnclientlessaccesspolicy_vpnvserver_binding":<vpnclientlessaccesspolicy_vpnvserver_binding[]_value>

}]}
```

get (all)

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/vpnclientlessaccesspolicy_binding

Query-parameters:

bulkbindings

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/vpnclientlessaccesspolicy_binding?**bulkbindings=yes**

NITRO allows you to fetch bindings in bulk.

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "vpnclientlessaccesspolicy_binding": [ {

"name":<String_value>,

"vpnclientlessaccesspolicy_vpnglobal_binding":<vpnclientlessaccesspolicy_vpnglobal_binding[]_value>,

"vpnclientlessaccesspolicy_vpnvserver_binding":<vpnclientlessaccesspolicy_vpnvserver_binding[]_value>

}]}
```

vpnclientlessaccesspolicy_vpnglobal_binding

Binding object showing the vpnglobal that can be bound to vpnclientlessaccesspolicy.

Properties

(click to see Operations)

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
boundto	<String>	Read-write	Location where policy is bound.
name	<String>	Read-write	Name of the clientless access policy to display. Minimum length = 1
priority	<Double>	Read-only	Specifies the priority of the policy. Minimum value = 1 Maximum value = 2147483647

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
activepolicy	<Integer>	Read-only	Indicates whether policy is bound or not.
__count	<Double>	Read-only	count parameter

Operations

(click to see Properties)

GET| GET (ALL)| COUNT

Some options that you can use for each operations:

- **Getting warnings in response:**NITRO allows you to get warnings in an operation by specifying the "warning" query parameter as "yes". For example, to get warnings while connecting to the Citrix ADC appliance, the URL is as follows:

  http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/login?warning=yes

  If any, the warnings are displayed in the response payload with the HTTP code "209 X-NITRO-WARNING".

- **Authenticated access for individual NITRO operations:**NITRO allows you to logon to the Citrix ADC appliance to perform individual operations. You can use this option instead of creating a NITRO session (using the login object) and then using that session to perform all operations,

  To do this, you must specify the username and password in the request header of the NITRO request as follows:

  X-NITRO-USER:<username>

  X-NITRO-PASS:<password>

  Note: In such cases, make sure that the request header DOES not include the following:

  Cookie:NITRO_AUTH_TOKEN=<tokenvalue>

`*Note:`*

Mandatory parameters are marked in redand placeholder content is marked in <green>.

get

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/vpnclientlessaccesspolicy_vpnglobal_binding/name_value<String>

Query-parameters:

filter

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/vpnclientlessaccesspolicy_vpnglobal_binding/name_value<String>?**filter=property-name1:property-value1,property-name2:property-value2**

Use this query-parameter to get the filtered set of vpnclientlessaccesspolicy_vpnglobal_binding resources configured on Citrix ADC. Filtering can be done on any of the properties of the resource.

pagination

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/vpnclientlessaccesspolicy_vpnglobal_binding/name_value<String>?**pagesize=#no;pageno=#no**

Use this query-parameter to get the vpnclientlessaccesspolicy_vpnglobal_binding resources in chunks.

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "vpnclientlessaccesspolicy_vpnglobal_binding": [ {

"boundto":<String_value>,
```

```
"name":<String_value>,

"priority":<Double_value>,

"activepolicy":<Integer_value>

}]}
```

get (all)

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/vpnclientlessaccesspolicy_vpnglobal_binding

Query-parameters:

bulkbindings

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/vpnclientlessaccesspolicy_vpnglobal_binding?**bulkbindings=yes**

NITRO allows you to fetch bindings in bulk.

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "vpnclientlessaccesspolicy_vpnglobal_binding": [ {

"boundto":<String_value>,
```

```
"name":<String_value>,

"priority":<Double_value>,

"activepolicy":<Integer_value>

}]}
```

count

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/vpnclientlessaccesspolicy_vpnglobal_binding/name_value<String>?**count=yes**

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{"vpnclientlessaccesspolicy_vpnglobal_binding": [ { "__count": "#no"} ] }
```

vpnclientlessaccesspolicy_vpnvserver_binding

Binding object showing the vpnvserver that can be bound to vpnclientlessaccesspolicy.

Properties

(click to see Operations)

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
boundto	<String>	Read-write	Location where policy is bound.
name	<String>	Read-write	Name of the clientless access policy to display. Minimum length = 1
priority	<Double>	Read-only	Specifies the priority of the policy. Minimum value = 1 Maximum value = 2147483647
activepolicy	<Integer>	Read-only	Indicates whether policy is bound or not.
__count	<Double>	Read-only	count parameter

Operations

(click to see Properties)

GET| GET (ALL)| COUNT

Some options that you can use for each operations:

- **Getting warnings in response:**NITRO allows you to get warnings in an operation by specifying the "warning" query parameter as "yes". For example, to get warnings while connecting to the Citrix ADC appliance, the URL is as follows:

  http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/login?warning=yes

  If any, the warnings are displayed in the response payload with the HTTP code "209 X-NITRO-WARNING".

- **Authenticated access for individual NITRO operations:**NITRO allows you to logon to the Citrix ADC appliance to perform individual operations. You can use this option instead of

creating a NITRO session (using the login object) and then using that session to perform all operations,

To do this, you must specify the username and password in the request header of the NITRO request as follows:

X-NITRO-USER:<username>

X-NITRO-PASS:<password>

Note: In such cases, make sure that the request header DOES not include the following:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue>

`*Note:`*

Mandatory parameters are marked in redand placeholder content is marked in <green>.

get

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/vpnclientlessaccesspolicy_vpnvserver_binding/name_value<String>

Query-parameters:

filter

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/vpnclientlessaccesspolicy_vpnvserver_binding/name_value<String>?**filter=property-name1:property-value1,property-name2:property-value2**

Use this query-parameter to get the filtered set of vpnclientlessaccesspolicy_vpnvserver_binding resources configured on Citrix ADC. Filtering can be done on any of the properties of the resource.

pagination

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/vpnclientlessaccesspolicy_vpnvserver_binding/name_value<String>?**pagesize=#no;pageno=#no**

Use this query-parameter to get the vpnclientlessaccesspolicy_vpnvserver_binding resources in chunks.

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "vpnclientlessaccesspolicy_vpnvserver_binding": [ {

"boundto":<String_value>,

"name":<String_value>,

"priority":<Double_value>,

"activepolicy":<Integer_value>

}]}
```

get (all)

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/vpnclientlessaccesspolicy_vpnvserver_binding

Query-parameters:

bulkbindings

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/vpnclientlessaccesspolicy_vpnvserver_binding?**bulkbindings=yes**

NITRO allows you to fetch bindings in bulk.

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "vpnclientlessaccesspolicy_vpnvserver_binding": [ {

"boundto":<String_value>,

"name":<String_value>,

"priority":<Double_value>,

"activepolicy":<Integer_value>

}]}
```

count

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/vpnclientlessaccesspolicy_vpnvserver_binding/name_value<String>?**count=yes**

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{"vpnclientlessaccesspolicy_vpnvserver_binding": [ { "__count": "#no"} ] }
```

vpnclientlessaccessprofile

Configuration for Clientless VPN rewrite profile resource.

Properties

(click to see Operations)

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
profilename	<String>	Read-write	Name for the Citrix Gateway clientless access profile. Must begin with an ASCII alphabetic or underscore (_) character, and must consist only of ASCII alphanumeric, underscore, hash (#), period (.), space, colon (:), at (@), equals (=), and hyphen (-) characters. Cannot be changed after the profile is created. The following requirement applies only to the Citrix ADC CLI:

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
			If the name includes one or more spaces, enclose the name in double or single quotation marks (for example, "my profile" or 'my profile'). Minimum length = 1
urlrewritepolicylabel	<String>	Read-write	Name of the configured URL rewrite policy label. If you do not specify a policy label name, then URLs are not rewritten. Minimum length = 1
javascriptrewritepolicylabel	<String>	Read-write	Name of the configured JavaScript rewrite policy label. If you do not specify a policy label name, then JAVA scripts are not rewritten. Minimum length = 1
reqhdrrewritepolicylabel	<String>	Read-write	Name of the configured Request rewrite policy label. If you do not specify a policy label name, then requests are not rewritten. Minimum length = 1

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
reshdrrewritepolicylabel	<String>	Read-write	Name of the configured Response rewrite policy label. Minimum length = 1
regexforfindingurlinjavascript	<String>	Read-write	Name of the pattern set that contains the regular expressions, which match the URL in Java script. Minimum length = 1
regexforfindingurlincss	<String>	Read-write	Name of the pattern set that contains the regular expressions, which match the URL in the CSS. Minimum length = 1
regexforfindingurlinxcomponent	<String>	Read-write	Name of the pattern set that contains the regular expressions, which match the URL in X Component. Minimum length = 1
regexforfindingurlinxml	<String>	Read-write	Name of the pattern set that contains the regular expressions, which match the URL in XML. Minimum length = 1
regexforfindingcustomurls	<String>	Read-write	Name of the pattern set that contains the regular expressions, which match the URLs in the custom content type other

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
			than HTML, CSS, XML, XCOMP, and JavaScript. The custom content type should be included in the patset ns_cvpn_custom_content_types. Minimum length = 1
clientconsumedcookies	<String>	Read-write	Specify the name of the pattern set containing the names of the cookies, which are allowed between the client and the server. If a pattern set is not specified, Citrix Gateway does not allow any cookies between the client and the server. A cookie that is not specified in the pattern set is handled by Citrix Gateway on behalf of the client. Minimum length = 1
requirepersistentcookie	<String>	Read-write	Specify whether a persistent session cookie is set and accepted for clientless access. If this parameter is set to ON, COM objects, such as MSOffice, which are invoked by the browser can

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
			access the files using clientless access. Use caution because the persistent cookie is stored on the disk. Default value: OFF Possible values = ON, OFF
cssrewritepolicylabel	<String>	Read-only	The configured CSS rewrite policylabel. Minimum length = 1
xmlrewritepolicylabel	<String>	Read-only	The configured XML rewrite policylabel. Minimum length = 1
xcomponentrewritepolicylabel	<String>	Read-only	The configured X-Component rewrite policylabel. Minimum length = 1
isdefault	<Boolean>	Read-only	A value of true is returned if it is a default vpnclientlessrwprofile.
description	<String>	Read-only	Description of the clientless access profile.
__count	<Double>	Read-only	count parameter

Operations

(click to see Properties)

ADD| DELETE| UPDATE| UNSET| GET (ALL)| GET| COUNT

Some options that you can use for each operations:

- **Getting warnings in response:**NITRO allows you to get warnings in an operation by specifying the "warning" query parameter as "yes". For example, to get warnings while connecting to the Citrix ADC appliance, the URL is as follows:

  http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/login?warning=yes

  If any, the warnings are displayed in the response payload with the HTTP code "209 X-NITRO-WARNING".

- **Authenticated access for individual NITRO operations:**NITRO allows you to logon to the Citrix ADC appliance to perform individual operations. You can use this option instead of creating a NITRO session (using the login object) and then using that session to perform all operations,

  To do this, you must specify the username and password in the request header of the NITRO request as follows:

  X-NITRO-USER:<username>

  X-NITRO-PASS:<password>

  Note: In such cases, make sure that the request header DOES not include the following:

  Cookie:NITRO_AUTH_TOKEN=<tokenvalue>

`*Note:`*

Mandatory parameters are marked in redand placeholder content is marked in <green>.

add

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/vpnclientlessaccessprofile

HTTP Method:POST

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Content-Type:application/json

Request Payload:

```
{"vpnclientlessaccessprofile":{

<b>"profilename":<String_value>

</b>}}
```

Response:

HTTP Status Code on Success: 201 Created HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error

delete

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/vpnclientlessaccessprofile/profilename_value<String>

HTTP Method:DELETE

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue>

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error

update

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/vpnclientlessaccessprofile

HTTP Method:PUT

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Content-Type:application/json

Request Payload:

```
{"vpnclientlessaccessprofile":{

<b>"profilename":<String_value>,

</b>"urlrewritepolicylabel":<String_value>,

"javascriptrewritepolicylabel":<String_value>,

"reqhdrrewritepolicylabel":<String_value>,

"reshdrrewritepolicylabel":<String_value>,

"regexforfindingurlinjavascript":<String_value>,

"regexforfindingurlincss":<String_value>,

"regexforfindingurlinxcomponent":<String_value>,

"regexforfindingurlinxml":<String_value>,

"regexforfindingcustomurls":<String_value>,

"clientconsumedcookies":<String_value>,

"requirepersistentcookie":<String_value>

}}
```

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error

unset

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/vpnclientlessaccessprofile?**action=unset**

HTTP Method:POST

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Content-Type:application/json

Request Payload:

```
{"vpnclientlessaccessprofile":{

<b>"profilename":<String_value>,

</b>"urlrewritepolicylabel":true,

"javascriptrewritepolicylabel":true,

"reqhdrrewritepolicylabel":true,

"reshdrrewritepolicylabel":true,

"regexforfindingurlinjavascript":true,

"regexforfindingurlincss":true,

"regexforfindingurlinxcomponent":true,

"regexforfindingurlinxml":true,

"regexforfindingcustomurls":true,

"clientconsumedcookies":true,

"requirepersistentcookie":true

}}
```

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error

get (all)

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/vpnclientlessaccessprofile

Query-parameters:

attrs

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/vpnclientlessaccessprofile?**attrs=property-name1,property-name2**

Use this query parameter to specify the resource details that you want to retrieve.

filter

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/vpnclientlessaccessprofile?**filter=property-name1:property-val1,property-name2:property-val2**

Use this query-parameter to get the filtered set of vpnclientlessaccessprofile resources configured on Citrix ADC. Filtering can be done on any of the properties of the resource.

view

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/vpnclientlessaccessprofile?**view=summary**

Note: By default, the retrieved results are displayed in detail view (?view=detail).

pagination

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/vpnclientlessaccessprofile?**pagesize=#no;pageno=#no**

Use this query-parameter to get the vpnclientlessaccessprofile resources in chunks.

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "vpnclientlessaccessprofile": [ {

"profilename":<String_value>,

"urlrewritepolicylabel":<String_value>,

"javascriptrewritepolicylabel":<String_value>,

"cssrewritepolicylabel":<String_value>,

"xmlrewritepolicylabel":<String_value>,

"xcomponentrewritepolicylabel":<String_value>,

"reqhdrrewritepolicylabel":<String_value>,

"reshdrrewritepolicylabel":<String_value>,

"regexforfindingurlinjavascript":<String_value>,

"regexforfindingurlincss":<String_value>,

"regexforfindingurlinxcomponent":<String_value>,

"regexforfindingurlinxml":<String_value>,

"regexforfindingcustomurls":<String_value>,

"clientconsumedcookies":<String_value>,

"requirepersistentcookie":<String_value>,

"isdefault":<Boolean_value>,

"description":<String_value>

}]}
```

get

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/vpnclientlessaccessprofile/profilename_value<String>

Query-parameters:

attrs

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/vpnclientlessaccessprofile/profilename_value<String>?**attrs=property-name1,property-name2**

Use this query parameter to specify the resource details that you want to retrieve.

view

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/vpnclientlessaccessprofile/profilename_value<String>?**view=summary**

Note: By default, the retrieved results are displayed in detail view (?view=detail).

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "vpnclientlessaccessprofile": [ {

"profilename":<String_value>,
```

```
"urlrewritepolicylabel":<String_value>,

"javascriptrewritepolicylabel":<String_value>,

"cssrewritepolicylabel":<String_value>,

"xmlrewritepolicylabel":<String_value>,

"xcomponentrewritepolicylabel":<String_value>,

"reqhdrrewritepolicylabel":<String_value>,

"reshdrrewritepolicylabel":<String_value>,

"regexforfindingurlinjavascript":<String_value>,

"regexforfindingurlincss":<String_value>,

"regexforfindingurlinxcomponent":<String_value>,

"regexforfindingurlinxml":<String_value>,

"regexforfindingcustomurls":<String_value>,

"clientconsumedcookies":<String_value>,

"requirepersistentcookie":<String_value>,

"isdefault":<Boolean_value>,

"description":<String_value>

}]}
```

count

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/vpnclientlessaccessprofile?**count=yes**

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "vpnclientlessaccessprofile": [ { "__count": "#no"} ] }
```

vpnepaprofile

Configuration for Epa profile resource.

Properties

(click to see Operations)

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
name	<String>	Read-write	name of device profile. Minimum length = 1
filename	<String>	Read-write	filename of the deviceprofile data xml. Minimum length = 1
data	<String>	Read-write	deviceprofile data xml. Minimum length = 1
__count	<Double>	Read-only	count parameter

Operations

(click to see Properties)

ADD| DELETE| GET (ALL)| GET| COUNT

Some options that you can use for each operations:

- **Getting warnings in response:**NITRO allows you to get warnings in an operation by specifying the "warning" query parameter as "yes". For example, to get warnings while connecting to the Citrix ADC appliance, the URL is as follows:

  http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/login?warning=yes

  If any, the warnings are displayed in the response payload with the HTTP code "209 X-NITRO-WARNING".

- **Authenticated access for individual NITRO operations:**NITRO allows you to logon to the Citrix ADC appliance to perform individual operations. You can use this option instead of creating a NITRO session (using the login object) and then using that session to perform all operations,

  To do this, you must specify the username and password in the request header of the NITRO request as follows:

  X-NITRO-USER:<username>

  X-NITRO-PASS:<password>

  Note: In such cases, make sure that the request header DOES not include the following:

  Cookie:NITRO_AUTH_TOKEN=<tokenvalue>

`*Note:`*

Mandatory parameters are marked in redand placeholder content is marked in <green>.

add

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/vpnepaprofile

HTTP Method:POST

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Content-Type:application/json

Request Payload:

```
{"vpnepaprofile":{

<b>"name":<String_value>,

</b>"filename":<String_value>,

"data":<String_value>

}}
```

Response:

HTTP Status Code on Success: 201 Created HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error

delete

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/vpnepaprofile/name_value<String>

HTTP Method:DELETE

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue>

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error

get (all)

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/vpnepaprofile

Query-parameters:

attrs

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/vpnepaprofile?**attrs=property-name1,property-name2**

Use this query parameter to specify the resource details that you want to retrieve.

filter

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/vpnepaprofile?**filter=property-name1:property-val1,property-name2:property-val2**

Use this query-parameter to get the filtered set of vpnepaprofile resources configured on Citrix ADC. Filtering can be done on any of the properties of the resource.

view

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/vpnepaprofile?**view=summary**

Note: By default, the retrieved results are displayed in detail view (?view=detail).

pagination

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/vpnepaprofile?**pagesize=#no;pageno=#no**

Use this query-parameter to get the vpnepaprofile resources in chunks.

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "vpnepaprofile": [ {
```

```
"name":<String_value>,

"data":<String_value>

}]}
```

get

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/vpnepaprofile/name_value<String>

Query-parameters:

attrs

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/vpnepaprofile/name_value<String>?**attrs=property-name1,property-name2**

Use this query parameter to specify the resource details that you want to retrieve.

view

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/vpnepaprofile/name_value<String>?**view=summary**

Note: By default, the retrieved results are displayed in detail view (?view=detail).

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "vpnepaprofile": [ {

"name":<String_value>,

"data":<String_value>

}]}
```

count

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/vpnepaprofile?**count=yes**

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "vpnepaprofile": [ { "__count": "#no"} ] }
```

vpneula

Configuration for EULA for vservers resource.

Properties

(click to see Operations)

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
name	<String>	Read-write	Name for the eula. Minimum length = 1
__count	<Double>	Read-only	count parameter

Operations

(click to see Properties)

ADD| DELETE| GET (ALL)| GET| COUNT

Some options that you can use for each operations:

- **Getting warnings in response:**NITRO allows you to get warnings in an operation by specifying the "warning" query parameter as "yes". For example, to get warnings while connecting to the Citrix ADC appliance, the URL is as follows:

  http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/login?warning=yes

  If any, the warnings are displayed in the response payload with the HTTP code "209 X-NITRO-WARNING".

- **Authenticated access for individual NITRO operations:**NITRO allows you to logon to the Citrix ADC appliance to perform individual operations. You can use this option instead of creating a NITRO session (using the login object) and then using that session to perform all operations,

  To do this, you must specify the username and password in the request header of the NITRO request as follows:

  X-NITRO-USER:<username>

  X-NITRO-PASS:<password>

  Note: In such cases, make sure that the request header DOES not include the following:

  Cookie:NITRO_AUTH_TOKEN=<tokenvalue>

`*Note:`*

Mandatory parameters are marked in redand placeholder content is marked in <green>.

add

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/vpneula

HTTP Method:POST

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Content-Type:application/json

Request Payload:

```
{"vpneula":{

<b>"name":<String_value>

</b>}}
```

Response:

HTTP Status Code on Success: 201 Created HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error

delete

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/vpneula/name_value<String>

HTTP Method:DELETE

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue>

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error

get (all)

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/vpneula

Query-parameters:

attrs

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/vpneula?**attrs=property-name1,property-name2**

Use this query parameter to specify the resource details that you want to retrieve.

filter

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/vpneula?**filter=property-name1:property-val1,property-name2:property-val2**

Use this query-parameter to get the filtered set of vpneula resources configured on Citrix ADC. Filtering can be done on any of the properties of the resource.

view

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/vpneula?**view=summary**

Note: By default, the retrieved results are displayed in detail view (?view=detail).

pagination

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/vpneula?**pagesize=#no;pageno=#no**

Use this query-parameter to get the vpneula resources in chunks.

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "vpneula": [ {

"name":<String_value>,



}]}
```

get

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/vpneula/name_value<String>

Query-parameters:

attrs

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/vpneula/name_value<String>?**attrs=property-name1,property-name2**

Use this query parameter to specify the resource details that you want to retrieve.

view

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/vpneula/name_value<String>?**view=summary**

Note: By default, the retrieved results are displayed in detail view (?view=detail).

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "vpneula": [ {

"name":<String_value>,



}]}
```

count

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/vpneula?**count=yes**

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "vpneula": [ { "__count": "#no"} ] }
```

vpnformssoaction

Configuration for Form sso action resource.

Properties

(click to see Operations)

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
name	<String>	Read-write	Name for the form based single sign-on profile. Minimum length = 1
actionurl	<String>	Read-write	Root-relative URL to which the completed form is submitted. Minimum length = 1
userfield	<String>	Read-write	Name of the form field in which the user types in the user ID. Minimum length = 1
passwdfield	<String>	Read-write	Name of the form field in which the user types in the password. Minimum length = 1
ssosuccessrule	<String>	Read-write	Expression that defines the criteria for SSO success. Expression such as checking for cookie in the response is a common example.
namevaluepair	<String>	Read-write	Other name-value pair attributes to send to the server, in addition to sending the user name and password. Value names are separated

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
			by an ampersand (;), such as in name1=value1;name2=value2.
responsesize	<Double>	Read-write	Maximum number of bytes to allow in the response size. Specifies the number of bytes in the response to be parsed for extracting the forms. Default value: 8096
nvtype	<String>	Read-write	How to process the name-value pair. Available settings function as follows: * STATIC - The administrator-configured values are used. * DYNAMIC - The response is parsed, the form is extracted, and then submitted. Default value: DYNAMIC Possible values = STATIC, DYNAMIC
submitmethod	<String>	Read-write	HTTP method (GET or POST) used by the single sign-on form to send the

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
			logon credentials to the logon server. Default value: GET Possible values = GET, POST
__count	<Double>	Read-only	count parameter

Operations

(click to see Properties)

ADD| DELETE| UPDATE| UNSET| GET (ALL)| GET| COUNT

Some options that you can use for each operations:

- **Getting warnings in response:**NITRO allows you to get warnings in an operation by specifying the "warning" query parameter as "yes". For example, to get warnings while connecting to the Citrix ADC appliance, the URL is as follows:

  http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/login?warning=yes

  If any, the warnings are displayed in the response payload with the HTTP code "209 X-NITRO-WARNING".

- **Authenticated access for individual NITRO operations:**NITRO allows you to logon to the Citrix ADC appliance to perform individual operations. You can use this option instead of creating a NITRO session (using the login object) and then using that session to perform all operations,

  To do this, you must specify the username and password in the request header of the NITRO request as follows:

  X-NITRO-USER:<username>

  X-NITRO-PASS:<password>

  Note: In such cases, make sure that the request header DOES not include the following:

  Cookie:NITRO_AUTH_TOKEN=<tokenvalue>

`*Note:`*

Mandatory parameters are marked in redand placeholder content is marked in <green>.

add

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/vpnformssoaction

HTTP Method:POST

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Content-Type:application/json

Request Payload:

```
{"vpnformssoaction":{

<b>"name":<String_value>,

</b><b>"actionurl":<String_value>,

</b><b>"userfield":<String_value>,

</b><b>"passwdfield":<String_value>,

</b><b>"ssosuccessrule":<String_value>,

</b>"namevaluepair":<String_value>,

"responsesize":<Double_value>,

"nvtype":<String_value>,

"submitmethod":<String_value>

}}
```

Response:

HTTP Status Code on Success: 201 Created HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error

delete

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/vpnformssoaction/name_value<String>

HTTP Method:DELETE

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue>

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error

update

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/vpnformssoaction

HTTP Method:PUT

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Content-Type:application/json

Request Payload:

```
{"vpnformssoaction":{

<b>"name":<String_value>,

</b>"actionurl":<String_value>,

"userfield":<String_value>,

"passwdfield":<String_value>,

"ssosuccessrule":<String_value>,

"responsesize":<Double_value>,

"namevaluepair":<String_value>,
```

```
"nvtype":<String_value>,

"submitmethod":<String_value>

}}
```

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error

unset

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/vpnformssoaction?**action=unset**

HTTP Method:POST

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Content-Type:application/json

Request Payload:

```
{"vpnformssoaction":{

<b>"name":<String_value>,

</b>"responsesize":true,

"namevaluepair":true,

"nvtype":true,

"submitmethod":true

}}
```

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error

get (all)

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/vpnformssoaction

Query-parameters:

attrs

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/vpnformssoaction?**attrs=property-name1,property-name2**

Use this query parameter to specify the resource details that you want to retrieve.

filter

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/vpnformssoaction?**filter=property-name1:property-val1,property-name2:property-val2**

Use this query-parameter to get the filtered set of vpnformssoaction resources configured on Citrix ADC. Filtering can be done on any of the properties of the resource.

view

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/vpnformssoaction?**view=summary**

Note: By default, the retrieved results are displayed in detail view (?view=detail).

pagination

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/vpnformssoaction?**pagesize=#no;pageno=#no**

Use this query-parameter to get the vpnformssoaction resources in chunks.

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "vpnformssoaction": [ {

"name":<String_value>,

"actionurl":<String_value>,

"userfield":<String_value>,

"passwdfield":<String_value>,

"responsesize":<Double_value>,

"namevaluepair":<String_value>,

"nvtype":<String_value>,

"ssosuccessrule":<String_value>,

"submitmethod":<String_value>

}]}
```

get

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/vpnformssoaction/name_value<String>

Query-parameters:

attrs

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/vpnformssoaction/name_value<String>?**attrs=property-name1,property-name2**

Use this query parameter to specify the resource details that you want to retrieve.

view

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/vpnformssoaction/name_value<String>?**view=summary**

Note: By default, the retrieved results are displayed in detail view (?view=detail).

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "vpnformssoaction": [ {

"name":<String_value>,

"actionurl":<String_value>,

"userfield":<String_value>,

"passwdfield":<String_value>,

"responsesize":<Double_value>,

"namevaluepair":<String_value>,

"nvtype":<String_value>,

"ssosuccessrule":<String_value>,
```

```
"submitmethod":<String_value>

}]}
```

count

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/vpnformssoaction?**count=yes**

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "vpnformssoaction": [ { "__count": "#no"} ] }
```

vpnglobal_appcontroller_binding

Binding object showing the appcontroller that can be bound to vpnglobal.

Properties

(click to see Operations)

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
gotopriorityexpression	<String>	Read-write	Applicable only to advance vpn session policy. An expression

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
			or other value specifying the priority of the next policy which will get evaluated if the current policy rule evaluates to TRUE.
appcontroller	<String>	Read-write	Configured App Controller server.
__count	<Double>	Read-write	count parameter

Operations

(click to see Properties)

ADD:| DELETE:| GET| GET (ALL)| COUNT

Some options that you can use for each operations:

- **Getting warnings in response:**NITRO allows you to get warnings in an operation by specifying the "warning" query parameter as "yes". For example, to get warnings while connecting to the Citrix ADC appliance, the URL is as follows:

  http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/login?warning=yes

  If any, the warnings are displayed in the response payload with the HTTP code "209 X-NITRO-WARNING".

- **Authenticated access for individual NITRO operations:**NITRO allows you to logon to the Citrix ADC appliance to perform individual operations. You can use this option instead of creating a NITRO session (using the login object) and then using that session to perform all operations,

  To do this, you must specify the username and password in the request header of the NITRO request as follows:

  X-NITRO-USER:<username>

  X-NITRO-PASS:<password>

  Note: In such cases, make sure that the request header DOES not include the following:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue>

`*Note:` *

Mandatory parameters are marked in redand placeholder content is marked in <green>.

add

URL:http://<Citrix-ADC-IP-address(NSIP)/nitro/v1/config/vpnglobal_appcontroller_binding

HTTP Method:PUT

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Content-Type:application/json

Request Payload:

```
{
"vpnglobal_appcontroller_binding":{
"appcontroller":<String_value>,
"gotopriorityexpression":<String_value>
}}
```

Response:

HTTP Status Code on Success: 201 Created HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error

delete

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/vpnglobal_appcontroller_binding

Query-parameters:

args

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/vpnglobal_appcontroller_binding?**args=appcontroller:<String_value>**

HTTP Method:DELETE

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue>

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error

get

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/vpnglobal_appcontroller_binding

Query-parameters:

filter

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/vpnglobal_appcontroller_binding?**filter=property-name1:property-value1,property-name2:property-value2**

Use this query-parameter to get the filtered set of vpnglobal_appcontroller_binding resources configured on Citrix ADC. Filtering can be done on any of the properties of the resource.

pagination

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/vpnglobal_appcontroller_binding?**pagesize=#no;pageno=#no**

Use this query-parameter to get the vpnglobal_appcontroller_binding resources in chunks.

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "vpnglobal_appcontroller_binding": [ {

"gotopriorityexpression":<String_value>,

"appcontroller":<String_value>

}]}
```

get (all)

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/vpnglobal_appcontroller_binding

Query-parameters:

bulkbindings

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/vpnglobal_appcontroller_binding?**bulkbindings=yes**

NITRO allows you to fetch bindings in bulk.

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "vpnglobal_appcontroller_binding": [ {

"gotopriorityexpression":<String_value>,

"appcontroller":<String_value>

}]}
```

count

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/vpnglobal_appcontroller_binding?**count=yes**

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "vpnglobal_appcontroller_binding": [ { "__count": "#no"} ] }
```

vpnglobal_auditnslogpolicy_binding

Binding object showing the auditnslogpolicy that can be bound to vpnglobal.

Properties

(click to see Operations)

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
priority	<Double>	Read-write	Integer specifying the policy's priority. The lower the priority number, the higher the policy's priority. Maximum value for default syntax policies is 2147483647 and for classic policies is 64000. Minimum value = 0 Maximum value = 2147483647
policyname	<String>	Read-write	The name of the policy.
gotopriorityexpression	<String>	Read-write	Applicable only to advance vpn session policy. An expression or other value specifying the priority of the next policy which will get evaluated if the current policy rule evaluates to TRUE.
secondary	<Boolean>	Read-write	Bind the authentication policy as the secondary policy to use in a

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
			two-factor configuration. A user must then authenticate not only to a primary authentication server but also to a secondary authentication server. User groups are aggregated across both authentication servers. The user name must be exactly the same on both authentication servers, but the authentication servers can require different passwords.
groupextraction	<Boolean>	Read-write	Bind the Authentication policy to a tertiary chain which will be used only for group extraction. The user will not authenticate against this server, and this will only be called it primary and/or secondary authentication has succeeded.
__count	<Double>	Read-write	count parameter

Operations

(click to see Properties)

ADD:| DELETE:| GET| GET (ALL)| COUNT

Some options that you can use for each operations:

- **Getting warnings in response:**NITRO allows you to get warnings in an operation by specifying the "warning" query parameter as "yes". For example, to get warnings while connecting to the Citrix ADC appliance, the URL is as follows:

  http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/login?warning=yes

  If any, the warnings are displayed in the response payload with the HTTP code "209 X-NITRO-WARNING".

- **Authenticated access for individual NITRO operations:**NITRO allows you to logon to the Citrix ADC appliance to perform individual operations. You can use this option instead of creating a NITRO session (using the login object) and then using that session to perform all operations,

  To do this, you must specify the username and password in the request header of the NITRO request as follows:

  X-NITRO-USER:<username>

  X-NITRO-PASS:<password>

  Note: In such cases, make sure that the request header DOES not include the following:

  Cookie:NITRO_AUTH_TOKEN=<tokenvalue>

`*Note:`*

Mandatory parameters are marked in redand placeholder content is marked in <green>.

add

URL:http://<Citrix-ADC-IP-address(NSIP)/nitro/v1/config/vpnglobal_auditnslogpolicy_binding

HTTP Method:PUT

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Content-Type:application/json

Request Payload:

```
{

"vpnglobal_auditnslogpolicy_binding":{

"policyname":<String_value>,

"priority":<Double_value>,

"secondary":<Boolean_value>,

"groupextraction":<Boolean_value>,

"gotopriorityexpression":<String_value>

}}
```

Response:

HTTP Status Code on Success: 201 Created HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error

delete

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/vpnglobal_auditnslogpolicy_binding

Query-parameters:

args

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/vpnglobal_auditnslogpolicy_binding?**args=policyname:<String_value>,secondary:<Boolean_value>,groupextraction:<Boolean_value>**

HTTP Method:DELETE

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue>

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error

get

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/vpnglobal_auditnslogpolicy_binding

Query-parameters:

filter

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/vpnglobal_auditnslogpolicy_binding?**filter=property-name1:property-value1,property-name2:property-value2**

Use this query-parameter to get the filtered set of vpnglobal_auditnslogpolicy_binding resources configured on Citrix ADC. Filtering can be done on any of the properties of the resource.

pagination

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/vpnglobal_auditnslogpolicy_binding?**pagesize=#no;pageno=#no**

Use this query-parameter to get the vpnglobal_auditnslogpolicy_binding resources in chunks.

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "vpnglobal_auditnslogpolicy_binding": [ {
```

```
"priority":<Double_value>,

"policyname":<String_value>,

"gotopriorityexpression":<String_value>,

"secondary":<Boolean_value>,

"groupextraction":<Boolean_value>

}]}
```

get (all)

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/vpnglobal_auditnslogpolicy_binding

Query-parameters:

bulkbindings

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/vpnglobal_auditnslogpolicy_binding?**bulkbindings=yes**

NITRO allows you to fetch bindings in bulk.

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "vpnglobal_auditnslogpolicy_binding": [ {

"priority":<Double_value>,

"policyname":<String_value>,

"gotopriorityexpression":<String_value>,

"secondary":<Boolean_value>,

"groupextraction":<Boolean_value>

}]}
```

count

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/vpnglobal_auditnslogpolicy_binding?**count=yes**

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "vpnglobal_auditnslogpolicy_binding": [ { "__count": "#no"} ] }
```

vpnglobal_auditsyslogpolicy_binding

Binding object showing the auditsyslogpolicy that can be bound to vpnglobal.

Properties

(click to see Operations)

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
priority	<Double>	Read-write	Integer specifying the policy's priority. The lower the priority number, the higher the policy's priority. Maximum value for default syntax policies is 2147483647 and for classic policies is 64000. Minimum value = 0 Maximum value = 2147483647
policyname	<String>	Read-write	The name of the policy.
gotopriorityexpression	<String>	Read-write	Applicable only to advance vpn session policy. An expression or other value specifying the priority of the next policy which will get evaluated if the current policy rule evaluates to TRUE.
secondary	<Boolean>	Read-write	Bind the authentication policy as the secondary policy to use in a

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
			two-factor configuration. A user must then authenticate not only to a primary authentication server but also to a secondary authentication server. User groups are aggregated across both authentication servers. The user name must be exactly the same on both authentication servers, but the authentication servers can require different passwords.
groupextraction	<Boolean>	Read-write	Bind the Authentication policy to a tertiary chain which will be used only for group extraction. The user will not authenticate against this server, and this will only be called it primary and/or secondary authentication has succeeded.
__count	<Double>	Read-write	count parameter

Operations

(click to see Properties)

ADD:| DELETE:| GET| GET (ALL)| COUNT

Some options that you can use for each operations:

- **Getting warnings in response:**NITRO allows you to get warnings in an operation by specifying the "warning" query parameter as "yes". For example, to get warnings while connecting to the Citrix ADC appliance, the URL is as follows:

  http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/login?warning=yes

  If any, the warnings are displayed in the response payload with the HTTP code "209 X-NITRO-WARNING".

- **Authenticated access for individual NITRO operations:**NITRO allows you to logon to the Citrix ADC appliance to perform individual operations. You can use this option instead of creating a NITRO session (using the login object) and then using that session to perform all operations,

  To do this, you must specify the username and password in the request header of the NITRO request as follows:

  X-NITRO-USER:<username>

  X-NITRO-PASS:<password>

  Note: In such cases, make sure that the request header DOES not include the following:

  Cookie:NITRO_AUTH_TOKEN=<tokenvalue>

`*Note:`*

Mandatory parameters are marked in redand placeholder content is marked in <green>.

add

URL:http://<Citrix-ADC-IP-address(NSIP)/nitro/v1/config/vpnglobal_auditsyslogpolicy_binding

HTTP Method:PUT

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Content-Type:application/json

Request Payload:

```
{

"vpnglobal_auditsyslogpolicy_binding":{

"policyname":<String_value>,

"priority":<Double_value>,

"secondary":<Boolean_value>,

"groupextraction":<Boolean_value>,

"gotopriorityexpression":<String_value>

}}
```

Response:

HTTP Status Code on Success: 201 Created HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error

delete

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/vpnglobal_auditsyslogpolicy_binding

Query-parameters:

args

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/vpnglobal_auditsyslogpolicy_binding?**args=policyname:<String_value>,secondary:<Boolean_value>,groupextraction:<Boolean_value>**

HTTP Method:DELETE

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue>

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error

get

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/vpnglobal_auditsyslogpolicy_binding

Query-parameters:

filter

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/vpnglobal_auditsyslogpolicy_binding?**filter=property-name1:property-value1,property-name2:property-value2**

Use this query-parameter to get the filtered set of vpnglobal_auditsyslogpolicy_binding resources configured on Citrix ADC. Filtering can be done on any of the properties of the resource.

pagination

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/vpnglobal_auditsyslogpolicy_binding?**pagesize=#no;pageno=#no**

Use this query-parameter to get the vpnglobal_auditsyslogpolicy_binding resources in chunks.

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "vpnglobal_auditsyslogpolicy_binding": [ {
```

```
"priority":<Double_value>,

"policyname":<String_value>,

"gotopriorityexpression":<String_value>,

"secondary":<Boolean_value>,

"groupextraction":<Boolean_value>

}]}
```

get (all)

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/vpnglobal_auditsyslogpolicy_binding

Query-parameters:

bulkbindings

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/vpnglobal_auditsyslogpolicy_binding?**bulkbindings=yes**

NITRO allows you to fetch bindings in bulk.

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "vpnglobal_auditsyslogpolicy_binding": [ {

"priority":<Double_value>,

"policyname":<String_value>,

"gotopriorityexpression":<String_value>,

"secondary":<Boolean_value>,

"groupextraction":<Boolean_value>

}]}
```

count

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/vpnglobal_auditsyslogpolicy_binding?**count=yes**

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "vpnglobal_auditsyslogpolicy_binding": [ { "__count": "#no"} ] }
```

vpnglobal_authenticationcertpolicy_binding

Binding object showing the authenticationcertpolicy that can be bound to vpnglobal.

Properties

(click to see Operations)

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
priority	<Double>	Read-write	Integer specifying the policy's priority. The lower the priority number, the higher the policy's priority. Maximum value for default syntax policies is 2147483647 and for classic policies is 64000. Minimum value = 0 Maximum value = 2147483647
policyname	<String>	Read-write	The name of the policy.
gotopriorityexpression	<String>	Read-write	Applicable only to advance vpn session policy. An expression or other value specifying the priority of the next policy which will get evaluated if the current policy rule evaluates to TRUE.
secondary	<Boolean>	Read-write	Bind the authentication policy as the secondary policy to use in a

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
			two-factor configuration. A user must then authenticate not only to a primary authentication server but also to a secondary authentication server. User groups are aggregated across both authentication servers. The user name must be exactly the same on both authentication servers, but the authentication servers can require different passwords.
groupextraction	<Boolean>	Read-write	Bind the Authentication policy to a tertiary chain which will be used only for group extraction. The user will not authenticate against this server, and this will only be called it primary and/or secondary authentication has succeeded.
__count	<Double>	Read-write	count parameter

Operations

(click to see Properties)

ADD:| DELETE:| GET| GET (ALL)| COUNT

Some options that you can use for each operations:

- **Getting warnings in response:**NITRO allows you to get warnings in an operation by specifying the "warning" query parameter as "yes". For example, to get warnings while connecting to the Citrix ADC appliance, the URL is as follows:

  http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/login?warning=yes

  If any, the warnings are displayed in the response payload with the HTTP code "209 X-NITRO-WARNING".

- **Authenticated access for individual NITRO operations:**NITRO allows you to logon to the Citrix ADC appliance to perform individual operations. You can use this option instead of creating a NITRO session (using the login object) and then using that session to perform all operations,

  To do this, you must specify the username and password in the request header of the NITRO request as follows:

  X-NITRO-USER:<username>

  X-NITRO-PASS:<password>

  Note: In such cases, make sure that the request header DOES not include the following:

  Cookie:NITRO_AUTH_TOKEN=<tokenvalue>

`*Note:`*

Mandatory parameters are marked in redand placeholder content is marked in <green>.

add

URL:http://<Citrix-ADC-IP-address(NSIP)/nitro/v1/config/vpnglobal_authenticationcertpolicy_binding

HTTP Method:PUT

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Content-Type:application/json

Request Payload:

```
{

"vpnglobal_authenticationcertpolicy_binding":{

"policyname":<String_value>,

"priority":<Double_value>,

"secondary":<Boolean_value>,

"groupextraction":<Boolean_value>,

"gotopriorityexpression":<String_value>

}}
```

Response:

HTTP Status Code on Success: 201 Created HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error

delete

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/vpnglobal_authenticationcertpolicy_binding

Query-parameters:

args

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/vpnglobal_authenticationcertpolicy_binding?**args=policyname:<String_value>,secondary:<Boolean_value>,groupextraction:<Boolean_value>**

HTTP Method:DELETE

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue>

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error

get

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/vpnglobal_authenticationcertpolicy_binding

Query-parameters:

filter

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/vpnglobal_authenticationcertpolicy_binding?**filter=property-name1:property-value1,property-name2:property-value2**

Use this query-parameter to get the filtered set of vpnglobal_authenticationcertpolicy_binding resources configured on Citrix ADC. Filtering can be done on any of the properties of the resource.

pagination

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/vpnglobal_authenticationcertpolicy_binding?**pagesize=#no;pageno=#no**

Use this query-parameter to get the vpnglobal_authenticationcertpolicy_binding resources in chunks.

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "vpnglobal_authenticationcertpolicy_binding": [ {

"priority":<Double_value>,

"policyname":<String_value>,

"gotopriorityexpression":<String_value>,

"secondary":<Boolean_value>,

"groupextraction":<Boolean_value>

}]}
```

get (all)

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/vpnglobal_authenticationcertpolicy_binding

Query-parameters:

bulkbindings

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/vpnglobal_authenticationcertpolicy_binding?**bulkbindings=yes**

NITRO allows you to fetch bindings in bulk.

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "vpnglobal_authenticationcertpolicy_binding": [ {

"priority":<Double_value>,

"policyname":<String_value>,

"gotopriorityexpression":<String_value>,

"secondary":<Boolean_value>,

"groupextraction":<Boolean_value>

}]}
```

count

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/vpnglobal_authenticationcertpolicy_binding?**count=yes**

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "vpnglobal_authenticationcertpolicy_binding": [ { "__count": "#no"} ] }
```

vpnglobal_authenticationldappolicy_binding

Binding object showing the authenticationldappolicy that can be bound to vpnglobal.

Properties

(click to see Operations)

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
priority	<Double>	Read-write	Integer specifying the policy's priority. The lower the priority number, the higher the policy's priority. Maximum value for default syntax policies is 2147483647 and for classic policies is 64000. Minimum value = 0 Maximum value = 2147483647
policyname	<String>	Read-write	The name of the policy.
gotopriorityexpression	<String>	Read-write	Applicable only to advance vpn session policy. An expression or other value specifying the priority of the next policy which will get evaluated if the current policy rule evaluates to TRUE.

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
secondary	<Boolean>	Read-write	Bind the authentication policy as the secondary policy to use in a two-factor configuration. A user must then authenticate not only to a primary authentication server but also to a secondary authentication server. User groups are aggregated across both authentication servers. The user name must be exactly the same on both authentication servers, but the authentication servers can require different passwords.
groupextraction	<Boolean>	Read-write	Bind the Authentication policy to a tertiary chain which will be used only for group extraction. The user will not authenticate against this server, and this will only be called it primary and/or secondary authentication has succeeded.

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
__count	<Double>	Read-write	count parameter

Operations

(click to see Properties)

ADD:| DELETE:| GET| GET (ALL)| COUNT

Some options that you can use for each operations:

- **Getting warnings in response:**NITRO allows you to get warnings in an operation by specifying the "warning" query parameter as "yes". For example, to get warnings while connecting to the Citrix ADC appliance, the URL is as follows:

  http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/login?warning=yes

  If any, the warnings are displayed in the response payload with the HTTP code "209 X-NITRO-WARNING".
- **Authenticated access for individual NITRO operations:**NITRO allows you to logon to the Citrix ADC appliance to perform individual operations. You can use this option instead of creating a NITRO session (using the login object) and then using that session to perform all operations,

  To do this, you must specify the username and password in the request header of the NITRO request as follows:

  X-NITRO-USER:<username>

  X-NITRO-PASS:<password>

  Note: In such cases, make sure that the request header DOES not include the following:

  Cookie:NITRO_AUTH_TOKEN=<tokenvalue>

`*Note:`*

Mandatory parameters are marked in redand placeholder content is marked in <green>.

add

URL:http://<Citrix-ADC-IP-address(NSIP)/nitro/v1/config/vpnglobal_authenticationldappolicy_binding

HTTP Method:PUT

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Content-Type:application/json

Request Payload:

```
{
"vpnglobal_authenticationldappolicy_binding":{
"policyname":<String_value>,
"priority":<Double_value>,
"secondary":<Boolean_value>,
"groupextraction":<Boolean_value>,
"gotopriorityexpression":<String_value>
}}
```

Response:

HTTP Status Code on Success: 201 Created HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error

delete

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/vpnglobal_authenticationldappolicy_binding

Query-parameters:

args

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/vpnglobal_authenticationldappolicy_binding?**args=policyname:<String_value>,secondary:<Boolean_value>,groupextraction:<Boolean_value>**

HTTP Method:DELETE

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue>

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error

get

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/vpnglobal_authenticationldappolicy_binding

Query-parameters:

filter

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/vpnglobal_authenticationldappolicy_binding?**filter=property-name1:property-value1,property-name2:property-value2**

Use this query-parameter to get the filtered set of vpnglobal_authenticationldappolicy_binding resources configured on Citrix ADC. Filtering can be done on any of the properties of the resource.

pagination

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/vpnglobal_authenticationldappolicy_binding?**pagesize=#no;pageno=#no**

Use this query-parameter to get the vpnglobal_authenticationldappolicy_binding resources in chunks.

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "vpnglobal_authenticationldappolicy_binding": [ {

"priority":<Double_value>,

"policyname":<String_value>,

"gotopriorityexpression":<String_value>,

"secondary":<Boolean_value>,

"groupextraction":<Boolean_value>

}]}
```

get (all)

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/vpnglobal_authenticationldappolicy_binding

Query-parameters:

bulkbindings

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/vpnglobal_authenticationldappolicy_binding?**bulkbindings=yes**

NITRO allows you to fetch bindings in bulk.

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "vpnglobal_authenticationldappolicy_binding": [ {

"priority":<Double_value>,

"policyname":<String_value>,

"gotopriorityexpression":<String_value>,

"secondary":<Boolean_value>,

"groupextraction":<Boolean_value>

}]}
```

count

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/vpnglobal_authenticationldappolicy_binding?**count=yes**

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "vpnglobal_authenticationldappolicy_binding": [ { "__count": "#no"} ] }
```

vpnglobal_authenticationlocalpolicy_binding

Binding object showing the authenticationlocalpolicy that can be bound to vpnglobal.

Properties

(click to see Operations)

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
priority	<Double>	Read-write	Integer specifying the policy's priority. The lower the priority number, the higher the policy's priority. Maximum value for default syntax policies is 2147483647 and for classic policies is 64000. Minimum value = 0 Maximum value = 2147483647
policyname	<String>	Read-write	The name of the policy.
gotopriorityexpression	<String>	Read-write	Applicable only to advance vpn session policy. An expression or other value specifying the priority

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
			of the next policy which will get evaluated if the current policy rule evaluates to TRUE.
secondary	<Boolean>	Read-write	Bind the authentication policy as the secondary policy to use in a two-factor configuration. A user must then authenticate not only to a primary authentication server but also to a secondary authentication server. User groups are aggregated across both authentication servers. The user name must be exactly the same on both authentication servers, but the authentication servers can require different passwords.
groupextraction	<Boolean>	Read-write	Bind the Authentication policy to a tertiary chain which will be used only for group extraction. The user will not authenticate against this server, and this will only be

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
			called it primary and/ or secondary authentication has succeeded.
__count	<Double>	Read-write	count parameter

Operations

(click to see Properties)

ADD:| DELETE:| GET| GET (ALL)| COUNT

Some options that you can use for each operations:

- **Getting warnings in response:**NITRO allows you to get warnings in an operation by specifying the "warning" query parameter as "yes". For example, to get warnings while connecting to the Citrix ADC appliance, the URL is as follows:

  http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/login?warning=yes

  If any, the warnings are displayed in the response payload with the HTTP code "209 X-NITRO-WARNING".

- **Authenticated access for individual NITRO operations:**NITRO allows you to logon to the Citrix ADC appliance to perform individual operations. You can use this option instead of creating a NITRO session (using the login object) and then using that session to perform all operations,

  To do this, you must specify the username and password in the request header of the NITRO request as follows:

  X-NITRO-USER:<username>

  X-NITRO-PASS:<password>

  Note: In such cases, make sure that the request header DOES not include the following:

  Cookie:NITRO_AUTH_TOKEN=<tokenvalue>

`*Note:`*

Mandatory parameters are marked in redand placeholder content is marked in <green>.

add

URL:http://<Citrix-ADC-IP-address(NSIP)/nitro/v1/config/vpnglobal_authenticationlocalpolicy_binding

HTTP Method:PUT

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Content-Type:application/json

Request Payload:

```
{

"vpnglobal_authenticationlocalpolicy_binding":{

"policyname":<String_value>,

"priority":<Double_value>,

"secondary":<Boolean_value>,

"groupextraction":<Boolean_value>,

"gotopriorityexpression":<String_value>

}}
```

Response:

HTTP Status Code on Success: 201 Created HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error

delete

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/vpnglobal_authenticationlocalpolicy_binding

Query-parameters:

args

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/vpnglobal_authenticationlocalpolicy_binding?**args=policyname:<String_value>,secondary:<Boolean_value>,groupextraction:<Boolean_value>**

HTTP Method:DELETE

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue>

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error

get

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/vpnglobal_authenticationlocalpolicy_binding

Query-parameters:

filter

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/vpnglobal_authenticationlocalpolicy_binding?**filter=property-name1:property-value1,property-name2:property-value2**

Use this query-parameter to get the filtered set of vpnglobal_authenticationlocalpolicy_binding resources configured on Citrix ADC. Filtering can be done on any of the properties of the resource.

pagination

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/vpnglobal_authenticationlocalpolicy_binding?**pagesize=#no;pageno=#no**

Use this query-parameter to get the vpnglobal_authenticationlocalpolicy_binding resources in chunks.

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "vpnglobal_authenticationlocalpolicy_binding": [ {

"priority":<Double_value>,

"policyname":<String_value>,

"gotopriorityexpression":<String_value>,

"secondary":<Boolean_value>,

"groupextraction":<Boolean_value>

}]}
```

get (all)

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/vpnglobal_authenticationlocalpolicy_binding

Query-parameters:

bulkbindings

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/vpnglobal_authenticationlocalpolicy_binding?**bulkbindings=yes**

NITRO allows you to fetch bindings in bulk.

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "vpnglobal_authenticationlocalpolicy_binding": [ {

"priority":<Double_value>,

"policyname":<String_value>,

"gotopriorityexpression":<String_value>,

"secondary":<Boolean_value>,

"groupextraction":<Boolean_value>

}]}
```

count

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/vpnglobal_authenticationlocalpolicy_binding?**count=yes**

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "vpnglobal_authenticationlocalpolicy_binding": [ { "__count": "#no"} ] }
```

vpnglobal_authenticationnegotiatepolicy_binding

Binding object showing the authenticationnegotiatepolicy that can be bound to vpnglobal.

Properties

(click to see Operations)

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
priority	<Double>	Read-write	Integer specifying the policy's priority. The lower the priority number, the higher the policy's priority. Maximum value for default syntax policies is 2147483647 and for classic policies is 64000. Minimum value = 0 Maximum value = 2147483647
policyname	<String>	Read-write	The name of the policy.

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
gotopriorityexpression	<String>	Read-write	Applicable only to advance vpn session policy. An expression or other value specifying the priority of the next policy which will get evaluated if the current policy rule evaluates to TRUE.
secondary	<Boolean>	Read-write	Bind the authentication policy as the secondary policy to use in a two-factor configuration. A user must then authenticate not only to a primary authentication server but also to a secondary authentication server. User groups are aggregated across both authentication servers. The user name must be exactly the same on both authentication servers, but the authentication servers can require different passwords.
groupextraction	<Boolean>	Read-write	Bind the Authentication policy to a tertiary chain which will be used

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
			only for group extraction. The user will not authenticate against this server, and this will only be called it primary and/or secondary authentication has succeeded.
__count	<Double>	Read-write	count parameter

Operations

(click to see Properties)

ADD:| DELETE:| GET| GET (ALL)| COUNT

Some options that you can use for each operations:

- **Getting warnings in response:**NITRO allows you to get warnings in an operation by specifying the "warning" query parameter as "yes". For example, to get warnings while connecting to the Citrix ADC appliance, the URL is as follows:

  http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/login?warning=yes

  If any, the warnings are displayed in the response payload with the HTTP code "209 X-NITRO-WARNING".

- **Authenticated access for individual NITRO operations:**NITRO allows you to logon to the Citrix ADC appliance to perform individual operations. You can use this option instead of creating a NITRO session (using the login object) and then using that session to perform all operations,

  To do this, you must specify the username and password in the request header of the NITRO request as follows:

  X-NITRO-USER:<username>

  X-NITRO-PASS:<password>

  Note: In such cases, make sure that the request header DOES not include the following:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue>

`*Note:`*

Mandatory parameters are marked in redand placeholder content is marked in <green>.

add

URL:http://<Citrix-ADC-IP-address(NSIP)/nitro/v1/config/vpnglobal_authenticationnegotiatepolicy_binding

HTTP Method:PUT

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Content-Type:application/json

Request Payload:

```
{

"vpnglobal_authenticationnegotiatepolicy_binding":{

"policyname":<String_value>,

"priority":<Double_value>,

"secondary":<Boolean_value>,

"groupextraction":<Boolean_value>,

"gotopriorityexpression":<String_value>

}}
```

Response:

HTTP Status Code on Success: 201 Created HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error

delete

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/vpnglobal_authenticationnegotiatepolicy_binding

Query-parameters:

args

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/vpnglobal_authenticationnegotiatepolicy_binding?**args=policyname:<String_value>,secondary:<Boolean_value>,groupextraction:<Boolean_value>**

HTTP Method:DELETE

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue>

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error

get

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/vpnglobal_authenticationnegotiatepolicy_binding

Query-parameters:

filter

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/vpnglobal_authenticationnegotiatepolicy_binding?**filter=property-name1:property-value1,property-name2:property-value2**

Use this query-parameter to get the filtered set of vpnglobal_authenticationnegotiatepolicy_binding resources configured on Citrix ADC. Filtering can be done on any of the properties of the resource.

pagination

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/vpnglobal_authenticationnegotiatepolicy_binding?**pagesize=#no;pageno=#no**

Use this query-parameter to get the vpnglobal_authenticationnegotiatepolicy_binding resources in chunks.

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "vpnglobal_authenticationnegotiatepolicy_binding": [ {

"priority":<Double_value>,

"policyname":<String_value>,

"gotopriorityexpression":<String_value>,

"secondary":<Boolean_value>,

"groupextraction":<Boolean_value>

}]}
```

get (all)

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/vpnglobal_authenticationnegotiatepolicy_binding

Query-parameters:

bulkbindings

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/vpnglobal_authenticationnegotiatepolicy_binding?**bulkbindings=yes**

NITRO allows you to fetch bindings in bulk.

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "vpnglobal_authenticationnegotiatepolicy_binding": [ {

"priority":<Double_value>,

"policyname":<String_value>,

"gotopriorityexpression":<String_value>,

"secondary":<Boolean_value>,

"groupextraction":<Boolean_value>

}]}
```

count

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/vpnglobal_authenticationnegotiatepolicy_binding?**count=yes**

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "vpnglobal_authenticationnegotiatepolicy_binding": [ { "__count":
"#no"} ] }
```

vpnglobal_authenticationpolicy_binding

Binding object showing the authenticationpolicy that can be bound to vpnglobal.

Properties

(click to see Operations)

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
priority	<Double>	Read-write	Integer specifying the policy's priority. The lower the priority number, the higher the policy's priority. Maximum value for default syntax policies is 2147483647 and for classic policies is 64000. Minimum value = 0

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
			Maximum value = 2147483647
policyname	<String>	Read-write	The name of the policy.
gotopriorityexpression	<String>	Read-write	Applicable only to advance vpn session policy. An expression or other value specifying the priority of the next policy which will get evaluated if the current policy rule evaluates to TRUE.
secondary	<Boolean>	Read-write	Bind the authentication policy as the secondary policy to use in a two-factor configuration. A user must then authenticate not only to a primary authentication server but also to a secondary authentication server. User groups are aggregated across both authentication servers. The user name must be exactly the same on both authentication servers, but the authentication

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
			servers can require different passwords.
groupextraction	<Boolean>	Read-write	Bind the Authentication policy to a tertiary chain which will be used only for group extraction. The user will not authenticate against this server, and this will only be called it primary and/or secondary authentication has succeeded.
__count	<Double>	Read-write	count parameter

Operations

(click to see Properties)

ADD:| DELETE:| GET| GET (ALL)| COUNT

Some options that you can use for each operations:

- **Getting warnings in response:**NITRO allows you to get warnings in an operation by specifying the "warning" query parameter as "yes". For example, to get warnings while connecting to the Citrix ADC appliance, the URL is as follows:

  http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/login?warning=yes

  If any, the warnings are displayed in the response payload with the HTTP code "209 X-NITRO-WARNING".
- **Authenticated access for individual NITRO operations:**NITRO allows you to logon to the Citrix ADC appliance to perform individual operations. You can use this option instead of creating a NITRO session (using the login object) and then using that session to perform all operations,

To do this, you must specify the username and password in the request header of the NITRO request as follows:

X-NITRO-USER:<username>

X-NITRO-PASS:<password>

Note: In such cases, make sure that the request header DOES not include the following:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue>

`*Note:`*

Mandatory parameters are marked in redand placeholder content is marked in <green>.

add

URL:http://<Citrix-ADC-IP-address(NSIP)/nitro/v1/config/vpnglobal_authenticationpolicy_binding

HTTP Method:PUT

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Content-Type:application/json

Request Payload:

```
{

"vpnglobal_authenticationpolicy_binding":{

"policyname":<String_value>,

"priority":<Double_value>,

"secondary":<Boolean_value>,

"groupextraction":<Boolean_value>,

"gotopriorityexpression":<String_value>

}}
```

Response:

HTTP Status Code on Success: 201 Created HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error

delete

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/vpnglobal_authenticationpolicy_binding

Query-parameters:

args

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/vpnglobal_authenticationpolicy_binding?**args=policyname:<String_value>,secondary:<Boolean_value>,groupextraction:<Boolean_value>**

HTTP Method:DELETE

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue>

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error

get

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/vpnglobal_authenticationpolicy_binding

Query-parameters:

filter

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/vpnglobal_authenticationpolicy_binding?**filter=property-name1:property-value1,property-name2:property-value2**

Use this query-parameter to get the filtered set of vpnglobal_authenticationpolicy_binding resources configured on Citrix ADC. Filtering can be done on any of the properties of the resource.

pagination

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/vpnglobal_authenticationpolicy_binding?**pagesize=#no;pageno=#no**

Use this query-parameter to get the vpnglobal_authenticationpolicy_binding resources in chunks.

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "vpnglobal_authenticationpolicy_binding": [ {

"priority":<Double_value>,

"policyname":<String_value>,

"gotopriorityexpression":<String_value>,

"secondary":<Boolean_value>,

"groupextraction":<Boolean_value>

}]}
```

get (all)

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/vpnglobal_authenticationpolicy_binding

Query-parameters:

bulkbindings

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/vpnglobal_authenticationpolicy_binding?**bulkbindings=yes**

NITRO allows you to fetch bindings in bulk.

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "vpnglobal_authenticationpolicy_binding": [ {

"priority":<Double_value>,

"policyname":<String_value>,

"gotopriorityexpression":<String_value>,

"secondary":<Boolean_value>,

"groupextraction":<Boolean_value>

}]}
```

count

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/vpnglobal_authenticationpolicy_binding?**count=yes**

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "vpnglobal_authenticationpolicy_binding": [ { "__count": "#no"} ] }
```

vpnglobal_authenticationradiuspolicy_binding

Binding object showing the authenticationradiuspolicy that can be bound to vpnglobal.

Properties

(click to see Operations)

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
priority	<Double>	Read-write	Integer specifying the policy's priority. The lower the priority number, the higher the policy's priority. Maximum value for default syntax policies is 2147483647 and for classic policies is 64000.

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
			Minimum value = 0 Maximum value = 2147483647
policyname	<String>	Read-write	The name of the policy.
gotopriorityexpression	<String>	Read-write	Applicable only to advance vpn session policy. An expression or other value specifying the priority of the next policy which will get evaluated if the current policy rule evaluates to TRUE.
secondary	<Boolean>	Read-write	Bind the authentication policy as the secondary policy to use in a two-factor configuration. A user must then authenticate not only to a primary authentication server but also to a secondary authentication server. User groups are aggregated across both authentication servers. The user name must be exactly the same on both authentication servers, but the

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
			authentication servers can require different passwords.
groupextraction	<Boolean>	Read-write	Bind the Authentication policy to a tertiary chain which will be used only for group extraction. The user will not authenticate against this server, and this will only be called it primary and/or secondary authentication has succeeded.
__count	<Double>	Read-write	count parameter

Operations

(click to see Properties)

ADD:| DELETE:| GET| GET (ALL)| COUNT

Some options that you can use for each operations:

- **Getting warnings in response:**NITRO allows you to get warnings in an operation by specifying the "warning" query parameter as "yes". For example, to get warnings while connecting to the Citrix ADC appliance, the URL is as follows:

  http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/login?warning=yes

  If any, the warnings are displayed in the response payload with the HTTP code "209 X-NITRO-WARNING".
- **Authenticated access for individual NITRO operations:**NITRO allows you to logon to the Citrix ADC appliance to perform individual operations. You can use this option instead of creating a NITRO session (using the login object) and then using that session to perform all operations,

To do this, you must specify the username and password in the request header of the NITRO request as follows:

X-NITRO-USER:<username>

X-NITRO-PASS:<password>

Note: In such cases, make sure that the request header DOES not include the following:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue>

`*Note:` *

Mandatory parameters are marked in redand placeholder content is marked in <green>.

add

URL:http://<Citrix-ADC-IP-address(NSIP)/nitro/v1/config/vpnglobal_authenticationradiuspolicy_binding

HTTP Method:PUT

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Content-Type:application/json

Request Payload:

```
{

"vpnglobal_authenticationradiuspolicy_binding":{

"policyname":<String_value>,

"priority":<Double_value>,

"secondary":<Boolean_value>,

"groupextraction":<Boolean_value>,

"gotopriorityexpression":<String_value>

}}
```

Response:

HTTP Status Code on Success: 201 Created HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error

delete

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/vpnglobal_authenticationradiuspolicy_binding

Query-parameters:

args

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/vpnglobal_authenticationradiuspolicy_binding?**args=policyname:<String_value>,secondary:<Boolean_value>,groupextraction:<Boolean_value>**

HTTP Method:DELETE

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue>

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error

get

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/vpnglobal_authenticationradiuspolicy_binding

Query-parameters:

filter

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/vpnglobal_authenticationradiuspolicy_binding?**filter=property-name1:property-value1,property-name2:property-value2**

Use this query-parameter to get the filtered set of vpnglobal_authenticationradiuspolicy_binding resources configured on Citrix ADC. Filtering can be done on any of the properties of the resource.

pagination

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/vpnglobal_authenticationradiuspolicy_binding?**pagesize=#no;pageno=#no**

Use this query-parameter to get the vpnglobal_authenticationradiuspolicy_binding resources in chunks.

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "vpnglobal_authenticationradiuspolicy_binding": [ {

"priority":<Double_value>,

"policyname":<String_value>,

"gotopriorityexpression":<String_value>,

"secondary":<Boolean_value>,

"groupextraction":<Boolean_value>

}]}
```

get (all)

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/vpnglobal_authenticationradiuspolicy_binding

Query-parameters:

bulkbindings

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/vpnglobal_authenticationradiuspolicy_binding?**bulkbindings=yes**

NITRO allows you to fetch bindings in bulk.

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "vpnglobal_authenticationradiuspolicy_binding": [ {

"priority":<Double_value>,

"policyname":<String_value>,

"gotopriorityexpression":<String_value>,

"secondary":<Boolean_value>,

"groupextraction":<Boolean_value>

}]}
```

count

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/vpnglobal_authenticationradiuspolicy_binding?**count=yes**

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "vpnglobal_authenticationradiuspolicy_binding": [ { "__count":
"#no"} ] }
```

vpnglobal_authenticationsamlpolicy_binding

Binding object showing the authenticationsamlpolicy that can be bound to vpnglobal.

Properties

(click to see Operations)

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
priority	<Double>	Read-write	Integer specifying the policy's priority. The lower the priority number, the higher the policy's priority.

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
			Maximum value for default syntax policies is 2147483647 and for classic policies is 64000. Minimum value = 0 Maximum value = 2147483647
policyname	<String>	Read-write	The name of the policy.
gotopriorityexpression	<String>	Read-write	Applicable only to advance vpn session policy. An expression or other value specifying the priority of the next policy which will get evaluated if the current policy rule evaluates to TRUE.
secondary	<Boolean>	Read-write	Bind the authentication policy as the secondary policy to use in a two-factor configuration. A user must then authenticate not only to a primary authentication server but also to a secondary authentication server. User groups are aggregated across both

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
			authentication servers. The user name must be exactly the same on both authentication servers, but the authentication servers can require different passwords.
groupextraction	<Boolean>	Read-write	Bind the Authentication policy to a tertiary chain which will be used only for group extraction. The user will not authenticate against this server, and this will only be called it primary and/or secondary authentication has succeeded.
__count	<Double>	Read-write	count parameter

Operations

(click to see Properties)

ADD:| DELETE:| GET| GET (ALL)| COUNT

Some options that you can use for each operations:

- **Getting warnings in response:**NITRO allows you to get warnings in an operation by specifying the "warning" query parameter as "yes". For example, to get warnings while connecting to the Citrix ADC appliance, the URL is as follows:

  http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/login?warning=yes

  If any, the warnings are displayed in the response payload with the HTTP code "209 X-NITRO-WARNING".

- **Authenticated access for individual NITRO operations:**NITRO allows you to logon to the Citrix ADC appliance to perform individual operations. You can use this option instead of creating a NITRO session (using the login object) and then using that session to perform all operations,

  To do this, you must specify the username and password in the request header of the NITRO request as follows:

  X-NITRO-USER:<username>

  X-NITRO-PASS:<password>

  Note: In such cases, make sure that the request header DOES not include the following:

  Cookie:NITRO_AUTH_TOKEN=<tokenvalue>

`*Note:`*

Mandatory parameters are marked in redand placeholder content is marked in <green>.

add

URL:http://<Citrix-ADC-IP-address(NSIP)/nitro/v1/config/vpnglobal_authenticationsamlpolicy_binding

HTTP Method:PUT

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Content-Type:application/json

Request Payload:

```
{

"vpnglobal_authenticationsamlpolicy_binding":{

"policyname":<String_value>,

"priority":<Double_value>,

"secondary":<Boolean_value>,
```

```
"groupextraction":<Boolean_value>,

"gotopriorityexpression":<String_value>

}}
```

Response:

HTTP Status Code on Success: 201 Created HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error

delete

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/vpnglobal_authenticationsamlpolicy_binding

Query-parameters:

args

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/vpnglobal_authenticationsamlpolicy_binding?**args=policyname:<String_value>,secondary:<Boolean_value>,groupextraction:<Boolean_value>**

HTTP Method:DELETE

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue>

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error

get

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/vpnglobal_authenticationsamlpolicy_binding

Query-parameters:

filter

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/vpnglobal_authenticationsamlpolicy_binding?**filter=property-name1:property-value1,property-name2:property-value2**

Use this query-parameter to get the filtered set of vpnglobal_authenticationsamlpolicy_binding resources configured on Citrix ADC. Filtering can be done on any of the properties of the resource.

pagination

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/vpnglobal_authenticationsamlpolicy_binding?**pagesize=#no;pageno=#no**

Use this query-parameter to get the vpnglobal_authenticationsamlpolicy_binding resources in chunks.

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "vpnglobal_authenticationsamlpolicy_binding": [ {

"priority":<Double_value>,

"policyname":<String_value>,

"gotopriorityexpression":<String_value>,

"secondary":<Boolean_value>,
```

```
"groupextraction":<Boolean_value>

}]}
```

get (all)

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/vpnglobal_authenticationsamlpolicy_binding

Query-parameters:

bulkbindings

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/vpnglobal_authenticationsamlpolicy_binding?**bulkbindings=yes**

NITRO allows you to fetch bindings in bulk.

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "vpnglobal_authenticationsamlpolicy_binding": [ {

"priority":<Double_value>,

"policyname":<String_value>,

"gotopriorityexpression":<String_value>,
```

```
"secondary":<Boolean_value>,

"groupextraction":<Boolean_value>

}]}
```

count

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/vpnglobal_authenticationsamlpolicy_binding?**count=yes**

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "vpnglobal_authenticationsamlpolicy_binding": [ { "__count": "#no"} ] }
```

vpnglobal_authenticationtacacspolicy_binding

Binding object showing the authenticationtacacspolicy that can be bound to vpnglobal.

Properties

(click to see Operations)

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
priority	<Double>	Read-write	Integer specifying the policy's priority. The lower the priority number, the higher the policy's priority. Maximum value for default syntax policies is 2147483647 and for classic policies is 64000. Minimum value = 0 Maximum value = 2147483647
policyname	<String>	Read-write	The name of the policy.
gotopriorityexpression	<String>	Read-write	Applicable only to advance vpn session policy. An expression or other value specifying the priority of the next policy which will get evaluated if the current policy rule evaluates to TRUE.
secondary	<Boolean>	Read-write	Bind the authentication policy as the secondary policy to use in a two-factor configuration. A user must then authenticate not only to a primary authentication server but also to a

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
			secondary authentication server. User groups are aggregated across both authentication servers. The user name must be exactly the same on both authentication servers, but the authentication servers can require different passwords.
groupextraction	<Boolean>	Read-write	Bind the Authentication policy to a tertiary chain which will be used only for group extraction. The user will not authenticate against this server, and this will only be called it primary and/or secondary authentication has succeeded.
__count	<Double>	Read-write	count parameter

Operations

(click to see Properties)

ADD:| DELETE:| GET| GET (ALL)| COUNT

Some options that you can use for each operations:

- **Getting warnings in response:**NITRO allows you to get warnings in an operation by specifying the "warning" query parameter as "yes". For example, to get warnings while connecting to the Citrix ADC appliance, the URL is as follows:

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/login?warning=yes

If any, the warnings are displayed in the response payload with the HTTP code "209 X-NITRO-WARNING".

- **Authenticated access for individual NITRO operations:**NITRO allows you to logon to the Citrix ADC appliance to perform individual operations. You can use this option instead of creating a NITRO session (using the login object) and then using that session to perform all operations,

  To do this, you must specify the username and password in the request header of the NITRO request as follows:

  X-NITRO-USER:<username>

  X-NITRO-PASS:<password>

  Note: In such cases, make sure that the request header DOES not include the following:

  Cookie:NITRO_AUTH_TOKEN=<tokenvalue>

`*Note:`*

Mandatory parameters are marked in redand placeholder content is marked in <green>.

add

URL:http://<Citrix-ADC-IP-address(NSIP)/nitro/v1/config/vpnglobal_authenticationtacacspolicy_binding

HTTP Method:PUT

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Content-Type:application/json

Request Payload:

```
{

"vpnglobal_authenticationtacacspolicy_binding":{

"policyname":<String_value>,
```

```
"priority":<Double_value>,

"secondary":<Boolean_value>,

"groupextraction":<Boolean_value>,

"gotopriorityexpression":<String_value>

}}
```

Response:

HTTP Status Code on Success: 201 Created HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error

delete

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/vpnglobal_authenticationtacacspolicy_binding

Query-parameters:

args

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/vpnglobal_authenticationtacacspolicy_binding?**args=policyname:<String_value>,secondary:<Boolean_value>,groupextraction:<Boolean_value>**

HTTP Method:DELETE

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue>

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error

get

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/vpnglobal_authenticationtacacspolicy_binding

Query-parameters:

filter

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/vpnglobal_authenticationtacacspolicy_binding?**filter=property-name1:property-value1,property-name2:property-value2**

Use this query-parameter to get the filtered set of vpnglobal_authenticationtacacspolicy_binding resources configured on Citrix ADC. Filtering can be done on any of the properties of the resource.

pagination

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/vpnglobal_authenticationtacacspolicy_binding?**pagesize=#no;pageno=#no**

Use this query-parameter to get the vpnglobal_authenticationtacacspolicy_binding resources in chunks.

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "vpnglobal_authenticationtacacspolicy_binding": [ {

"priority":<Double_value>,
```

```
"policyname":<String_value>,

"gotopriorityexpression":<String_value>,

"secondary":<Boolean_value>,

"groupextraction":<Boolean_value>

}]}
```

get (all)

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/vpnglobal_authenticationtacacspolicy_binding

Query-parameters:

bulkbindings

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/vpnglobal_authenticationtacacspolicy_binding?**bulkbindings=yes**

NITRO allows you to fetch bindings in bulk.

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "vpnglobal_authenticationtacacspolicy_binding": [ {

"priority":<Double_value>,

"policyname":<String_value>,

"gotopriorityexpression":<String_value>,

"secondary":<Boolean_value>,

"groupextraction":<Boolean_value>

}]}
```

count

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/vpnglobal_authenticationtacacspolicy_binding?**count=yes**

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "vpnglobal_authenticationtacacspolicy_binding": [ { "__count":
"#no"} ] }
```

vpnglobal_binding

Binding object which returns the resources bound to vpnglobal.

Properties

(click to see Operations)

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
vpnglobal_intranetip6_binding	<vpnglobal_intranetip6_binding[]>	Read-only	intranetip6 that can be bound to vpnglobal.
vpnglobal_auditnslogpolicy_binding	<vpnglobal_auditnslogpolicy_binding[]>	Read-only	auditnslogpolicy that can be bound to vpnglobal.
vpnglobal_authenticationpolicy_binding	<vpnglobal_authenticationpolicy_binding[]>	Read-only	authenticationpolicy that can be bound to vpnglobal.
vpnglobal_authenticationradiuspolicy_binding	<vpnglobal_authenticationradiuspolicy_binding[]>	Read-only	authenticationradiuspolicy that can be bound to vpnglobal.
vpnglobal_vpnsessionpolicy_binding	<vpnglobal_vpnsessionpolicy_binding[]>	Read-only	vpnsessionpolicy that can be bound to vpnglobal.
vpnglobal_domain_binding	<vpnglobal_domain_binding[]>	Read-only	domain that can be bound to vpnglobal.
vpnglobal_vpneula_binding	<vpnglobal_vpneula_binding[]>	Read-only	vpneula that can be bound to vpnglobal.
vpnglobal_vpnclientlessaccesspolicy_binding	<vpnglobal_vpnclientlessaccesspolicy_binding[]>	Read-only	vpnclientlessaccesspolicy that can be bound to vpnglobal.

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
vpnglobal_vpnurlpolicy_binding	<vpnglobal_vpnurlpolicy_binding[]>	Read-only	vpnurlpolicy that can be bound to vpnglobal.
vpnglobal_intranetip_binding	<vpnglobal_intranetip_binding[]>	Read-only	intranetip that can be bound to vpnglobal.
vpnglobal_vpnnexthopserver_binding	<vpnglobal_vpnnexthopserver_binding[]>	Read-only	vpnnexthopserver that can be bound to vpnglobal.
vpnglobal_authenticationldappolicy_binding	<vpnglobal_authenticationldappolicy_binding[]>	Read-only	authenticationldappolicy that can be bound to vpnglobal.
vpnglobal_sharefileserver_binding	<vpnglobal_sharefileserver_binding[]>	Read-only	sharefileserver that can be bound to vpnglobal.
vpnglobal_vpntrafficpolicy_binding	<vpnglobal_vpntrafficpolicy_binding[]>	Read-only	vpntrafficpolicy that can be bound to vpnglobal.
vpnglobal_authenticationlocalpolicy_binding	<vpnglobal_authenticationlocalpolicy_binding[]>	Read-only	authenticationlocalpolicy that can be bound to vpnglobal.
vpnglobal_vpnintranetapplication_binding	<vpnglobal_vpnintranetapplication_binding[]>	Read-only	vpnintranetapplication that can be bound to vpnglobal.
vpnglobal_appcontroller_binding	<vpnglobal_appcontroller_binding[]>	Read-only	appcontroller that can be bound to vpnglobal.
vpnglobal_authenticationnegotiatepolicy_binding	<vpnglobal_authenticationnegotiatepolicy_binding[]>	Read-only	authenticationnegotiatepolicy that can be bound to vpnglobal.

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
vpnglobal_authenticationtacacspolicy_binding	<vpnglobal_authenticationtacacspolicy_binding[]>	Read-only	authenticationtacacspolicy that can be bound to vpnglobal.
vpnglobal_vpnportaltheme_binding	<vpnglobal_vpnportaltheme_binding[]>	Read-only	vpnportaltheme that can be bound to vpnglobal.
vpnglobal_authenticationsamlpolicy_binding	<vpnglobal_authenticationsamlpolicy_binding[]>	Read-only	authenticationsamlpolicy that can be bound to vpnglobal.
vpnglobal_staserver_binding	<vpnglobal_staserver_binding[]>	Read-only	staserver that can be bound to vpnglobal.
vpnglobal_auditsyslogpolicy_binding	<vpnglobal_auditsyslogpolicy_binding[]>	Read-only	auditsyslogpolicy that can be bound to vpnglobal.
vpnglobal_authenticationcertpolicy_binding	<vpnglobal_authenticationcertpolicy_binding[]>	Read-only	authenticationcertpolicy that can be bound to vpnglobal.
vpnglobal_vpnurl_binding	<vpnglobal_vpnurl_binding[]>	Read-only	vpnurl that can be bound to vpnglobal.
vpnglobal_sslcertkey_binding	<vpnglobal_sslcertkey_binding[]>	Read-only	sslcertkey that can be bound to vpnglobal.

Operations

(click to see Properties)

GET| GET (ALL)

Some options that you can use for each operations:

- **Getting warnings in response:**NITRO allows you to get warnings in an operation by specifying the "warning" query parameter as "yes". For example, to get warnings while connecting to the Citrix ADC appliance, the URL is as follows:

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/login?warning=yes

If any, the warnings are displayed in the response payload with the HTTP code "209 X-NITRO-WARNING".

- **Authenticated access for individual NITRO operations:**NITRO allows you to logon to the Citrix ADC appliance to perform individual operations. You can use this option instead of creating a NITRO session (using the login object) and then using that session to perform all operations,

  To do this, you must specify the username and password in the request header of the NITRO request as follows:

  X-NITRO-USER:<username>

  X-NITRO-PASS:<password>

  Note: In such cases, make sure that the request header DOES not include the following:

  Cookie:NITRO_AUTH_TOKEN=<tokenvalue>

`*Note:`*

Mandatory parameters are marked in redand placeholder content is marked in <green>.

get

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/vpnglobal_binding

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "vpnglobal_binding": [ {

"vpnglobal_vpnintranetapplication_binding":<vpnglobal_vpnintranetapplicati
on_binding[]_value>,

"vpnglobal_sslcertkey_binding":<vpnglobal_sslcertkey_binding[]_value>,

"vpnglobal_authenticationsamlpolicy_binding":<vpnglobal_authenticationsaml
policy_binding[]_value>,

"vpnglobal_intranetip_binding":<vpnglobal_intranetip_binding[]_value>,

"vpnglobal_authenticationradiuspolicy_binding":<vpnglobal_authenticationra
diuspolicy_binding[]_value>,

"vpnglobal_staserver_binding":<vpnglobal_staserver_binding[]_value>,

"vpnglobal_sharefileserver_binding":<vpnglobal_sharefileserver_binding[]_v
alue>,

"vpnglobal_intranetip6_binding":<vpnglobal_intranetip6_binding[]_value>,

"vpnglobal_appcontroller_binding":<vpnglobal_appcontroller_binding[]_value
>,

"vpnglobal_authenticationtacacspolicy_binding":<vpnglobal_authenticationta
cacspolicy_binding[]_value>,

"vpnglobal_authenticationpolicy_binding":<vpnglobal_authenticationpolicy_b
inding[]_value>,

"vpnglobal_vpntrafficpolicy_binding":<vpnglobal_vpntrafficpolicy_binding[]
_value>,

"vpnglobal_domain_binding":<vpnglobal_domain_binding[]_value>,

"vpnglobal_vpnnexthopserver_binding":<vpnglobal_vpnnexthopserver_binding[]
_value>,

"vpnglobal_authenticationldappolicy_binding":<vpnglobal_authenticationldap
policy_binding[]_value>,
```

```
"vpnglobal_vpnsessionpolicy_binding":<vpnglobal_vpnsessionpolicy_binding[]
_value>,

"vpnglobal_authenticationcertpolicy_binding":<vpnglobal_authenticationcert
policy_binding[]_value>,

"vpnglobal_authenticationnegotiatepolicy_binding":<vpnglobal_authenticatio
nnegotiatepolicy_binding[]_value>,

"vpnglobal_auditsyslogpolicy_binding":<vpnglobal_auditsyslogpolicy_binding
[]_value>,

"vpnglobal_vpneula_binding":<vpnglobal_vpneula_binding[]_value>,

"vpnglobal_vpnurl_binding":<vpnglobal_vpnurl_binding[]_value>,

"vpnglobal_vpnportaltheme_binding":<vpnglobal_vpnportaltheme_binding[]_val
ue>,

"vpnglobal_vpnclientlessaccesspolicy_binding":<vpnglobal_vpnclientlessacce
sspolicy_binding[]_value>,

"vpnglobal_authenticationlocalpolicy_binding":<vpnglobal_authenticationloc
alpolicy_binding[]_value>,

"vpnglobal_auditnslogpolicy_binding":<vpnglobal_auditnslogpolicy_binding[]
_value>,

"vpnglobal_vpnurlpolicy_binding":<vpnglobal_vpnurlpolicy_binding[]_value>

}]}
```

get (all)

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/vpnglobal_binding

Query-parameters:

bulkbindings

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/vpnglobal_binding?**bulkbindings=yes**

NITRO allows you to fetch bindings in bulk.

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "vpnglobal_binding": [ {

"vpnglobal_vpnintranetapplication_binding":<vpnglobal_vpnintranetapplicati
on_binding[]_value>,

"vpnglobal_sslcertkey_binding":<vpnglobal_sslcertkey_binding[]_value>,

"vpnglobal_authenticationsamlpolicy_binding":<vpnglobal_authenticationsaml
policy_binding[]_value>,

"vpnglobal_intranetip_binding":<vpnglobal_intranetip_binding[]_value>,

"vpnglobal_authenticationradiuspolicy_binding":<vpnglobal_authenticationra
diuspolicy_binding[]_value>,

"vpnglobal_staserver_binding":<vpnglobal_staserver_binding[]_value>,

"vpnglobal_sharefileserver_binding":<vpnglobal_sharefileserver_binding[]_v
alue>,

"vpnglobal_intranetip6_binding":<vpnglobal_intranetip6_binding[]_value>,

"vpnglobal_appcontroller_binding":<vpnglobal_appcontroller_binding[]_value
>,
```

```
"vpnglobal_authenticationtacacspolicy_binding":<vpnglobal_authenticationta
cacspolicy_binding[]_value>,

"vpnglobal_authenticationpolicy_binding":<vpnglobal_authenticationpolicy_b
inding[]_value>,

"vpnglobal_vpntrafficpolicy_binding":<vpnglobal_vpntrafficpolicy_binding[]
_value>,

"vpnglobal_domain_binding":<vpnglobal_domain_binding[]_value>,

"vpnglobal_vpnnexthopserver_binding":<vpnglobal_vpnnexthopserver_binding[]
_value>,

"vpnglobal_authenticationldappolicy_binding":<vpnglobal_authenticationldap
policy_binding[]_value>,

"vpnglobal_vpnsessionpolicy_binding":<vpnglobal_vpnsessionpolicy_binding[]
_value>,

"vpnglobal_authenticationcertpolicy_binding":<vpnglobal_authenticationcert
policy_binding[]_value>,

"vpnglobal_authenticationnegotiatepolicy_binding":<vpnglobal_authenticatio
nnegotiatepolicy_binding[]_value>,

"vpnglobal_auditsyslogpolicy_binding":<vpnglobal_auditsyslogpolicy_binding
[]_value>,

"vpnglobal_vpneula_binding":<vpnglobal_vpneula_binding[]_value>,

"vpnglobal_vpnurl_binding":<vpnglobal_vpnurl_binding[]_value>,

"vpnglobal_vpnportaltheme_binding":<vpnglobal_vpnportaltheme_binding[]_val
ue>,

"vpnglobal_vpnclientlessaccesspolicy_binding":<vpnglobal_vpnclientlessacce
sspolicy_binding[]_value>,

"vpnglobal_authenticationlocalpolicy_binding":<vpnglobal_authenticationloc
alpolicy_binding[]_value>,

"vpnglobal_auditnslogpolicy_binding":<vpnglobal_auditnslogpolicy_binding[]
```

```
_value>,

"vpnglobal_vpnurlpolicy_binding":<vpnglobal_vpnurlpolicy_binding[]_value>

}]}
```

vpnglobal_domain_binding

Binding object showing the domain that can be bound to vpnglobal.

Properties

(click to see Operations)

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
intranetdomain	<String>	Read-write	The conflicting intranet domain name.
gotopriorityexpression	<String>	Read-write	Applicable only to advance vpn session policy. An expression or other value specifying the priority of the next policy which will get evaluated if the current policy rule evaluates to TRUE.
__count	<Double>	Read-write	count parameter

Operations

(click to see Properties)

ADD:| DELETE:| GET| GET (ALL)| COUNT

Some options that you can use for each operations:

- **Getting warnings in response:**NITRO allows you to get warnings in an operation by specifying the "warning" query parameter as "yes". For example, to get warnings while connecting to the Citrix ADC appliance, the URL is as follows:

  http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/login?warning=yes

  If any, the warnings are displayed in the response payload with the HTTP code "209 X-NITRO-WARNING".

- **Authenticated access for individual NITRO operations:**NITRO allows you to logon to the Citrix ADC appliance to perform individual operations. You can use this option instead of creating a NITRO session (using the login object) and then using that session to perform all operations,

  To do this, you must specify the username and password in the request header of the NITRO request as follows:

  X-NITRO-USER:<username>

  X-NITRO-PASS:<password>

  Note: In such cases, make sure that the request header DOES not include the following:

  Cookie:NITRO_AUTH_TOKEN=<tokenvalue>

`*Note:`*

Mandatory parameters are marked in redand placeholder content is marked in <green>.

add

URL:http://<Citrix-ADC-IP-address(NSIP)/nitro/v1/config/vpnglobal_domain_binding

HTTP Method:PUT

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Content-Type:application/json

Request Payload:

```
{
```

```
"vpnglobal_domain_binding":{

"intranetdomain":<String_value>,

"gotopriorityexpression":<String_value>

}}
```

Response:

HTTP Status Code on Success: 201 Created HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error

delete

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/vpnglobal_domain_binding

Query-parameters:

args

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/vpnglobal_domain_binding?**args=intranetdomain:<String_value>**

HTTP Method:DELETE

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue>

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error

get

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/vpnglobal_domain_binding

Query-parameters:

filter

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/vpnglobal_domain_binding?**filter=property-name1:property-value1,property-name2:property-value2**

Use this query-parameter to get the filtered set of vpnglobal_domain_binding resources configured on Citrix ADC. Filtering can be done on any of the properties of the resource.

pagination

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/vpnglobal_domain_binding?**pagesize=#no;pageno=#no**

Use this query-parameter to get the vpnglobal_domain_binding resources in chunks.

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "vpnglobal_domain_binding": [ {

"intranetdomain":<String_value>,

"gotopriorityexpression":<String_value>

}]}
```

get (all)

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/vpnglobal_domain_binding

Query-parameters:

bulkbindings

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/vpnglobal_domain_binding?**bulkbindings=yes**

NITRO allows you to fetch bindings in bulk.

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "vpnglobal_domain_binding": [ {

"intranetdomain":<String_value>,

"gotopriorityexpression":<String_value>

}]}
```

count

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/vpnglobal_domain_binding?**count=yes**

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "vpnglobal_domain_binding": [ { "__count": "#no"} ] }
```

vpnglobal_intranetip6_binding

Binding object showing the intranetip6 that can be bound to vpnglobal.

Properties

(click to see Operations)

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
intranetip6	<String>	Read-write	The intranet ip address or range.
gotopriorityexpression	<String>	Read-write	Applicable only to advance vpn session policy. An expression or other value specifying the priority of the next policy which will get evaluated if the current policy rule evaluates to TRUE.
numaddr	<Double>	Read-write	The intranet ip address or range's netmask.

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
__count	<Double>	Read-write	count parameter

Operations

(click to see Properties)

ADD:| DELETE:| GET| GET (ALL)| COUNT

Some options that you can use for each operations:

- **Getting warnings in response:**NITRO allows you to get warnings in an operation by specifying the "warning" query parameter as "yes". For example, to get warnings while connecting to the Citrix ADC appliance, the URL is as follows:

  http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/login?warning=yes

  If any, the warnings are displayed in the response payload with the HTTP code "209 X-NITRO-WARNING".
- **Authenticated access for individual NITRO operations:**NITRO allows you to logon to the Citrix ADC appliance to perform individual operations. You can use this option instead of creating a NITRO session (using the login object) and then using that session to perform all operations,

  To do this, you must specify the username and password in the request header of the NITRO request as follows:

  X-NITRO-USER:<username>

  X-NITRO-PASS:<password>

  Note: In such cases, make sure that the request header DOES not include the following:

  Cookie:NITRO_AUTH_TOKEN=<tokenvalue>

`*Note:`*

Mandatory parameters are marked in redand placeholder content is marked in <green>.

add

URL:http://<Citrix-ADC-IP-address(NSIP)/nitro/v1/config/vpnglobal_intranetip6_binding

HTTP Method:PUT

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Content-Type:application/json

Request Payload:

```
{

"vpnglobal_intranetip6_binding":{

"intranetip6":<String_value>,

"numaddr":<Double_value>,

"gotopriorityexpression":<String_value>

}}
```

Response:

HTTP Status Code on Success: 201 Created HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error

delete

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/vpnglobal_intranetip6_binding

Query-parameters:

args

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/vpnglobal_intranetip6_binding?**args=intranetip6:<String_value>,numaddr:<Double_value>**

HTTP Method:DELETE

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue>

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error

get

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/vpnglobal_intranetip6_binding

Query-parameters:

filter

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/vpnglobal_intranetip6_binding?**filter=property-name1:property-value1,property-name2:property-value2**

Use this query-parameter to get the filtered set of vpnglobal_intranetip6_binding resources configured on Citrix ADC. Filtering can be done on any of the properties of the resource.

pagination

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/vpnglobal_intranetip6_binding?**pagesize=#no;pageno=#no**

Use this query-parameter to get the vpnglobal_intranetip6_binding resources in chunks.

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "vpnglobal_intranetip6_binding": [ {
```

```
"intranetip6":<String_value>,

"gotopriorityexpression":<String_value>,

"numaddr":<Double_value>

}]}
```

get (all)

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/vpnglobal_intranetip6_binding

Query-parameters:

bulkbindings

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/vpnglobal_intranetip6_binding?**bulkbindings=yes**

NITRO allows you to fetch bindings in bulk.

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "vpnglobal_intranetip6_binding": [ {

"intranetip6":<String_value>,
```

```
"gotopriorityexpression":<String_value>,

"numaddr":<Double_value>

}]}
```

count

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/vpnglobal_intranetip6_binding?**count=yes**

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "vpnglobal_intranetip6_binding": [ { "__count": "#no"} ] }
```

vpnglobal_intranetip_binding

Binding object showing the intranetip that can be bound to vpnglobal.

Properties

(click to see Operations)

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
intranetip	<String>	Read-write	The intranet ip address or range.
gotopriorityexpression	<String>	Read-write	Applicable only to advance vpn session policy. An expression or other value specifying the priority of the next policy which will get evaluated if the current policy rule evaluates to TRUE.
netmask	<String>	Read-write	The intranet ip address or range's netmask.
__count	<Double>	Read-write	count parameter

Operations

(click to see Properties)

ADD:| DELETE:| GET| GET (ALL)| COUNT

Some options that you can use for each operations:

- **Getting warnings in response:**NITRO allows you to get warnings in an operation by specifying the "warning" query parameter as "yes". For example, to get warnings while connecting to the Citrix ADC appliance, the URL is as follows:

  http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/login?warning=yes

  If any, the warnings are displayed in the response payload with the HTTP code "209 X-NITRO-WARNING".

- **Authenticated access for individual NITRO operations:**NITRO allows you to logon to the Citrix ADC appliance to perform individual operations. You can use this option instead of creating a NITRO session (using the login object) and then using that session to perform all operations,

To do this, you must specify the username and password in the request header of the NITRO request as follows:

X-NITRO-USER:<username>

X-NITRO-PASS:<password>

Note: In such cases, make sure that the request header DOES not include the following:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue>

`*Note:`*

Mandatory parameters are marked in redand placeholder content is marked in <green>.

add

URL:http://<Citrix-ADC-IP-address(NSIP)/nitro/v1/config/vpnglobal_intranetip_binding

HTTP Method:PUT

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Content-Type:application/json

Request Payload:

```
{

"vpnglobal_intranetip_binding":{

"intranetip":<String_value>,

"netmask":<String_value>,

"gotopriorityexpression":<String_value>

}}
```

Response:

HTTP Status Code on Success: 201 Created HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error

delete

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/vpnglobal_intranetip_binding

Query-parameters:

args

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/vpnglobal_intranetip_binding?**args=intranetip:<String_value>,netmask:<String_value>**

HTTP Method:DELETE

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue>

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error

get

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/vpnglobal_intranetip_binding

Query-parameters:

filter

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/vpnglobal_intranetip_binding?**filter=property-name1:property-value1,property-name2:property-value2**

Use this query-parameter to get the filtered set of vpnglobal_intranetip_binding resources configured on Citrix ADC. Filtering can be done on any of the properties of the resource.

pagination

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/vpnglobal_intranetip_binding?**pagesize=#no;pageno=#no**

Use this query-parameter to get the vpnglobal_intranetip_binding resources in chunks.

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "vpnglobal_intranetip_binding": [ {

"intranetip":<String_value>,

"gotopriorityexpression":<String_value>,

"netmask":<String_value>

}]}
```

get (all)

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/vpnglobal_intranetip_binding

Query-parameters:

bulkbindings

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/vpnglobal_intranetip_binding?**bulkbindings=yes**

NITRO allows you to fetch bindings in bulk.

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "vpnglobal_intranetip_binding": [ {

"intranetip":<String_value>,

"gotopriorityexpression":<String_value>,

"netmask":<String_value>

}]}
```

count

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/vpnglobal_intranetip_binding?**count=yes**

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "vpnglobal_intranetip_binding": [ { "__count": "#no"} ] }
```

vpnglobal_sharefileserver_binding

Binding object showing the sharefileserver that can be bound to vpnglobal.

Properties

(click to see Operations)

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
gotopriorityexpression	<String>	Read-write	Applicable only to advance vpn session policy. An expression or other value specifying the priority of the next policy which will get evaluated if the current policy rule evaluates to TRUE.
sharefile	<String>	Read-write	Configured Sharefile server, in the format IP:PORT / FQDN:PORT.
__count	<Double>	Read-write	count parameter

Operations

(click to see Properties)

ADD:| DELETE:| GET| GET (ALL)| COUNT

Some options that you can use for each operations:

- **Getting warnings in response:**NITRO allows you to get warnings in an operation by specifying the "warning" query parameter as "yes". For example, to get warnings while connecting to the Citrix ADC appliance, the URL is as follows:

  http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/login?warning=yes

  If any, the warnings are displayed in the response payload with the HTTP code "209 X-NITRO-WARNING".

- **Authenticated access for individual NITRO operations:**NITRO allows you to logon to the Citrix ADC appliance to perform individual operations. You can use this option instead of creating a NITRO session (using the login object) and then using that session to perform all operations,

  To do this, you must specify the username and password in the request header of the NITRO request as follows:

  X-NITRO-USER:<username>

  X-NITRO-PASS:<password>

  Note: In such cases, make sure that the request header DOES not include the following:

  Cookie:NITRO_AUTH_TOKEN=<tokenvalue>

`*Note:`*

Mandatory parameters are marked in redand placeholder content is marked in <green>.

add

URL:http://<Citrix-ADC-IP-address(NSIP)/nitro/v1/config/vpnglobal_sharefileserver_binding

HTTP Method:PUT

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Content-Type:application/json

Request Payload:

```
{
```

```
"vpnglobal_sharefileserver_binding":{

"sharefile":<String_value>,

"gotopriorityexpression":<String_value>

}}
```

Response:

HTTP Status Code on Success: 201 Created HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error

delete

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/vpnglobal_sharefileserver_binding

Query-parameters:

args

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/vpnglobal_sharefileserver_binding?**args=sharefile:<String_value>**

HTTP Method:DELETE

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue>

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error

get

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/vpnglobal_sharefileserver_binding

Query-parameters:

filter

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/vpnglobal_sharefileserver_binding?**filter=property-name1:property-value1,property-name2:property-value2**

Use this query-parameter to get the filtered set of vpnglobal_sharefileserver_binding resources configured on Citrix ADC. Filtering can be done on any of the properties of the resource.

pagination

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/vpnglobal_sharefileserver_binding?**pagesize=#no;pageno=#no**

Use this query-parameter to get the vpnglobal_sharefileserver_binding resources in chunks.

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "vpnglobal_sharefileserver_binding": [ {

"gotopriorityexpression":<String_value>,

"sharefile":<String_value>

}]}
```

get (all)

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/vpnglobal_sharefileserver_binding

Query-parameters:

bulkbindings

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/vpnglobal_sharefileserver_binding?**bulkbindings=yes**

NITRO allows you to fetch bindings in bulk.

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "vpnglobal_sharefileserver_binding": [ {

"gotopriorityexpression":<String_value>,

"sharefile":<String_value>

}]}
```

count

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/vpnglobal_sharefileserver_binding?**count=yes**

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "vpnglobal_sharefileserver_binding": [ { "__count": "#no"} ] }
```

vpnglobal_sslcertkey_binding

Binding object showing the sslcertkey that can be bound to vpnglobal.

Properties

(click to see Operations)

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
crlcheck	<String>	Read-write	The state of the CRL check parameter (Mandatory/ Optional). Possible values = Mandatory, Optional
cacert	<String>	Read-write	The name of the CA certificate binding.
certkeyname	<String>	Read-write	SSL certkey to use in signing tokens.
gotopriorityexpression	<String>	Read-write	Applicable only to advance vpn session policy. An expression

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
			or other value specifying the priority of the next policy which will get evaluated if the current policy rule evaluates to TRUE.
ocspcheck	<String>	Read-write	The state of the OCSP check parameter (Mandatory/ Optional). Possible values = Mandatory, Optional
userdataencryptionkey	<String>	Read-write	Certificate to be used for encrypting user data like KB Question and Answers, Alternate Email Address, etc.
__count	<Double>	Read-write	count parameter

Operations

(click to see Properties)

ADD:| DELETE:| GET| GET (ALL)| COUNT

Some options that you can use for each operations:

- **Getting warnings in response:**NITRO allows you to get warnings in an operation by specifying the "warning" query parameter as "yes". For example, to get warnings while connecting to the Citrix ADC appliance, the URL is as follows:

  http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/login?warning=yes

  If any, the warnings are displayed in the response payload with the HTTP code "209 X-NITRO-WARNING".

- **Authenticated access for individual NITRO operations:**NITRO allows you to logon to the Citrix ADC appliance to perform individual operations. You can use this option instead of creating a NITRO session (using the login object) and then using that session to perform all operations,

  To do this, you must specify the username and password in the request header of the NITRO request as follows:

  X-NITRO-USER:<username>

  X-NITRO-PASS:<password>

  Note: In such cases, make sure that the request header DOES not include the following:

  Cookie:NITRO_AUTH_TOKEN=<tokenvalue>

`*Note:`*

Mandatory parameters are marked in redand placeholder content is marked in <green>.

add

URL:http://<Citrix-ADC-IP-address(NSIP)/nitro/v1/config/vpnglobal_sslcertkey_binding

HTTP Method:PUT

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Content-Type:application/json

Request Payload:

```
{

"vpnglobal_sslcertkey_binding":{

"gotopriorityexpression":<String_value>,

"certkeyname":<String_value>,

"userdataencryptionkey":<String_value>,

"cacert":<String_value>,
```

```
"crlcheck":<String_value>,

"ocspcheck":<String_value>

}}
```

Response:

HTTP Status Code on Success: 201 Created HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error

delete

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/vpnglobal_sslcertkey_binding

Query-parameters:

args

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/vpnglobal_sslcertkey_binding?**args=certkeyname:<String_value>,userdataencryptionkey:<String_value>,cacert:<String_value>**

HTTP Method:DELETE

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue>

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error

get

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/vpnglobal_sslcertkey_binding

Query-parameters:

filter

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/vpnglobal_sslcertkey_binding?**filter=property-name1:property-value1,property-name2:property-value2**

Use this query-parameter to get the filtered set of vpnglobal_sslcertkey_binding resources configured on Citrix ADC. Filtering can be done on any of the properties of the resource.

pagination

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/vpnglobal_sslcertkey_binding?**pagesize=#no;pageno=#no**

Use this query-parameter to get the vpnglobal_sslcertkey_binding resources in chunks.

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "vpnglobal_sslcertkey_binding": [ {

"crlcheck":<String_value>,

"cacert":<String_value>,

"certkeyname":<String_value>,

"gotopriorityexpression":<String_value>,

"ocspcheck":<String_value>,

"userdataencryptionkey":<String_value>
```

```
}]}
```

get (all)

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/vpnglobal_sslcertkey_binding

Query-parameters:

bulkbindings

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/vpnglobal_sslcertkey_binding?**bulkbindings=yes**

NITRO allows you to fetch bindings in bulk.

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "vpnglobal_sslcertkey_binding": [ {

"crlcheck":<String_value>,

"cacert":<String_value>,

"certkeyname":<String_value>,

"gotopriorityexpression":<String_value>,
```

```
"ocspcheck":<String_value>,

"userdataencryptionkey":<String_value>

}]}
```

count

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/vpnglobal_sslcertkey_binding?**count=yes**

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "vpnglobal_sslcertkey_binding": [ { "__count": "#no"} ] }
```

vpnglobal_staserver_binding

Binding object showing the staserver that can be bound to vpnglobal.

Properties

(click to see Operations)

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
staserver	<String>	Read-write	Configured Secure Ticketing Authority (STA) server.
gotopriorityexpression	<String>	Read-write	Applicable only to advance vpn session policy. An expression or other value specifying the priority of the next policy which will get evaluated if the current policy rule evaluates to TRUE.
staaddresstype	<String>	Read-write	Type of the STA server address(ipv4/v6). Possible values = IPV4, IPV6
stastate	<String>	Read-only	State of the STA Server. If Authority ID is set then STA Server is UP else DOWN. Possible values = UP, DOWN
staauthid	<String>	Read-only	Authority ID of the STA Server. Authority ID is used to match incoming STA Tickets in the SOCKS/CGP protocol with the right STA Server.
__count	<Double>	Read-only	count parameter

Operations

(click to see Properties)

ADD:| DELETE:| GET| GET (ALL)| COUNT

Some options that you can use for each operations:

- **Getting warnings in response:**NITRO allows you to get warnings in an operation by specifying the "warning" query parameter as "yes". For example, to get warnings while connecting to the Citrix ADC appliance, the URL is as follows:

  http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/login?warning=yes

  If any, the warnings are displayed in the response payload with the HTTP code "209 X-NITRO-WARNING".

- **Authenticated access for individual NITRO operations:**NITRO allows you to logon to the Citrix ADC appliance to perform individual operations. You can use this option instead of creating a NITRO session (using the login object) and then using that session to perform all operations,

  To do this, you must specify the username and password in the request header of the NITRO request as follows:

  X-NITRO-USER:<username>

  X-NITRO-PASS:<password>

  Note: In such cases, make sure that the request header DOES not include the following:

  Cookie:NITRO_AUTH_TOKEN=<tokenvalue>

`*Note:`*

Mandatory parameters are marked in redand placeholder content is marked in <green>.

add

URL:http://<Citrix-ADC-IP-address(NSIP)/nitro/v1/config/vpnglobal_staserver_binding

HTTP Method:PUT

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Content-Type:application/json

Request Payload:

```
{

"vpnglobal_staserver_binding":{

"staserver":<String_value>,

"staaddresstype":<String_value>,

"gotopriorityexpression":<String_value>

}}
```

Response:

HTTP Status Code on Success: 201 Created HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error

delete

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/vpnglobal_staserver_binding

Query-parameters:

args

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/vpnglobal_staserver_binding?**args=staserver:<String_value>**

HTTP Method:DELETE

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue>

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error

get

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/vpnglobal_staserver_binding

Query-parameters:

filter

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/vpnglobal_staserver_binding?**filter=property-name1:property-value1,property-name2:property-value2**

Use this query-parameter to get the filtered set of vpnglobal_staserver_binding resources configured on Citrix ADC. Filtering can be done on any of the properties of the resource.

pagination

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/vpnglobal_staserver_binding?**pagesize=#no;pageno=#no**

Use this query-parameter to get the vpnglobal_staserver_binding resources in chunks.

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "vpnglobal_staserver_binding": [ {
```

```
"staserver":<String_value>,

"gotopriorityexpression":<String_value>,

"staaddresstype":<String_value>,

"stastate":<String_value>,

"staauthid":<String_value>

}]}
```

get (all)

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/vpnglobal_staserver_binding

Query-parameters:

bulkbindings

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/vpnglobal_staserver_binding?**bulkbindings=yes**

NITRO allows you to fetch bindings in bulk.

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "vpnglobal_staserver_binding": [ {

"staserver":<String_value>,

"gotopriorityexpression":<String_value>,

"staaddresstype":<String_value>,

"stastate":<String_value>,

"staauthid":<String_value>

}]}
```

count

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/vpnglobal_staserver_binding?**count=yes**

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "vpnglobal_staserver_binding": [ { "__count": "#no"} ] }
```

vpnglobal_vpnclientlessaccesspolicy_binding

Binding object showing the vpnclientlessaccesspolicy that can be bound to vpnglobal.

Properties

(click to see Operations)

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
priority	<Double>	Read-write	Integer specifying the policy's priority. The lower the priority number, the higher the policy's priority. Maximum value for default syntax policies is 2147483647 and for classic policies is 64000. Minimum value = 0 Maximum value = 2147483647
globalbindtype	<String>	Read-write	. Possible values = SYSTEM_GLOBAL, VPN_GLOBAL, RNAT_GLOBAL
policyname	<String>	Read-write	The name of the policy.
gotopriorityexpression	<String>	Read-write	Applicable only to advance vpn session policy. An expression or other value specifying the priority of the next policy which will get evaluated if the

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
			current policy rule evaluates to TRUE.
secondary	<Boolean>	Read-write	Bind the authentication policy as the secondary policy to use in a two-factor configuration. A user must then authenticate not only to a primary authentication server but also to a secondary authentication server. User groups are aggregated across both authentication servers. The user name must be exactly the same on both authentication servers, but the authentication servers can require different passwords.
type	<String>	Read-write	Bindpoint to which the policy is bound. Possible values = REQ_OVERRIDE, REQ_DEFAULT, RES_OVERRIDE, RES_DEFAULT
groupextraction	<Boolean>	Read-write	Bind the Authentication policy to a tertiary chain which will be used

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
			only for group extraction. The user will not authenticate against this server, and this will only be called it primary and/or secondary authentication has succeeded.
__count	<Double>	Read-write	count parameter

Operations

(click to see Properties)

ADD:| DELETE:| GET| GET (ALL)| COUNT

Some options that you can use for each operations:

- **Getting warnings in response:**NITRO allows you to get warnings in an operation by specifying the "warning" query parameter as "yes". For example, to get warnings while connecting to the Citrix ADC appliance, the URL is as follows:

  http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/login?warning=yes

  If any, the warnings are displayed in the response payload with the HTTP code "209 X-NITRO-WARNING".

- **Authenticated access for individual NITRO operations:**NITRO allows you to logon to the Citrix ADC appliance to perform individual operations. You can use this option instead of creating a NITRO session (using the login object) and then using that session to perform all operations,

  To do this, you must specify the username and password in the request header of the NITRO request as follows:

  X-NITRO-USER:<username>

  X-NITRO-PASS:<password>

  Note: In such cases, make sure that the request header DOES not include the following:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue>

`*Note:`*

Mandatory parameters are marked in redand placeholder content is marked in <green>.

add

URL:http://<Citrix-ADC-IP-address(NSIP)/nitro/v1/config/vpnglobal_vpnclientlessaccesspolicy_binding

HTTP Method:PUT

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Content-Type:application/json

Request Payload:

```
{
"vpnglobal_vpnclientlessaccesspolicy_binding":{
"policyname":<String_value>,
"priority":<Double_value>,
"secondary":<Boolean_value>,
"groupextraction":<Boolean_value>,
"gotopriorityexpression":<String_value>
}}
```

Response:

HTTP Status Code on Success: 201 Created HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error

delete

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/vpnglobal_vpnclientlessaccesspolicy_binding

Query-parameters:

args

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/vpnglobal_vpnclientlessaccesspolicy_binding?**args=policyname:<String_value>,secondary:<Boolean_value>,groupextraction:<Boolean_value>**

HTTP Method:DELETE

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue>

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error

get

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/vpnglobal_vpnclientlessaccesspolicy_binding

Query-parameters:

filter

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/vpnglobal_vpnclientlessaccesspolicy_binding?**filter=property-name1:property-value1,property-name2:property-value2**

Use this query-parameter to get the filtered set of vpnglobal_vpnclientlessaccesspolicy_binding resources configured on Citrix ADC. Filtering can be done on any of the properties of the resource.

pagination

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/vpnglobal_vpnclientlessaccesspolicy_binding?**pagesize=#no;pageno=#no**

Use this query-parameter to get the vpnglobal_vpnclientlessaccesspolicy_binding resources in chunks.

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "vpnglobal_vpnclientlessaccesspolicy_binding": [ {

"priority":<Double_value>,

"globalbindtype":<String_value>,

"policyname":<String_value>,

"gotopriorityexpression":<String_value>,

"secondary":<Boolean_value>,

"type":<String_value>,

"groupextraction":<Boolean_value>

}]}
```

get (all)

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/vpnglobal_vpnclientlessaccesspolicy_binding

Query-parameters:

bulkbindings

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/vpnglobal_vpnclientlessaccesspolicy_binding?**bulkbindings=yes**

NITRO allows you to fetch bindings in bulk.

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "vpnglobal_vpnclientlessaccesspolicy_binding": [ {

"priority":<Double_value>,

"globalbindtype":<String_value>,

"policyname":<String_value>,

"gotopriorityexpression":<String_value>,

"secondary":<Boolean_value>,

"type":<String_value>,

"groupextraction":<Boolean_value>

}]}
```

count

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/vpnglobal_vpnclientlessaccesspolicy_binding?**count=yes**

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "vpnglobal_vpnclientlessaccesspolicy_binding": [ { "__count": "#no"} ] }
```

vpnglobal_vpneula_binding

Binding object showing the vpneula that can be bound to vpnglobal.

Properties

(click to see Operations)

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
eula	<String>	Read-write	Name of the EULA bound to vpnglobal.
gotopriorityexpression	<String>	Read-write	Applicable only to advance vpn session policy. An expression

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
			or other value specifying the priority of the next policy which will get evaluated if the current policy rule evaluates to TRUE.
__count	<Double>	Read-write	count parameter

Operations

(click to see Properties)

ADD:| DELETE:| GET| GET (ALL)| COUNT

Some options that you can use for each operations:

- **Getting warnings in response:**NITRO allows you to get warnings in an operation by specifying the "warning" query parameter as "yes". For example, to get warnings while connecting to the Citrix ADC appliance, the URL is as follows:

  http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/login?warning=yes

  If any, the warnings are displayed in the response payload with the HTTP code "209 X-NITRO-WARNING".

- **Authenticated access for individual NITRO operations:**NITRO allows you to logon to the Citrix ADC appliance to perform individual operations. You can use this option instead of creating a NITRO session (using the login object) and then using that session to perform all operations,

  To do this, you must specify the username and password in the request header of the NITRO request as follows:

  X-NITRO-USER:<username>

  X-NITRO-PASS:<password>

  Note: In such cases, make sure that the request header DOES not include the following:

  Cookie:NITRO_AUTH_TOKEN=<tokenvalue>

`*Note:` *

Mandatory parameters are marked in redand placeholder content is marked in <green>.

add

URL:http://<Citrix-ADC-IP-address(NSIP)/nitro/v1/config/vpnglobal_vpneula_binding

HTTP Method:PUT

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Content-Type:application/json

Request Payload:

```
{
"vpnglobal_vpneula_binding":{
"gotopriorityexpression":<String_value>,
"eula":<String_value>
}}
```

Response:

HTTP Status Code on Success: 201 Created HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error

delete

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/vpnglobal_vpneula_binding

Query-parameters:

args

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/vpnglobal_vpneula_binding?**args=eula:<String_value>**

HTTP Method:DELETE

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue>

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error

get

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/vpnglobal_vpneula_binding

Query-parameters:

filter

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/vpnglobal_vpneula_binding?**filter=property-name1:property-value1,property-name2:property-value2**

Use this query-parameter to get the filtered set of vpnglobal_vpneula_binding resources configured on Citrix ADC. Filtering can be done on any of the properties of the resource.

pagination

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/vpnglobal_vpneula_binding?**pagesize=#no;pageno=#no**

Use this query-parameter to get the vpnglobal_vpneula_binding resources in chunks.

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "vpnglobal_vpneula_binding": [ {

"eula":<String_value>,

"gotopriorityexpression":<String_value>

}]}
```

get (all)

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/vpnglobal_vpneula_binding

Query-parameters:

bulkbindings

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/vpnglobal_vpneula_binding?**bulkbindings=yes**

NITRO allows you to fetch bindings in bulk.

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "vpnglobal_vpneula_binding": [ {

"eula":<String_value>,

"gotopriorityexpression":<String_value>

}]}
```

count

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/vpnglobal_vpneula_binding?**count=yes**

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "vpnglobal_vpneula_binding": [ { "__count": "#no"} ] }
```

vpnglobal_vpnintranetapplication_binding

Binding object showing the vpnintranetapplication that can be bound to vpnglobal.

Properties

(click to see Operations)

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
gotopriorityexpression	<String>	Read-write	Applicable only to advance vpn session policy. An expression or other value specifying the priority of the next policy which will get evaluated if the current policy rule evaluates to TRUE.
intranetapplication	<String>	Read-write	The intranet vpn application.
__count	<Double>	Read-write	count parameter

Operations

(click to see Properties)

ADD:| DELETE:| GET| GET (ALL)| COUNT

Some options that you can use for each operations:

- **Getting warnings in response:**NITRO allows you to get warnings in an operation by specifying the "warning" query parameter as "yes". For example, to get warnings while connecting to the Citrix ADC appliance, the URL is as follows:

  http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/login?warning=yes

  If any, the warnings are displayed in the response payload with the HTTP code "209 X-NITRO-WARNING".

- **Authenticated access for individual NITRO operations:**NITRO allows you to logon to the Citrix ADC appliance to perform individual operations. You can use this option instead of creating a NITRO session (using the login object) and then using that session to perform all operations,

  To do this, you must specify the username and password in the request header of the NITRO request as follows:

  X-NITRO-USER:<username>

X-NITRO-PASS:<password>

Note: In such cases, make sure that the request header DOES not include the following:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue>

`*Note:`*

Mandatory parameters are marked in redand placeholder content is marked in <green>.

add

URL:http://<Citrix-ADC-IP-address(NSIP)/nitro/v1/config/vpnglobal_vpnintranetapplication_binding

HTTP Method:PUT

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Content-Type:application/json

Request Payload:

```
{
"vpnglobal_vpnintranetapplication_binding":{
"intranetapplication":<String_value>,
"gotopriorityexpression":<String_value>
}}
```

Response:

HTTP Status Code on Success: 201 Created HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error

delete

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/vpnglobal_vpnintranetapplication_binding

Query-parameters:

args

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/vpnglobal_vpnintranetapplication_binding?**args=intranetapplication:<String_value>**

HTTP Method:DELETE

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue>

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error

get

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/vpnglobal_vpnintranetapplication_binding

Query-parameters:

filter

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/vpnglobal_vpnintranetapplication_binding?**filter=property-name1:property-value1,property-name2:property-value2**

Use this query-parameter to get the filtered set of vpnglobal_vpnintranetapplication_binding resources configured on Citrix ADC. Filtering can be done on any of the properties of the resource.

pagination

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/vpnglobal_vpnintranetapplication_binding?**pagesize=#no;pageno=#no**

Use this query-parameter to get the vpnglobal_vpnintranetapplication_binding resources in chunks.

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "vpnglobal_vpnintranetapplication_binding": [ {

"gotopriorityexpression":<String_value>,

"intranetapplication":<String_value>

}]}
```

get (all)

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/vpnglobal_vpnintranetapplication_binding

Query-parameters:

bulkbindings

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/vpnglobal_vpnintranetapplication_binding?**bulkbindings=yes**

NITRO allows you to fetch bindings in bulk.

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "vpnglobal_vpnintranetapplication_binding": [ {

"gotopriorityexpression":<String_value>,

"intranetapplication":<String_value>

}]}
```

count

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/vpnglobal_vpnintranetapplication_binding?**count=yes**

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "vpnglobal_vpnintranetapplication_binding": [ { "__count": "#no"} ] }
```

vpnglobal_vpnnexthopserver_binding

Binding object showing the vpnnexthopserver that can be bound to vpnglobal.

Properties

(click to see Operations)

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
gotopriorityexpression	<String>	Read-write	Applicable only to advance vpn session policy. An expression or other value specifying the priority of the next policy which will get evaluated if the current policy rule evaluates to TRUE.
nexthopserver	<String>	Read-write	The name of the next hop server bound to vpn global.
__count	<Double>	Read-write	count parameter

Operations

(click to see Properties)

ADD:| DELETE:| GET| GET (ALL)| COUNT

Some options that you can use for each operations:

- **Getting warnings in response:**NITRO allows you to get warnings in an operation by specifying the "warning" query parameter as "yes". For example, to get warnings while connecting to the Citrix ADC appliance, the URL is as follows:

  http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/login?warning=yes

If any, the warnings are displayed in the response payload with the HTTP code "209 X-NITRO-WARNING".

- **Authenticated access for individual NITRO operations:**NITRO allows you to logon to the Citrix ADC appliance to perform individual operations. You can use this option instead of creating a NITRO session (using the login object) and then using that session to perform all operations,

  To do this, you must specify the username and password in the request header of the NITRO request as follows:

  X-NITRO-USER:<username>

  X-NITRO-PASS:<password>

  Note: In such cases, make sure that the request header DOES not include the following:

  Cookie:NITRO_AUTH_TOKEN=<tokenvalue>

`*Note:`*

Mandatory parameters are marked in redand placeholder content is marked in <green>.

add

URL:http://<Citrix-ADC-IP-address(NSIP)/nitro/v1/config/vpnglobal_vpnnexthopserver_binding

HTTP Method:PUT

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Content-Type:application/json

Request Payload:

```
{

"vpnglobal_vpnnexthopserver_binding":{

"nexthopserver":<String_value>,

"gotopriorityexpression":<String_value>
```

```
}}
```

Response:

HTTP Status Code on Success: 201 Created HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error

delete

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/vpnglobal_vpnnexthopserver_binding

Query-parameters:

args

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/vpnglobal_vpnnexthopserver_binding?**args=nexthopserver:<String_value>**

HTTP Method:DELETE

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue>

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error

get

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/vpnglobal_vpnnexthopserver_binding

Query-parameters:

filter

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/vpnglobal_vpnnexthopserver_binding?**filter=property-name1:property-value1,property-name2:property-value2**

Use this query-parameter to get the filtered set of vpnglobal_vpnnexthopserver_binding resources configured on Citrix ADC. Filtering can be done on any of the properties of the resource.

pagination

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/vpnglobal_vpnnexthopserver_binding?**pagesize=#no;pageno=#no**

Use this query-parameter to get the vpnglobal_vpnnexthopserver_binding resources in chunks.

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "vpnglobal_vpnnexthopserver_binding": [ {

"gotopriorityexpression":<String_value>,

"nexthopserver":<String_value>

}]}
```

get (all)

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/vpnglobal_vpnnexthopserver_binding

Query-parameters:

bulkbindings

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/vpnglobal_vpnnexthopserver_binding?**bulkbindings=yes**

NITRO allows you to fetch bindings in bulk.

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "vpnglobal_vpnnexthopserver_binding": [ {

"gotopriorityexpression":<String_value>,

"nexthopserver":<String_value>

}]}
```

count

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/vpnglobal_vpnnexthopserver_binding?**count=yes**

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "vpnglobal_vpnnexthopserver_binding": [ { "__count": "#no"} ] }
```

vpnglobal_vpnportaltheme_binding

Binding object showing the vpnportaltheme that can be bound to vpnglobal.

Properties

(click to see Operations)

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
gotopriorityexpression	<String>	Read-write	Applicable only to advance vpn session policy. An expression or other value specifying the priority of the next policy which will get evaluated if the current policy rule evaluates to TRUE.
portaltheme	<String>	Read-write	Name of the portal theme bound to vpnglobal.
__count	<Double>	Read-write	count parameter

Operations

(click to see Properties)

ADD:| DELETE:| GET| GET (ALL)| COUNT

Some options that you can use for each operations:

- **Getting warnings in response:**NITRO allows you to get warnings in an operation by specifying the "warning" query parameter as "yes". For example, to get warnings while connecting to the Citrix ADC appliance, the URL is as follows:

  http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/login?warning=yes

  If any, the warnings are displayed in the response payload with the HTTP code "209 X-NITRO-WARNING".

- **Authenticated access for individual NITRO operations:**NITRO allows you to logon to the Citrix ADC appliance to perform individual operations. You can use this option instead of creating a NITRO session (using the login object) and then using that session to perform all operations,

  To do this, you must specify the username and password in the request header of the NITRO request as follows:

  X-NITRO-USER:<username>

  X-NITRO-PASS:<password>

  Note: In such cases, make sure that the request header DOES not include the following:

  Cookie:NITRO_AUTH_TOKEN=<tokenvalue>

`*Note:`*

Mandatory parameters are marked in redand placeholder content is marked in <green>.

add

URL:http://<Citrix-ADC-IP-address(NSIP)/nitro/v1/config/vpnglobal_vpnportaltheme_binding

HTTP Method:PUT

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Content-Type:application/json

Request Payload:

```
{

"vpnglobal_vpnportaltheme_binding":{

"gotopriorityexpression":<String_value>,

"portaltheme":<String_value>

}}
```

Response:

HTTP Status Code on Success: 201 Created HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error

delete

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/vpnglobal_vpnportaltheme_binding

Query-parameters:

args

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/vpnglobal_vpnportaltheme_binding?**args=portaltheme:<String_value>**

HTTP Method:DELETE

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue>

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error

get

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/vpnglobal_vpnportaltheme_binding

Query-parameters:

filter

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/vpnglobal_vpnportaltheme_binding?**filter=property-name1:property-value1,property-name2:property-value2**

Use this query-parameter to get the filtered set of vpnglobal_vpnportaltheme_binding resources configured on Citrix ADC. Filtering can be done on any of the properties of the resource.

pagination

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/vpnglobal_vpnportaltheme_binding?**pagesize=#no;pageno=#no**

Use this query-parameter to get the vpnglobal_vpnportaltheme_binding resources in chunks.

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "vpnglobal_vpnportaltheme_binding": [ {

"gotopriorityexpression":<String_value>,

"portaltheme":<String_value>

}]}
```

get (all)

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/vpnglobal_vpnportaltheme_binding

Query-parameters:

bulkbindings

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/vpnglobal_vpnportaltheme_binding?**bulkbindings=yes**

NITRO allows you to fetch bindings in bulk.

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "vpnglobal_vpnportaltheme_binding": [ {

"gotopriorityexpression":<String_value>,

"portaltheme":<String_value>

}]}
```

count

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/vpnglobal_vpnportaltheme_binding?**count=yes**

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "vpnglobal_vpnportaltheme_binding": [ { "__count": "#no"} ] }
```

vpnglobal_vpnsessionpolicy_binding

Binding object showing the vpnsessionpolicy that can be bound to vpnglobal.

Properties

(click to see Operations)

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
priority	<Double>	Read-write	Integer specifying the policy's priority. The lower the priority number, the higher the policy's priority. Maximum value for default syntax policies is 2147483647 and for classic policies is 64000.

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
			Minimum value = 0 Maximum value = 2147483647
policyname	<String>	Read-write	The name of the policy.
gotopriorityexpression	<String>	Read-write	Applicable only to advance vpn session policy. An expression or other value specifying the priority of the next policy which will get evaluated if the current policy rule evaluates to TRUE.
secondary	<Boolean>	Read-write	Bind the authentication policy as the secondary policy to use in a two-factor configuration. A user must then authenticate not only to a primary authentication server but also to a secondary authentication server. User groups are aggregated across both authentication servers. The user name must be exactly the same on both authentication servers, but the

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
			authentication servers can require different passwords.
groupextraction	<Boolean>	Read-write	Bind the Authentication policy to a tertiary chain which will be used only for group extraction. The user will not authenticate against this server, and this will only be called it primary and/or secondary authentication has succeeded.
__count	<Double>	Read-write	count parameter

Operations

(click to see Properties)

ADD:| DELETE:| GET| GET (ALL)| COUNT

Some options that you can use for each operations:

- **Getting warnings in response:**NITRO allows you to get warnings in an operation by specifying the "warning" query parameter as "yes". For example, to get warnings while connecting to the Citrix ADC appliance, the URL is as follows:

  http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/login?warning=yes

  If any, the warnings are displayed in the response payload with the HTTP code "209 X-NITRO-WARNING".
- **Authenticated access for individual NITRO operations:**NITRO allows you to logon to the Citrix ADC appliance to perform individual operations. You can use this option instead of creating a NITRO session (using the login object) and then using that session to perform all operations,

To do this, you must specify the username and password in the request header of the NITRO request as follows:

X-NITRO-USER:<username>

X-NITRO-PASS:<password>

Note: In such cases, make sure that the request header DOES not include the following:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue>

`*Note:`*

Mandatory parameters are marked in redand placeholder content is marked in <green>.

add

URL:http://<Citrix-ADC-IP-address(NSIP)/nitro/v1/config/vpnglobal_vpnsessionpolicy_binding

HTTP Method:PUT

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Content-Type:application/json

Request Payload:

```
{

"vpnglobal_vpnsessionpolicy_binding":{

"policyname":<String_value>,

"priority":<Double_value>,

"secondary":<Boolean_value>,

"groupextraction":<Boolean_value>,

"gotopriorityexpression":<String_value>

}}
```

Response:

HTTP Status Code on Success: 201 Created HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error

delete

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/vpnglobal_vpnsessionpolicy_binding

Query-parameters:

args

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/vpnglobal_vpnsessionpolicy_binding?**args=policyname:<String_value>,secondary:<Boolean_value>,groupextraction:<Boolean_value>**

HTTP Method:DELETE

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue>

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error

get

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/vpnglobal_vpnsessionpolicy_binding

Query-parameters:

filter

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/vpnglobal_vpnsessionpolicy_binding?**filter=property-name1:property-value1,property-name2:property-value2**

Use this query-parameter to get the filtered set of vpnglobal_vpnsessionpolicy_binding resources configured on Citrix ADC. Filtering can be done on any of the properties of the resource.

pagination

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/vpnglobal_vpnsessionpolicy_binding?**pagesize=#no;pageno=#no**

Use this query-parameter to get the vpnglobal_vpnsessionpolicy_binding resources in chunks.

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "vpnglobal_vpnsessionpolicy_binding": [ {

"priority":<Double_value>,

"policyname":<String_value>,

"gotopriorityexpression":<String_value>,

"secondary":<Boolean_value>,

"groupextraction":<Boolean_value>

}]}
```

get (all)

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/vpnglobal_vpnsessionpolicy_binding

Query-parameters:

bulkbindings

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/vpnglobal_vpnsessionpolicy_binding?**bulkbindings=yes**

NITRO allows you to fetch bindings in bulk.

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "vpnglobal_vpnsessionpolicy_binding": [ {

"priority":<Double_value>,

"policyname":<String_value>,

"gotopriorityexpression":<String_value>,

"secondary":<Boolean_value>,

"groupextraction":<Boolean_value>

}]}
```

count

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/vpnglobal_vpnsessionpolicy_binding?**count=yes**

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "vpnglobal_vpnsessionpolicy_binding": [ { "__count": "#no"} ] }
```

vpnglobal_vpntrafficpolicy_binding

Binding object showing the vpntrafficpolicy that can be bound to vpnglobal.

Properties

(click to see Operations)

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
priority	<Double>	Read-write	Integer specifying the policy's priority. The lower the priority number, the higher the policy's priority. Maximum value for default syntax policies is 2147483647 and for classic policies is 64000.

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
			Minimum value = 0 Maximum value = 2147483647
policyname	<String>	Read-write	The name of the policy.
gotopriorityexpression	<String>	Read-write	Applicable only to advance vpn session policy. An expression or other value specifying the priority of the next policy which will get evaluated if the current policy rule evaluates to TRUE.
secondary	<Boolean>	Read-write	Bind the authentication policy as the secondary policy to use in a two-factor configuration. A user must then authenticate not only to a primary authentication server but also to a secondary authentication server. User groups are aggregated across both authentication servers. The user name must be exactly the same on both authentication servers, but the

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
			authentication servers can require different passwords.
groupextraction	<Boolean>	Read-write	Bind the Authentication policy to a tertiary chain which will be used only for group extraction. The user will not authenticate against this server, and this will only be called it primary and/or secondary authentication has succeeded.
__count	<Double>	Read-write	count parameter

Operations

(click to see Properties)

ADD:| DELETE:| GET| GET (ALL)| COUNT

Some options that you can use for each operations:

- **Getting warnings in response:**NITRO allows you to get warnings in an operation by specifying the "warning" query parameter as "yes". For example, to get warnings while connecting to the Citrix ADC appliance, the URL is as follows:

  http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/login?warning=yes

  If any, the warnings are displayed in the response payload with the HTTP code "209 X-NITRO-WARNING".
- **Authenticated access for individual NITRO operations:**NITRO allows you to logon to the Citrix ADC appliance to perform individual operations. You can use this option instead of creating a NITRO session (using the login object) and then using that session to perform all operations,

To do this, you must specify the username and password in the request header of the NITRO request as follows:

X-NITRO-USER:<username>

X-NITRO-PASS:<password>

Note: In such cases, make sure that the request header DOES not include the following:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue>

`*Note:`*

Mandatory parameters are marked in redand placeholder content is marked in <green>.

add

URL:http://<Citrix-ADC-IP-address(NSIP)/nitro/v1/config/vpnglobal_vpntrafficpolicy_binding

HTTP Method:PUT

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Content-Type:application/json

Request Payload:

```
{

"vpnglobal_vpntrafficpolicy_binding":{

"policyname":<String_value>,

"priority":<Double_value>,

"secondary":<Boolean_value>,

"groupextraction":<Boolean_value>,

"gotopriorityexpression":<String_value>

}}
```

Response:

HTTP Status Code on Success: 201 Created HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error

delete

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/vpnglobal_vpntrafficpolicy_binding

Query-parameters:

args

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/vpnglobal_vpntrafficpolicy_binding?**args=policyname:<String_value>,secondary:<Boolean_value>,groupextraction:<Boolean_value>**

HTTP Method:DELETE

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue>

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error

get

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/vpnglobal_vpntrafficpolicy_binding

Query-parameters:

filter

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/vpnglobal_vpntrafficpolicy_binding?**filter=property-name1:property-value1,property-name2:property-value2**

Use this query-parameter to get the filtered set of vpnglobal_vpntrafficpolicy_binding resources configured on Citrix ADC. Filtering can be done on any of the properties of the resource.

pagination

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/vpnglobal_vpntrafficpolicy_binding?**pagesize=#no;pageno=#no**

Use this query-parameter to get the vpnglobal_vpntrafficpolicy_binding resources in chunks.

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "vpnglobal_vpntrafficpolicy_binding": [ {

"priority":<Double_value>,

"policyname":<String_value>,

"gotopriorityexpression":<String_value>,

"secondary":<Boolean_value>,

"groupextraction":<Boolean_value>

}]}
```

get (all)

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/vpnglobal_vpntrafficpolicy_binding

Query-parameters:

bulkbindings

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/vpnglobal_vpntrafficpolicy_binding?**bulkbindings=yes**

NITRO allows you to fetch bindings in bulk.

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "vpnglobal_vpntrafficpolicy_binding": [ {

"priority":<Double_value>,

"policyname":<String_value>,

"gotopriorityexpression":<String_value>,

"secondary":<Boolean_value>,

"groupextraction":<Boolean_value>

}]}
```

count

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/vpnglobal_vpntrafficpolicy_binding?**count=yes**

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "vpnglobal_vpntrafficpolicy_binding": [ { "__count": "#no"} ] }
```

vpnglobal_vpnurl_binding

Binding object showing the vpnurl that can be bound to vpnglobal.

Properties

(click to see Operations)

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
urlname	<String>	Read-write	The intranet url.
gotopriorityexpression	<String>	Read-write	Applicable only to advance vpn session policy. An expression or other value specifying the priority of the next policy which will get evaluated if the current policy rule evaluates to TRUE.

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
__count	<Double>	Read-write	count parameter

Operations

(click to see Properties)

ADD:| DELETE:| GET| GET (ALL)| COUNT

Some options that you can use for each operations:

- **Getting warnings in response:**NITRO allows you to get warnings in an operation by specifying the "warning" query parameter as "yes". For example, to get warnings while connecting to the Citrix ADC appliance, the URL is as follows:

  http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/login?warning=yes

  If any, the warnings are displayed in the response payload with the HTTP code "209 X-NITRO-WARNING".
- **Authenticated access for individual NITRO operations:**NITRO allows you to logon to the Citrix ADC appliance to perform individual operations. You can use this option instead of creating a NITRO session (using the login object) and then using that session to perform all operations,

  To do this, you must specify the username and password in the request header of the NITRO request as follows:

  X-NITRO-USER:<username>

  X-NITRO-PASS:<password>

  Note: In such cases, make sure that the request header DOES not include the following:

  Cookie:NITRO_AUTH_TOKEN=<tokenvalue>

`*Note:`*

Mandatory parameters are marked in redand placeholder content is marked in <green>.

add

URL:http://<Citrix-ADC-IP-address(NSIP)/nitro/v1/config/vpnglobal_vpnurl_binding

HTTP Method:PUT

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Content-Type:application/json

Request Payload:

```
{

"vpnglobal_vpnurl_binding":{

"urlname":<String_value>,

"gotopriorityexpression":<String_value>

}}
```

Response:

HTTP Status Code on Success: 201 Created HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error

delete

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/vpnglobal_vpnurl_binding

Query-parameters:

args

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/vpnglobal_vpnurl_binding?**args=urlname:<String_value>**

HTTP Method:DELETE

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue>

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error

get

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/vpnglobal_vpnurl_binding

Query-parameters:

filter

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/vpnglobal_vpnurl_binding?**filter=property-name1:property-value1,property-name2:property-value2**

Use this query-parameter to get the filtered set of vpnglobal_vpnurl_binding resources configured on Citrix ADC. Filtering can be done on any of the properties of the resource.

pagination

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/vpnglobal_vpnurl_binding?**pagesize=#no;pageno=#no**

Use this query-parameter to get the vpnglobal_vpnurl_binding resources in chunks.

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "vpnglobal_vpnurl_binding": [ {
```

```
"urlname":<String_value>,

"gotopriorityexpression":<String_value>

}]}
```

get (all)

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/vpnglobal_vpnurl_binding

Query-parameters:

bulkbindings

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/vpnglobal_vpnurl_binding?**bulkbindings=yes**

NITRO allows you to fetch bindings in bulk.

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "vpnglobal_vpnurl_binding": [ {

"urlname":<String_value>,

"gotopriorityexpression":<String_value>

}]}
```

count

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/vpnglobal_vpnurl_binding?**count=yes**

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "vpnglobal_vpnurl_binding": [ { "__count": "#no"} ] }
```

vpnicaconnection

Configuration for active ica connections resource.

Properties

(click to see Operations)

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
username	<String>	Read-write	User name for which to display connections. Minimum length = 1

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
transproto	<String>	Read-write	Transport type for the existing Existing ICA conenction. Possible values = TCP, UDP
nodeid	<Double>	Read-write	Unique number that identifies the cluster node. Minimum value = 0 Maximum value = 31
all	<Boolean>	Read-write	Terminate all active icaconnections.
domain	<String>	Read-only	The domain name.
srcip	<String>	Read-only	The client IP address.
srcport	<Integer>	Read-only	The client port. Range 1 - 65535 * in CLI is represented as 65535 in NITRO API
destip	<String>	Read-only	The CPS server IP address.
destport	<Integer>	Read-only	The CPS server port. Range 1 - 65535 * in CLI is represented as 65535 in NITRO API

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
peid	<Double>	Read-only	Core id of the session owner.
__count	<Double>	Read-only	count parameter

Operations

(click to see Properties)

KILL| GET (ALL)| COUNT

Some options that you can use for each operations:

- **Getting warnings in response:**NITRO allows you to get warnings in an operation by specifying the "warning" query parameter as "yes". For example, to get warnings while connecting to the Citrix ADC appliance, the URL is as follows:

  http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/login?warning=yes

  If any, the warnings are displayed in the response payload with the HTTP code "209 X-NITRO-WARNING".

- **Authenticated access for individual NITRO operations:**NITRO allows you to logon to the Citrix ADC appliance to perform individual operations. You can use this option instead of creating a NITRO session (using the login object) and then using that session to perform all operations,

  To do this, you must specify the username and password in the request header of the NITRO request as follows:

  X-NITRO-USER:<username>

  X-NITRO-PASS:<password>

  Note: In such cases, make sure that the request header DOES not include the following:

  Cookie:NITRO_AUTH_TOKEN=<tokenvalue>

`*Note:`*

Mandatory parameters are marked in redand placeholder content is marked in <green>.

kill

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/vpnicaconnection?**action=kill**

HTTP Method:POST

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Content-Type:application/json

Request Payload:

```
{"vpnicaconnection":{

"username":<String_value>,

"transproto":<String_value>,

"all":<Boolean_value>

}}
```

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error

get (all)

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/vpnicaconnection

Query-parameters:

args

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/vpnicaconnection?**args=username:<String_value>,transproto:<String_value>,nodeid:<Double_value>**

Use this query-parameter to get vpnicaconnection resources based on additional properties.

attrs

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/vpnicaconnection?**attrs=property-name1,property-name2**

Use this query parameter to specify the resource details that you want to retrieve.

filter

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/vpnicaconnection?**filter=property-name1:property-val1,property-name2:property-val2**

Use this query-parameter to get the filtered set of vpnicaconnection resources configured on Citrix ADC. Filtering can be done on any of the properties of the resource.

view

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/vpnicaconnection?**view=summary**

Note: By default, the retrieved results are displayed in detail view (?view=detail).

pagination

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/vpnicaconnection?**pagesize=#no;pageno=#no**

Use this query-parameter to get the vpnicaconnection resources in chunks.

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "vpnicaconnection": [ {
```

```
username:<String_value>,transproto:<String_value>,nodeid:<Double_value>"do
main":<String_value>,

"srcip":<String_value>,

"srcport":<Integer_value>,

"destip":<String_value>,

"destport":<Integer_value>,

"peid":<Double_value>

}]}
```

count

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/vpnicaconnection?**count=yes**

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "vpnicaconnection": [ { "__count": "#no"} ] }
```

vpnicadtlsconnection

Configuration for active ica connections resource.

Properties

(click to see Operations)

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
username	<String>	Read-write	User name for which to display connections. Minimum length = 1
nodeid	<Double>	Read-write	Unique number that identifies the cluster node. Minimum value = 0 Maximum value = 31
domain	<String>	Read-only	The domain name.
srcip	<String>	Read-only	The client IP address.
srcport	<Integer>	Read-only	The client port. Range 1 - 65535 * in CLI is represented as 65535 in NITRO API
destip	<String>	Read-only	The CPS server IP address.
destport	<Integer>	Read-only	The CPS server port. Range 1 - 65535

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
			* in CLI is represented as 65535 in NITRO API
channelnumber	<Double>	Read-only	The channel number.
peid	<Double>	Read-only	Core id of the session owner.
__count	<Double>	Read-only	count parameter

Operations

(click to see Properties)

GET (ALL)| COUNT

Some options that you can use for each operations:

- **Getting warnings in response:**NITRO allows you to get warnings in an operation by specifying the "warning" query parameter as "yes". For example, to get warnings while connecting to the Citrix ADC appliance, the URL is as follows:

  http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/login?warning=yes

  If any, the warnings are displayed in the response payload with the HTTP code "209 X-NITRO-WARNING".

- **Authenticated access for individual NITRO operations:**NITRO allows you to logon to the Citrix ADC appliance to perform individual operations. You can use this option instead of creating a NITRO session (using the login object) and then using that session to perform all operations,

  To do this, you must specify the username and password in the request header of the NITRO request as follows:

  X-NITRO-USER:<username>

  X-NITRO-PASS:<password>

  Note: In such cases, make sure that the request header DOES not include the following:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue>

`*Note:`*

Mandatory parameters are marked in redand placeholder content is marked in <green>.

get (all)

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/vpnicadtlsconnection

Query-parameters:

args

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/vpnicadtlsconnection?**args=username:<String_value>,nodeid:<Double_value>**

Use this query-parameter to get vpnicadtlsconnection resources based on additional properties.

attrs

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/vpnicadtlsconnection?**attrs=property-name1,property-name2**

Use this query parameter to specify the resource details that you want to retrieve.

filter

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/vpnicadtlsconnection?**filter=property-name1:property-val1,property-name2:property-val2**

Use this query-parameter to get the filtered set of vpnicadtlsconnection resources configured on Citrix ADC. Filtering can be done on any of the properties of the resource.

view

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/vpnicadtlsconnection?**view=summary**

Note: By default, the retrieved results are displayed in detail view (?view=detail).

pagination

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/vpnicadtlsconnection?**pagesize=#no;pageno=#no**

Use this query-parameter to get the vpnicadtlsconnection resources in chunks.

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "vpnicadtlsconnection": [ {

username:<String_value>,nodeid:<Double_value>"domain":<String_value>,

"srcip":<String_value>,

"srcport":<Integer_value>,

"destip":<String_value>,

"destport":<Integer_value>,

"channelnumber":<Double_value>,

"peid":<Double_value>

}]}
```

count

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/vpnicadtlsconnection?**count=yes**

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "vpnicadtlsconnection": [ { "__count": "#no"} ] }
```

vpnintranetapplication

Configuration for SSLVPN intranet application resource.

Properties

(click to see Operations)

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
intranetapplication	<String>	Read-write	Name of the intranet application. Minimum length = 1 Maximum length = 31
protocol	<String>	Read-write	Protocol used by the intranet application. If protocol is set to BOTH, TCP and UDP traffic is allowed. Possible values = TCP, UDP, ANY

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
destip	<String>	Read-write	Destination IP address, IP range, or host name of the intranet application. This address is the server IP address. Minimum length = 1
netmask	<String>	Read-write	Destination subnet mask for the intranet application.
iprange	<String>	Read-write	If you have multiple servers in your network, such as web, email, and file shares, configure an intranet application that includes the IP range for all the network applications. This allows users to access all the intranet applications contained in the IP address range. Minimum length = 1
hostname	<String>	Read-write	Name of the host for which to configure interception. The names are resolved during interception when users log on with the Citrix Gateway Plug-in. Minimum length = 1

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
clientapplication	<String[]>	Read-write	Names of the client applications, such as PuTTY and Xshell. Minimum length = 1
spoofiip	<String>	Read-write	IP address that the intranet application will use to route the connection through the virtual adapter. Default value: ON Possible values = ON, OFF
destport	<String>	Read-write	Destination TCP or UDP port number for the intranet application. Use a hyphen to specify a range of port numbers, for example 90-95. Minimum length = 1
interception	<String>	Read-write	Interception mode for the intranet application or resource. Correct value depends on the type of client software used to make connections. If the interception mode is set to TRANSPARENT, users connect with the Citrix Gateway Plug-in for Windows. With the PROXY setting, users

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
			connect with the Citrix Gateway Plug-in for Java. Possible values = PROXY, TRANSPARENT
srcip	<String>	Read-write	Source IP address. Required if interception mode is set to PROXY. Default is the loopback address, 127.0.0.1. Minimum length = 1
srcport	<Integer>	Read-write	Source port for the application for which the Citrix Gateway virtual server proxies the traffic. If users are connecting from a device that uses the Citrix Gateway Plug-in for Java, applications must be configured manually by using the source IP address and TCP port values specified in the intranet application profile. If a port value is not set, the destination port value is used. Minimum value = 1
ipaddress	<String>	Read-only	The IP address for the application. This address is the real

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
			application server IP address.
__count	<Double>	Read-only	count parameter

Operations

(click to see Properties)

ADD| DELETE| GET (ALL)| GET| COUNT

Some options that you can use for each operations:

- **Getting warnings in response:**NITRO allows you to get warnings in an operation by specifying the "warning" query parameter as "yes". For example, to get warnings while connecting to the Citrix ADC appliance, the URL is as follows:

  http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/login?warning=yes

  If any, the warnings are displayed in the response payload with the HTTP code "209 X-NITRO-WARNING".
- **Authenticated access for individual NITRO operations:**NITRO allows you to logon to the Citrix ADC appliance to perform individual operations. You can use this option instead of creating a NITRO session (using the login object) and then using that session to perform all operations,

  To do this, you must specify the username and password in the request header of the NITRO request as follows:

  X-NITRO-USER:<username>

  X-NITRO-PASS:<password>

  Note: In such cases, make sure that the request header DOES not include the following:

  Cookie:NITRO_AUTH_TOKEN=<tokenvalue>

`*Note:`*

Mandatory parameters are marked in redand placeholder content is marked in <green>.

add

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/vpnintranetapplication

HTTP Method:POST

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Content-Type:application/json

Request Payload:

```
{"vpnintranetapplication":{

<b>"intranetapplication":<String_value>,

</b>"protocol":<String_value>,

"destip":<String_value>,

"netmask":<String_value>,

"iprange":<String_value>,

"hostname":<String_value>,

"clientapplication":<String[]_value>,

"spoofiip":<String_value>,

"destport":<String_value>,

"interception":<String_value>,

"srcip":<String_value>,

"srcport":<Integer_value>

}}
```

Response:

HTTP Status Code on Success: 201 Created HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error

delete

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/vpnintranetapplication/intranetapplication_value<String>

HTTP Method:DELETE

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue>

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error

get (all)

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/vpnintranetapplication

Query-parameters:

attrs

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/vpnintranetapplication?**attrs=property-name1,property-name2**

Use this query parameter to specify the resource details that you want to retrieve.

filter

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/vpnintranetapplication?**filter=property-name1:property-val1,property-name2:property-val2**

Use this query-parameter to get the filtered set of vpnintranetapplication resources configured on Citrix ADC. Filtering can be done on any of the properties of the resource.

view

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/vpnintranetapplication?**view=summary**

Note: By default, the retrieved results are displayed in detail view (?view=detail).

pagination

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/vpnintranetapplication?**pagesize=#no;pageno=#no**

Use this query-parameter to get the vpnintranetapplication resources in chunks.

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "vpnintranetapplication": [ {

"intranetapplication":<String_value>,

"protocol":<String_value>,

"destip":<String_value>,

"netmask":<String_value>,

"ipaddress":<String_value>,

"hostname":<String_value>,

"destport":<String_value>,

"clientapplication":<String[]_value>,
```

```
"spoofiip":<String_value>,

"interception":<String_value>,

"srcip":<String_value>,

"srcport":<Integer_value>

}]}
```

get

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/vpnintranetapplication/intranetapplication_value<String>

Query-parameters:

attrs

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/vpnintranetapplication/intranetapplication_value<String>?**attrs=property-name1,property-name2**

Use this query parameter to specify the resource details that you want to retrieve.

view

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/vpnintranetapplication/intranetapplication_value<String>?**view=summary**

Note: By default, the retrieved results are displayed in detail view (?view=detail).

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "vpnintranetapplication": [ {

"intranetapplication":<String_value>,

"protocol":<String_value>,

"destip":<String_value>,

"netmask":<String_value>,

"ipaddress":<String_value>,

"hostname":<String_value>,

"destport":<String_value>,

"clientapplication":<String[]_value>,

"spoofiip":<String_value>,

"interception":<String_value>,

"srcip":<String_value>,

"srcport":<Integer_value>

}]}
```

count

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/vpnintranetapplication?**count=yes**

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "vpnintranetapplication": [ { "__count": "#no"} ] }
```

vpnnexthopserver

Configuration for Next Hop Server resource.

Properties

(click to see Operations)

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
name	<String>	Read-write	Name for the Citrix Gateway appliance in the first DMZ. Minimum length = 1 Maximum length = 32
nexthopip	<String>	Read-write	IP address of the Citrix Gateway proxy in the second DMZ.
nexthopfqdn	<String>	Read-write	FQDN of the Citrix Gateway proxy in the second DMZ. Minimum length = 1

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
resaddresstype	<String>	Read-write	Address Type (IPV4/IPv6) of DNS name of nextHopServer FQDN. Minimum length = 1 Possible values = IPV4, IPV6
nexthopport	<Integer>	Read-write	Port number of the Citrix Gateway proxy in the second DMZ. Minimum value = 1 Maximum value = 65535
secure	<String>	Read-write	Use of a secure port, such as 443, for the double-hop configuration. Default value: OFF Possible values = ON, OFF
__count	<Double>	Read-only	count parameter

Operations

(click to see Properties)

ADD| DELETE| GET (ALL)| GET| COUNT

Some options that you can use for each operations:

- **Getting warnings in response:**NITRO allows you to get warnings in an operation by specifying the "warning" query parameter as "yes". For example, to get warnings while connecting to the Citrix ADC appliance, the URL is as follows:

  http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/login?warning=yes

If any, the warnings are displayed in the response payload with the HTTP code "209 X-NITRO-WARNING".

- **Authenticated access for individual NITRO operations:**NITRO allows you to logon to the Citrix ADC appliance to perform individual operations. You can use this option instead of creating a NITRO session (using the login object) and then using that session to perform all operations,

  To do this, you must specify the username and password in the request header of the NITRO request as follows:

  X-NITRO-USER:<username>

  X-NITRO-PASS:<password>

  Note: In such cases, make sure that the request header DOES not include the following:

  Cookie:NITRO_AUTH_TOKEN=<tokenvalue>

`*Note:`*

Mandatory parameters are marked in redand placeholder content is marked in <green>.

add

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/vpnnexthopserver

HTTP Method:POST

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Content-Type:application/json

Request Payload:

```
{"vpnnexthopserver":{

<b>"name":<String_value>,

</b>"nexthopip":<String_value>,

"nexthopfqdn":<String_value>,
```

```
"resaddresstype":<String_value>,

<b>"nexthopport":<Integer_value>,

</b>"secure":<String_value>

}}
```

Response:

HTTP Status Code on Success: 201 Created HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error

delete

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/vpnnexthopserver/name_value<String>

HTTP Method:DELETE

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue>

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error

get (all)

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/vpnnexthopserver

Query-parameters:

attrs

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/vpnnexthopserver?**attrs=property-name1,property-name2**

Use this query parameter to specify the resource details that you want to retrieve.

filter

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/vpnnexthopserver?**filter=property-name1:property-val1,property-name2:property-val2**

Use this query-parameter to get the filtered set of vpnnexthopserver resources configured on Citrix ADC. Filtering can be done on any of the properties of the resource.

view

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/vpnnexthopserver?**view=summary**

Note: By default, the retrieved results are displayed in detail view (?view=detail).

pagination

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/vpnnexthopserver?**pagesize=#no;pageno=#no**

Use this query-parameter to get the vpnnexthopserver resources in chunks.

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "vpnnexthopserver": [ {

"name":<String_value>,

"nexthopip":<String_value>,

"nexthopfqdn":<String_value>,
```

```
"resaddresstype":<String_value>,

"nexthopport":<Integer_value>,

"secure":<String_value>

}]}
```

get

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/vpnnexthopserver/name_value<String>

Query-parameters:

attrs

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/vpnnexthopserver/name_value<String>?**attrs=property-name1,property-name2**

Use this query parameter to specify the resource details that you want to retrieve.

view

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/vpnnexthopserver/name_value<String>?**view=summary**

Note: By default, the retrieved results are displayed in detail view (?view=detail).

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "vpnnexthopserver": [ {

"name":<String_value>,

"nexthopip":<String_value>,

"nexthopfqdn":<String_value>,

"resaddresstype":<String_value>,

"nexthopport":<Integer_value>,

"secure":<String_value>

}]}
```

count

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/vpnnexthopserver?**count=yes**

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "vpnnexthopserver": [ { "__count": "#no"} ] }
```

vpnparameter

Configuration for VPN parameter resource.

Properties

(click to see Operations)

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
httpport	<Integer[]>	Read-write	Destination port numbers other than port 80, added as a comma-separated list. Traffic to these ports is processed as HTTP traffic, which allows functionality, such as HTTP authorization and single sign-on to a web application to work. Minimum value = 1
winsip	<String>	Read-write	WINS server IP address to add to Citrix Gateway for name resolution. Minimum length = 1
dnsvservername	<String>	Read-write	Name of the DNS virtual server for the user session. Minimum length = 1
splitdns	<String>	Read-write	Route the DNS requests to the local DNS server configured on the user device, or Citrix

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
			Gateway (remote), or both. Possible values = LOCAL, REMOTE, BOTH
icauseraccounting	<String>	Read-write	The name of the radiusPolicy to use for RADIUS user accounting info on the session.
sesstimeout	<Double>	Read-write	Number of minutes after which the session times out. Default value: 30 Minimum value = 1 Maximum value = 65535
clientsecurity	<String>	Read-write	Specify the client security check for the user device to permit a Citrix Gateway session. The web address or IP address is not included in the expression for the client security check.
clientsecuritygroup	<String>	Read-write	The client security group that will be assigned on failure of the client security check. Users can in general be organized into Groups. In this case, the Client

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
			Security Group may have a more restrictive security policy. Minimum length = 1
clientsecuritymessage	<String>	Read-write	The client security message that will be displayed on failure of the client security check. Minimum length = 1 Maximum length = 127
clientsecuritylog	<String>	Read-write	Specifies whether or not to display all failed Client Security scans to the end user. Default value: OFF Possible values = ON, OFF
smartgroup	<String>	Read-write	This is the default group that is chosen when the authentication succeeds in addition to extracted groups. Minimum length = 1 Maximum length = 64
splittunnel	<String>	Read-write	Send, through the tunnel, traffic only for intranet applications that are defined in

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
			Citrix Gateway. Route all other traffic directly to the Internet. The OFF setting routes all traffic through Citrix Gateway. With the REVERSE setting, intranet applications define the network traffic that is not intercepted. All network traffic directed to internal IP addresses bypasses the VPN tunnel, while other traffic goes through Citrix Gateway. Reverse split tunneling can be used to log all non-local LAN traffic. For example, if users have a home network and are logged on through the Citrix Gateway Plug-in, network traffic destined to a printer or another device within the home network is not intercepted. Default value: OFF Possible values = ON, OFF, REVERSE

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
locallanaccess	<String>	Read-write	Set local LAN access. If split tunneling is OFF, and you set local LAN access to ON, the local client can route traffic to its local interface. When the local area network switch is specified, this combination of switches is useful. The client can allow local LAN access to devices that commonly have non-routable addresses, such as local printers or local file servers. Default value: OFF Possible values = ON, OFF
rfc1918	<String>	Read-write	As defined in the local area network, allow only the following local area network addresses to bypass the VPN tunnel when the local LAN access feature is enabled: * 10.*.*.*, * 172.16.*.*, * 192.168.*.*. Default value: OFF

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
			Possible values = ON, OFF
spoofiip	<String>	Read-write	Indicate whether or not the application requires IP spoofing, which routes the connection to the intranet application through the virtual adapter. Default value: ON Possible values = ON, OFF
killconnections	<String>	Read-write	Specify whether the Citrix Gateway Plug-in should disconnect all preexisting connections, such as the connections existing before the user logged on to Citrix Gateway, and prevent new incoming connections on the Citrix Gateway Plug-in for Windows and MAC when the user is connected to Citrix Gateway and split tunneling is disabled. Default value: OFF Possible values = ON, OFF

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
transparentintercepti on	<String>	Read-write	Allow access to network resources by using a single IP address and subnet mask or a range of IP addresses. The OFF setting sets the mode to proxy, in which you configure destination and source IP addresses and port numbers. If you are using the Citrix Gateway Plug-in for Windows, set this parameter to ON, in which the mode is set to transparent. If you are using the Citrix Gateway Plug-in for Java, set this parameter to OFF. Default value: OFF Possible values = ON, OFF
windowsclienttype	<String>	Read-write	The Windows client type. Choose between two types of Windows Client a) Application Agent - which always runs in the task bar as a standalone application and also has a supporting service which runs

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
			permanently when installed b) Activex Control - ActiveX control run by Microsoft Internet Explorer. Default value: AGENT Possible values = AGENT, PLUGIN
defaultauthorizationaction	<String>	Read-write	Specify the network resources that users have access to when they log on to the internal network. The default setting for authorization is to deny access to all network resources. Citrix recommends using the default global setting and then creating authorization policies to define the network resources users can access. If you set the default authorization policy to DENY, you must explicitly authorize access to any network resource, which improves security.

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
			Default value: DENY Possible values = ALLOW, DENY
authorizationgroup	<String>	Read-write	Comma-separated list of groups in which the user is placed when none of the groups that the user is a part of is configured on Citrix Gateway. The authorization policy can be bound to these groups to control access to the resources. Minimum length = 1
clientidletimeout	<Double>	Read-write	Time, in minutes, after which to time out the user session if Citrix Gateway does not detect mouse or keyboard activity. Minimum value = 1 Maximum value = 9999
proxy	<String>	Read-write	Set options to apply proxy for accessing the internal resources. Available settings function as follows:

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
			* BROWSER - Proxy settings are configured only in Internet Explorer and Firefox browsers. * NS - Proxy settings are configured on the Citrix ADC. * OFF - Proxy settings are not configured. Possible values = BROWSER, NS, OFF
allprotocolproxy	<String>	Read-write	IP address of the proxy server to use for all protocols supported by Citrix Gateway. Minimum length = 1
httpproxy	<String>	Read-write	IP address of the proxy server to be used for HTTP access for all subsequent connections to the internal network. Minimum length = 1
ftpproxy	<String>	Read-write	IP address of the proxy server to be used for FTP access for all subsequent

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
			connections to the internal network. Minimum length = 1
socksproxy	<String>	Read-write	IP address of the proxy server to be used for SOCKS access for all subsequent connections to the internal network. Minimum length = 1
gopherproxy	<String>	Read-write	IP address of the proxy server to be used for GOPHER access for all subsequent connections to the internal network. Minimum length = 1
sslproxy	<String>	Read-write	IP address of the proxy server to be used for SSL access for all subsequent connections to the internal network. Minimum length = 1
proxyexception	<String>	Read-write	Proxy exception string that will be configured in the browser for bypassing the previously configured proxies. Allowed only if proxy type is Browser. Minimum length = 1

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
proxylocalbypass	<String>	Read-write	Bypass proxy server for local addresses option in Internet Explorer and Firefox proxy server settings. Default value: DISABLED Possible values = ENABLED, DISABLED
clientcleanupprompt	<String>	Read-write	Prompt for client-side cache clean-up when a client-initiated session closes. Default value: ON Possible values = ON, OFF
forcecleanup	<String[]>	Read-write	Force cache clean-up when the user closes a session. You can specify all, none, or any combination of the client-side items. Possible values = none, all, cookie, addressbar, plugin, filesystemapplicatio n, application, applicationdata, clientcertificate, autocomplete, cache

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
clientoptions	<String[]>	Read-write	Display only the configured menu options when you select the "Configure Citrix Gateway" option in the Citrix Gateway Plug-in's system tray icon for Windows. Possible values = none, all, services, filetransfer, configuration
clientconfiguration	<String[]>	Read-write	Allow users to change client Debug logging level in Configuration tab of the Citrix Gateway Plug-in for Windows. Possible values = none, trace
sso	<String>	Read-write	Set single sign-on (SSO) for the session. When the user accesses a server, the user's logon credentials are passed to the server for authentication. NOTE : Authentication mechanisms like Basic-authentication require the user credentials to be sent in plaintext which is not secure

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
			if the server is running on HTTP (instead of HTTPS). By default, Single Sign On is not attempted to a Public IP. Default value: OFF Possible values = ON, OFF
ssocredential	<String>	Read-write	Specify whether to use the primary or secondary authentication credentials for single sign-on to the server. Default value: PRIMARY Possible values = PRIMARY, SECONDARY
windowsautologon	<String>	Read-write	Enable or disable the Windows Auto Logon for the session. If a VPN session is established after this setting is enabled, the user is automatically logged on by using Windows credentials after the system is restarted. Default value: OFF

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
			Possible values = ON, OFF
usemip	<String>	Read-write	Enable or disable the use of a unique IP address alias, or a mapped IP address, as the client IP address for each client session. Allow Citrix Gateway to use the mapped IP address as an intranet IP address when all other IP addresses are not available. When IP pooling is configured and the mapped IP is used as an intranet IP address, the mapped IP address is used when an intranet IP address cannot be assigned. Default value: NS Possible values = NS, OFF
useiip	<String>	Read-write	Define IP address pool options. Available settings function as follows: * SPILLOVER - When an address pool is configured

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
			and the mapped IP is used as an intranet IP address, the mapped IP address is used when an intranet IP address cannot be assigned. * NOSPILLOVER - When intranet IP addresses are enabled and the mapped IP address is not used, the Transfer Login page appears for users who have used all available intranet IP addresses. * OFF - Address pool is not configured. Default value: NOSPILLOVER Possible values = NOSPILLOVER, SPILLOVER, OFF
clientdebug	<String>	Read-write	Set the trace level on Citrix Gateway. Technical support technicians use these debug logs for in-depth debugging and troubleshooting

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
			purposes. Available settings function as follows: * DEBUG - Detailed debug messages are collected and written into the specified file. * STATS - Application audit level error messages and debug statistic counters are written into the specified file. * EVENTS - Application audit-level error messages are written into the specified file. * OFF - Only critical events are logged into the Windows Application Log. Default value: OFF Possible values = debug, stats, events, OFF
loginscript	<String>	Read-write	Path to the logon script that is run when a session is

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
			established. Separate multiple scripts by using comma. A "$" in the path signifies that the word following the "$" is an environment variable. Minimum length = 1
logoutscript	<String>	Read-write	Path to the logout script. Separate multiple scripts by using comma. A "$" in the path signifies that the word following the "$" is an environment variable. Minimum length = 1
homepage	<String>	Read-write	Web address of the home page that appears when users log on. Otherwise, users receive the default home page for Citrix Gateway, which is the Access Interface.
icaproxy	<String>	Read-write	Enable ICA proxy to configure secure Internet access to servers running Citrix XenApp or XenDesktop by using Citrix Receiver instead of the Citrix Gateway Plug-in.

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
			Default value: OFF Possible values = ON, OFF
wihome	<String>	Read-write	Web address of the Web Interface server, such as http://<ipAddress>/Citrix/XenApp, or Receiver for Web, which enumerates the virtualized resources, such as XenApp, XenDesktop, and cloud applications. This web address is used as the home page in ICA proxy mode. If Client Choices is ON, you must configure this setting. Because the user can choose between FullClient and ICAProxy, the user may see a different home page. An Internet web site may appear if the user gets the FullClient option, or a Web Interface site if the user gets the ICAProxy option. If

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
			the setting is not configured, the XenApp option does not appear as a client choice.
wihomeaddresstype	<String>	Read-write	Type of the wihome address(IPV4/V6). Possible values = IPV4, IPV6
citrixreceiverhome	<String>	Read-write	Web address for the Citrix Receiver home page. Configure Citrix Gateway so that when users log on to the appliance, the Citrix Gateway Plug-in opens a web browser that allows single sign-on to the Citrix Receiver home page.
wiportalmode	<String>	Read-write	Layout on the Access Interface. The COMPACT value indicates the use of small icons. Possible values = NORMAL, COMPACT
clientchoices	<String>	Read-write	Provide users with multiple logon options. With client choices, users have the option of logging on by using the Citrix Gateway Plug-in for

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
			Windows, Citrix Gateway Plug-in for Java, the Web Interface, or clientless access from one location. Depending on how Citrix Gateway is configured, users are presented with up to three icons for logon choices. The most common are the Citrix Gateway Plug-in for Windows, Web Interface, and clientless access. Default value: OFF Possible values = ON, OFF
epaclienttype	<String>	Read-write	Choose between two types of End point Windows Client a) Application Agent - which always runs in the task bar as a standalone application and also has a supporting service which runs permanently when installed b) Activex Control - ActiveX control run

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
			by Microsoft Internet Explorer. Possible values = AGENT, PLUGIN
iipdnssuffix	<String>	Read-write	An intranet IP DNS suffix. When a user logs on to Citrix Gateway and is assigned an IP address, a DNS record for the user name and IP address combination is added to the Citrix Gateway DNS cache. You can configure a DNS suffix to append to the user name when the DNS record is added to the cache. You can reach to the host from where the user is logged on by using the user's name, which can be easier to remember than an IP address. When the user logs off from Citrix Gateway, the record is removed from the DNS cache. Minimum length = 1
forcedtimeout	<Double>	Read-write	Force a disconnection from the Citrix Gateway Plug-in with Citrix

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
			Gateway after a specified number of minutes. If the session closes, the user must log on again. Minimum value = 1 Maximum value = 65535
forcedtimeoutwarning	<Double>	Read-write	Number of minutes to warn a user before the user session is disconnected. Minimum value = 1 Maximum value = 255
ntdomain	<String>	Read-write	Single sign-on domain to use for single sign-on to applications in the internal network. This setting can be overwritten by the domain that users specify at the time of logon or by the domain that the authentication server returns. Minimum length = 1 Maximum length = 32
clientlessvpnmode	<String>	Read-write	Enable clientless access for web, XenApp or

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
			XenDesktop, and FileShare resources without installing the Citrix Gateway Plug-in. Available settings function as follows: * ON - Allow only clientless access. * OFF - Allow clientless access after users log on with the Citrix Gateway Plug-in. * DISABLED - Do not allow clientless access. Default value: OFF Possible values = ON, OFF, DISABLED
clientlessmodeurlencoding	<String>	Read-write	When clientless access is enabled, you can choose to encode the addresses of internal web applications or to leave the address as clear text. Available settings function as follows: * OPAQUE - Use standard encoding mechanisms to make the domain and protocol part of

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
			the resource unclear to users. * TRANSPARENT - Do not encode the web address and make it visible to users. * ENCRYPT - Allow the domain and protocol to be encrypted using a session key. When the web address is encrypted, the URL is different for each user session for the same web resource. If users bookmark the encoded web address, save it in the web browser and then log off, they cannot connect to the web address when they log on and use the bookmark. If users save the encrypted bookmark in the Access Interface during their session, the bookmark works each time the user logs on.

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
			Default value: OPAQUE Possible values = TRANSPARENT, OPAQUE, ENCRYPT
clientlesspersistentcookie	<String>	Read-write	State of persistent cookies in clientless access mode. Persistent cookies are required for accessing certain features of SharePoint, such as opening and editing Microsoft Word, Excel, and PowerPoint documents hosted on the SharePoint server. A persistent cookie remains on the user device and is sent with each HTTP request. Citrix Gateway encrypts the persistent cookie before sending it to the plug-in on the user device, and refreshes the cookie periodically as long as the session exists. The cookie becomes stale if the session ends. Available settings function as follows:

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
			* ALLOW - Enable persistent cookies. Users can open and edit Microsoft documents stored in SharePoint. * DENY - Disable persistent cookies. Users cannot open and edit Microsoft documents stored in SharePoint. * PROMPT - Prompt users to allow or deny persistent cookies during the session. Persistent cookies are not required for clientless access if users do not connect to SharePoint. Default value: DENY Possible values = ALLOW, DENY, PROMPT
emailhome	<String>	Read-write	Web address for the web-based email, such as Outlook Web Access.

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
allowedlogingroups	<String>	Read-write	Specify groups that have permission to log on to Citrix Gateway. Users who do not belong to this group or groups are denied access even if they have valid credentials. Minimum length = 1 Maximum length = 511
encryptcsecexp	<String>	Read-write	Enable encryption of client security expressions. Default value: ENABLED Possible values = ENABLED, DISABLED
apptokentimeout	<Double>	Read-write	The timeout value in seconds for tokens to access XenMobile applications. Default value: 100 Minimum value = 1 Maximum value = 255
mdxtokentimeout	<Double>	Read-write	Validity of MDX Token in minutes. This token is used for mdx services to access backend and

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
			valid HEAD and GET request. Default value: 10 Minimum value = 1 Maximum value = 1440
uitheme	<String>	Read-write	Set VPN UI Theme to Green-Bubble, Caxton or Custom; default is Caxton. Possible values = DEFAULT, GREENBUBBLE, CUSTOM
securebrowse	<String>	Read-write	Allow users to connect through Citrix Gateway to network resources from iOS and Android mobile devices with Citrix Receiver. Users do not need to establish a full VPN tunnel to access resources in the secure network. Default value: ENABLED Possible values = ENABLED, DISABLED
storefronturl	<String>	Read-write	Web address for StoreFront to be used in this session

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
			for enumeration of resources from XenApp or XenDesktop. Minimum length = 1 Maximum length = 255
kcdaccount	<String>	Read-write	The KCD account details to be used in SSO.
clientversions	<String>	Read-write	checkversion api. Minimum length = 1 Maximum length = 100
rdpclientprofilename	<String>	Read-write	Name of the RDP profile associated with the vserver. Minimum length = 1 Maximum length = 127
windowspluginupgrade	<String>	Read-write	Option to set plugin upgrade behaviour for Win. Default value: Always Possible values = Always, Essential, Never
macpluginupgrade	<String>	Read-write	Option to set plugin upgrade behaviour for Mac.

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
			Default value: Always Possible values = Always, Essential, Never
linuxpluginupgrade	<String>	Read-write	Option to set plugin upgrade behaviour for Linux. Default value: Always Possible values = Always, Essential, Never
iconwithreceiver	<String>	Read-write	Option to decide whether to show plugin icon along with receiver icon. Default value: OFF Possible values = ON, OFF
userdomains	<String>	Read-write	List of user domains specified as comma seperated value.
icasessiontimeout	<String>	Read-write	Enable or disable ica session timeout. If enabled and in case AAA session gets terminated, ICA connections associated with that will also get terminated.

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
			Default value: OFF Possible values = ON, OFF
alwaysonprofilename	<String>	Read-write	Name of the AlwaysON profile. The builtin profile named none can be used to explicitly disable AlwaysON. Minimum length = 1 Maximum length = 127
autoproxyurl	<String>	Read-write	URL to auto proxy config file.
advancedclientlessvpnmode	<String>	Read-write	Option to enable/disable Advanced ClientlessVpnMode. Default value: DISABLED Possible values = ENABLED, DISABLED
pcoipprofilename	<String>	Read-write	Name of the PCOIP profile. Minimum length = 1 Maximum length = 127
backendserversni	<String>	Read-write	enables sni extension for backend server handshakes.

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
			Default value: DISABLED Possible values = ENABLED, DISABLED
backendcertvalidation	<String>	Read-write	enables backend server certificate validation. Default value: DISABLED Possible values = ENABLED, DISABLED
fqdnspoofedip	<String>	Read-write	Spoofed IP address range that can be used by client for FQDN based split tunneling. Minimum length = 1
netmask	<String>	Read-write	The netmask for the spoofed ip address. Minimum length = 1
name	<String>	Read-only	The VPN name.
clientidletimeoutwarning	<Double>	Read-only	The time after which the client gets a timeout warning, in minutes.
vpnsessionpolicybindtype	<String>	Read-only	Indicates current bind type (Classic/ Advanced) for VPN session policy across all bind entities.

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
			Default value: Advanced Policy Possible values = Classic Policy, Advanced Policy
vpnsessionpolicycount	<Double>	Read-only	Count of VPN session policies across all bind entities.

Operations

(click to see Properties)

UPDATE| UNSET| GET (ALL)

Some options that you can use for each operations:

- **Getting warnings in response:**NITRO allows you to get warnings in an operation by specifying the "warning" query parameter as "yes". For example, to get warnings while connecting to the Citrix ADC appliance, the URL is as follows:

  http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/login?warning=yes

  If any, the warnings are displayed in the response payload with the HTTP code "209 X-NITRO-WARNING".

- **Authenticated access for individual NITRO operations:**NITRO allows you to logon to the Citrix ADC appliance to perform individual operations. You can use this option instead of creating a NITRO session (using the login object) and then using that session to perform all operations,

  To do this, you must specify the username and password in the request header of the NITRO request as follows:

  X-NITRO-USER:<username>

  X-NITRO-PASS:<password>

  Note: In such cases, make sure that the request header DOES not include the following:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue>

`*Note:` *

Mandatory parameters are marked in redand placeholder content is marked in <green>.

update

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/vpnparameter

HTTP Method:PUT

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Content-Type:application/json

Request Payload:

```
{"vpnparameter":{

"httpport":<Integer[]_value>,

"winsip":<String_value>,

"dnsvservername":<String_value>,

"splitdns":<String_value>,

"icauseraccounting":<String_value>,

"sesstimeout":<Double_value>,

"clientsecurity":<String_value>,

"clientsecuritygroup":<String_value>,

"clientsecuritymessage":<String_value>,

"clientsecuritylog":<String_value>,

"smartgroup":<String_value>,

"splittunnel":<String_value>,
```

```
"locallanaccess":<String_value>,

"rfc1918":<String_value>,

"spoofiip":<String_value>,

"killconnections":<String_value>,

"transparentinterception":<String_value>,

"windowsclienttype":<String_value>,

"defaultauthorizationaction":<String_value>,

"authorizationgroup":<String_value>,

"clientidletimeout":<Double_value>,

"proxy":<String_value>,

"allprotocolproxy":<String_value>,

"httpproxy":<String_value>,

"ftpproxy":<String_value>,

"socksproxy":<String_value>,

"gopherproxy":<String_value>,

"sslproxy":<String_value>,

"proxyexception":<String_value>,

"proxylocalbypass":<String_value>,

"clientcleanupprompt":<String_value>,

"forcecleanup":<String[]_value>,

"clientoptions":<String[]_value>,

"clientconfiguration":<String[]_value>,
```

```
"sso":<String_value>,

"ssocredential":<String_value>,

"windowsautologon":<String_value>,

"usemip":<String_value>,

"useiip":<String_value>,

"clientdebug":<String_value>,

"loginscript":<String_value>,

"logoutscript":<String_value>,

"homepage":<String_value>,

"icaproxy":<String_value>,

"wihome":<String_value>,

"wihomeaddresstype":<String_value>,

"citrixreceiverhome":<String_value>,

"wiportalmode":<String_value>,

"clientchoices":<String_value>,

"epaclienttype":<String_value>,

"iipdnssuffix":<String_value>,

"forcedtimeout":<Double_value>,

"forcedtimeoutwarning":<Double_value>,

"ntdomain":<String_value>,

"clientlessvpnmode":<String_value>,

"clientlessmodeurlencoding":<String_value>,
```

```
"clientlesspersistentcookie":<String_value>,

"emailhome":<String_value>,

"allowedlogingroups":<String_value>,

"encryptcsecexp":<String_value>,

"apptokentimeout":<Double_value>,

"mdxtokentimeout":<Double_value>,

"uitheme":<String_value>,

"securebrowse":<String_value>,

"storefronturl":<String_value>,

"kcdaccount":<String_value>,

"clientversions":<String_value>,

"rdpclientprofilename":<String_value>,

"windowspluginupgrade":<String_value>,

"macpluginupgrade":<String_value>,

"linuxpluginupgrade":<String_value>,

"iconwithreceiver":<String_value>,

"userdomains":<String_value>,

"icasessiontimeout":<String_value>,

"alwaysonprofilename":<String_value>,

"autoproxyurl":<String_value>,

"advancedclientlessvpnmode":<String_value>,

"pcoipprofilename":<String_value>,
```

```
"backendserversni":<String_value>,

"backendcertvalidation":<String_value>,

"fqdnspoofedip":<String_value>,

"netmask":<String_value>

}}
```

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error

unset

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/vpnparameter?**action=unset**

HTTP Method:POST

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Content-Type:application/json

Request Payload:

```
{"vpnparameter":{

"httpport":true,

"winsip":true,

"dnsvservername":true,

"splitdns":true,

"icauseraccounting":true,
```

```
"sesstimeout":true,
"clientsecurity":true,
"clientsecuritygroup":true,
"smartgroup":true,
"clientsecuritymessage":true,
"clientsecuritylog":true,
"authorizationgroup":true,
"clientidletimeout":true,
"allprotocolproxy":true,
"httpproxy":true,
"ftpproxy":true,
"socksproxy":true,
"gopherproxy":true,
"sslproxy":true,
"proxyexception":true,
"forcecleanup":true,
"clientoptions":true,
"clientconfiguration":true,
"loginscript":true,
"logoutscript":true,
"homepage":true,
"proxy":true,
```

```
"wihome":true,

"citrixreceiverhome":true,

"wiportalmode":true,

"iipdnssuffix":true,

"forcedtimeout":true,

"forcedtimeoutwarning":true,

"defaultauthorizationaction":true,

"ntdomain":true,

"clientlessvpnmode":true,

"emailhome":true,

"clientlessmodeurlencoding":true,

"clientlesspersistentcookie":true,

"allowedlogingroups":true,

"apptokentimeout":true,

"mdxtokentimeout":true,

"storefronturl":true,

"uitheme":true,

"kcdaccount":true,

"rdpclientprofilename":true,

"windowspluginupgrade":true,

"macpluginupgrade":true,

"linuxpluginupgrade":true,
```

```
"iconwithreceiver":true,

"userdomains":true,

"alwaysonprofilename":true,

"autoproxyurl":true,

"pcoipprofilename":true,

"advancedclientlessvpnmode":true,

"fqdnspoofedip":true,

"splittunnel":true,

"locallanaccess":true,

"rfc1918":true,

"spoofiip":true,

"killconnections":true,

"transparentinterception":true,

"windowsclienttype":true,

"proxylocalbypass":true,

"clientcleanupprompt":true,

"sso":true,

"ssocredential":true,

"windowsautologon":true,

"usemip":true,

"useiip":true,

"clientdebug":true,
```

```
"icaproxy":true,

"clientchoices":true,

"epaclienttype":true,

"encryptcsecexp":true,

"securebrowse":true,

"clientversions":true,

"icasessiontimeout":true,

"backendserversni":true,

"backendcertvalidation":true,

"netmask":true

}}
```

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error

get (all)

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/vpnparameter

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "vpnparameter": [ {

"name":<String_value>,

"httpport":<Integer[]_value>,

"winsip":<String_value>,

"dnsvservername":<String_value>,

"splitdns":<String_value>,

"icauseraccounting":<String_value>,

"sesstimeout":<Double_value>,

"clientsecurity":<String_value>,

"clientsecuritygroup":<String_value>,

"clientsecuritymessage":<String_value>,

"clientsecuritylog":<String_value>,

"smartgroup":<String_value>,

"splittunnel":<String_value>,

"locallanaccess":<String_value>,

"rfc1918":<String_value>,

"spoofiip":<String_value>,
```

```
"killconnections":<String_value>,

"transparentinterception":<String_value>,

"windowsclienttype":<String_value>,

"defaultauthorizationaction":<String_value>,

"authorizationgroup":<String_value>,

"clientidletimeout":<Double_value>,

"clientidletimeoutwarning":<Double_value>,

"proxy":<String_value>,

"allprotocolproxy":<String_value>,

"httpproxy":<String_value>,

"ftpproxy":<String_value>,

"socksproxy":<String_value>,

"gopherproxy":<String_value>,

"sslproxy":<String_value>,

"proxyexception":<String_value>,

"proxylocalbypass":<String_value>,

"clientcleanupprompt":<String_value>,

"forcecleanup":<String[]_value>,

"clientoptions":<String[]_value>,

"clientconfiguration":<String[]_value>,

"sso":<String_value>,

"ssocredential":<String_value>,
```

```
"windowsautologon":<String_value>,

"usemip":<String_value>,

"useiip":<String_value>,

"clientdebug":<String_value>,

"loginscript":<String_value>,

"logoutscript":<String_value>,

"homepage":<String_value>,

"icaproxy":<String_value>,

"wihome":<String_value>,

"wihomeaddresstype":<String_value>,

"citrixreceiverhome":<String_value>,

"wiportalmode":<String_value>,

"clientchoices":<String_value>,

"epaclienttype":<String_value>,

"iipdnssuffix":<String_value>,

"forcedtimeout":<Double_value>,

"forcedtimeoutwarning":<Double_value>,

"ntdomain":<String_value>,

"clientlessvpnmode":<String_value>,

"clientlessmodeurlencoding":<String_value>,

"clientlesspersistentcookie":<String_value>,

"emailhome":<String_value>,
```

```
"allowedlogingroups":<String_value>,

"encryptcsecexp":<String_value>,

"apptokentimeout":<Double_value>,

"mdxtokentimeout":<Double_value>,

"uitheme":<String_value>,

"securebrowse":<String_value>,

"storefronturl":<String_value>,

"kcdaccount":<String_value>,

"clientversions":<String_value>,

"rdpclientprofilename":<String_value>,

"windowspluginupgrade":<String_value>,

"macpluginupgrade":<String_value>,

"linuxpluginupgrade":<String_value>,

"iconwithreceiver":<String_value>,

"userdomains":<String_value>,

"icasessiontimeout":<String_value>,

"alwaysonprofilename":<String_value>,

"autoproxyurl":<String_value>,

"advancedclientlessvpnmode":<String_value>,

"pcoipprofilename":<String_value>,

"backendserversni":<String_value>,

"backendcertvalidation":<String_value>,
```

```
"vpnsessionpolicybindtype":<String_value>,

"vpnsessionpolicycount":<Double_value>,

"fqdnspoofedip":<String_value>,

"netmask":<String_value>

}]}
```

vpnpcoipconnection

Configuration for PCoIP connection resource.

Properties

(click to see Operations)

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
username	<String>	Read-write	User name for the PCOIP connections. Minimum length = 1
nodeid	<Double>	Read-write	Unique number that identifies the cluster node. Minimum value = 0 Maximum value = 31
all	<Boolean>	Read-write	All active pcoip connections.
srcip	<String>	Read-only	The client IP address.
srcport	<Integer>	Read-only	The client port. Range 1 - 65535

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
			* in CLI is represented as 65535 in NITRO API
destip	<String>	Read-only	The VDA IP address.
destport	<Integer>	Read-only	The VDA server port. Range 1 - 65535 * in CLI is represented as 65535 in NITRO API
peid	<Double>	Read-only	Core id of the session owner.
__count	<Double>	Read-only	count parameter

Operations

(click to see Properties)

KILL| GET (ALL)| COUNT

Some options that you can use for each operations:

- **Getting warnings in response:**NITRO allows you to get warnings in an operation by specifying the "warning" query parameter as "yes". For example, to get warnings while connecting to the Citrix ADC appliance, the URL is as follows:

  http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/login?warning=yes

  If any, the warnings are displayed in the response payload with the HTTP code "209 X-NITRO-WARNING".

- **Authenticated access for individual NITRO operations:**NITRO allows you to logon to the Citrix ADC appliance to perform individual operations. You can use this option instead of creating a NITRO session (using the login object) and then using that session to perform all operations,

To do this, you must specify the username and password in the request header of the NITRO request as follows:

X-NITRO-USER:<username>

X-NITRO-PASS:<password>

Note: In such cases, make sure that the request header DOES not include the following:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue>

`*Note:`*

Mandatory parameters are marked in redand placeholder content is marked in <green>.

kill

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/vpnpcoipconnection?**action=kill**

HTTP Method:POST

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Content-Type:application/json

Request Payload:

```
{"vpnpcoipconnection":{

"username":<String_value>,

"all":<Boolean_value>

}}
```

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error

get (all)

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/vpnpcoipconnection

Query-parameters:

args

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/vpnpcoipconnection?**args=username:<String_value>,nodeid:<Double_value>**

Use this query-parameter to get vpnpcoipconnection resources based on additional properties.

attrs

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/vpnpcoipconnection?**attrs=property-name1,property-name2**

Use this query parameter to specify the resource details that you want to retrieve.

filter

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/vpnpcoipconnection?**filter=property-name1:property-val1,property-name2:property-val2**

Use this query-parameter to get the filtered set of vpnpcoipconnection resources configured on Citrix ADC. Filtering can be done on any of the properties of the resource.

view

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/vpnpcoipconnection?**view=summary**

Note: By default, the retrieved results are displayed in detail view (?view=detail).

pagination

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/vpnpcoipconnection?**pagesize=#no;pageno=#no**

Use this query-parameter to get the vpnpcoipconnection resources in chunks.

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "vpnpcoipconnection": [ {

username:<String_value>,nodeid:<Double_value>"srcip":<String_value>,

"srcport":<Integer_value>,

"destip":<String_value>,

"destport":<Integer_value>,

"peid":<Double_value>

}]}
```

count

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/vpnpcoipconnection?**count=yes**

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "vpnpcoipconnection": [ { "__count": "#no"} ] }
```

vpnpcoipprofile

Configuration for PCoIP session profile resource.

Properties

(click to see Operations)

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
name	<String>	Read-write	name of PCoIP profile. Minimum length = 1
conserverurl	<String>	Read-write	Connection server URL.
icvverification	<String>	Read-write	ICV verification for PCOIP transport packets. Default value: DISABLED Possible values = ENABLED, DISABLED
sessionidletimeout	<Double>	Read-write	PCOIP Idle Session timeout. Default value: 180 Minimum value = 30 Maximum value = 240

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
__count	<Double>	Read-only	count parameter

Operations

(click to see Properties)

ADD| DELETE| UPDATE| UNSET| GET (ALL)| GET| COUNT

Some options that you can use for each operations:

- **Getting warnings in response:**NITRO allows you to get warnings in an operation by specifying the "warning" query parameter as "yes". For example, to get warnings while connecting to the Citrix ADC appliance, the URL is as follows:

  http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/login?warning=yes

  If any, the warnings are displayed in the response payload with the HTTP code "209 X-NITRO-WARNING".

- **Authenticated access for individual NITRO operations:**NITRO allows you to logon to the Citrix ADC appliance to perform individual operations. You can use this option instead of creating a NITRO session (using the login object) and then using that session to perform all operations,

  To do this, you must specify the username and password in the request header of the NITRO request as follows:

  X-NITRO-USER:<username>

  X-NITRO-PASS:<password>

  Note: In such cases, make sure that the request header DOES not include the following:

  Cookie:NITRO_AUTH_TOKEN=<tokenvalue>

`*Note:`*

Mandatory parameters are marked in redand placeholder content is marked in <green>.

add

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/vpnpcoipprofile

HTTP Method:POST

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Content-Type:application/json

Request Payload:

```
{"vpnpcoipprofile":{

<b>"name":<String_value>,

</b><b>"conserverurl":<String_value>,

</b>"icvverification":<String_value>,

"sessionidletimeout":<Double_value>

}}
```

Response:

HTTP Status Code on Success: 201 Created HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error

delete

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/vpnpcoipprofile/name_value<String>

HTTP Method:DELETE

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue>

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error

update

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/vpnpcoipprofile

HTTP Method:PUT

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Content-Type:application/json

Request Payload:

```
{"vpnpcoipprofile":{

<b>"name":<String_value>,

</b>"conserverurl":<String_value>,

"icvverification":<String_value>,

"sessionidletimeout":<Double_value>

}}
```

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error

unset

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/vpnpcoipprofile?**action=unset**

HTTP Method:POST

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Content-Type:application/json

Request Payload:

```
{"vpnpcoipprofile":{

<b>"name":<String_value>,

</b>"icvverification":true,

"sessionidletimeout":true

}}
```

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error

get (all)

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/vpnpcoipprofile

Query-parameters:

attrs

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/vpnpcoipprofile?**attrs=property-name1,property-name2**

Use this query parameter to specify the resource details that you want to retrieve.

filter

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/vpnpcoipprofile?**filter=property-name1:property-val1,property-name2:property-val2**

Use this query-parameter to get the filtered set of vpnpcoipprofile resources configured on Citrix ADC. Filtering can be done on any of the properties of the resource.

view

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/vpnpcoipprofile?**view=summary**

Note: By default, the retrieved results are displayed in detail view (?view=detail).

pagination

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/vpnpcoipprofile?**pagesize=#no;pageno=#no**

Use this query-parameter to get the vpnpcoipprofile resources in chunks.

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "vpnpcoipprofile": [ {

"name":<String_value>,

"conserverurl":<String_value>,

"icvverification":<String_value>,

"sessionidletimeout":<Double_value>

}]}
```

get

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/vpnpcoipprofile/name_value<String>

Query-parameters:

attrs

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/vpnpcoipprofile/name_value<String>?**attrs=property-name1,property-name2**

Use this query parameter to specify the resource details that you want to retrieve.

view

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/vpnpcoipprofile/name_value<String>?**view=summary**

Note: By default, the retrieved results are displayed in detail view (?view=detail).

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "vpnpcoipprofile": [ {

"name":<String_value>,

"conserverurl":<String_value>,

"icvverification":<String_value>,

"sessionidletimeout":<Double_value>

}]}
```

count

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/vpnpcoipprofile?**count=yes**

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "vpnpcoipprofile": [ { "__count": "#no"} ] }
```

vpnpcoipvserverprofile

Configuration for PCoIP vserver profile resource.

Properties

(click to see Operations)

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
name	<String>	Read-write	name of PCoIP vserver profile. Minimum length = 1
logindomain	<String>	Read-write	Login domain for PCoIP users.

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
udpport	<Integer>	Read-write	UDP port for PCoIP data traffic. Default value: 4172 Range 1 - 65535 * in CLI is represented as 65535 in NITRO API
__count	<Double>	Read-only	count parameter

Operations

(click to see Properties)

ADD| DELETE| UPDATE| UNSET| GET (ALL)| GET| COUNT

Some options that you can use for each operations:

- **Getting warnings in response:**NITRO allows you to get warnings in an operation by specifying the "warning" query parameter as "yes". For example, to get warnings while connecting to the Citrix ADC appliance, the URL is as follows:

  http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/login?warning=yes

  If any, the warnings are displayed in the response payload with the HTTP code "209 X-NITRO-WARNING".

- **Authenticated access for individual NITRO operations:**NITRO allows you to logon to the Citrix ADC appliance to perform individual operations. You can use this option instead of creating a NITRO session (using the login object) and then using that session to perform all operations,

  To do this, you must specify the username and password in the request header of the NITRO request as follows:

  X-NITRO-USER:<username>

  X-NITRO-PASS:<password>

Note: In such cases, make sure that the request header DOES not include the following:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue>

`*Note:` *

Mandatory parameters are marked in redand placeholder content is marked in <green>.

add

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/vpnpcoipvserverprofile

HTTP Method:POST

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Content-Type:application/json

Request Payload:

```
{"vpnpcoipvserverprofile":{

<b>"name":<String_value>,

</b><b>"logindomain":<String_value>,

</b>"udpport":<Integer_value>

}}
```

Response:

HTTP Status Code on Success: 201 Created HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error

delete

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/vpnpcoipvserverprofile/name_value<String>

HTTP Method:DELETE

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue>

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error

update

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/vpnpcoipvserverprofile

HTTP Method:PUT

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Content-Type:application/json

Request Payload:

```
{"vpnpcoipvserverprofile":{

<b>"name":<String_value>,

</b>"udpport":<Integer_value>,

"logindomain":<String_value>

}}
```

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error

unset

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/vpnpcoipvserverprofile?**action=unset**

HTTP Method:POST

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Content-Type:application/json

Request Payload:

```
{"vpnpcoipvserverprofile":{

<b>"name":<String_value>,

</b>"udpport":true

}}
```

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error

get (all)

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/vpnpcoipvserverprofile

Query-parameters:

attrs

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/vpnpcoipvserverprofile?**attrs=property-name1,property-name2**

Use this query parameter to specify the resource details that you want to retrieve.

filter

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/vpnpcoipvserverprofile?**filter=property-name1:property-val1,property-name2:property-val2**

Use this query-parameter to get the filtered set of vpnpcoipvserverprofile resources configured on Citrix ADC. Filtering can be done on any of the properties of the resource.

view

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/vpnpcoipvserverprofile?**view=summary**

Note: By default, the retrieved results are displayed in detail view (?view=detail).

pagination

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/vpnpcoipvserverprofile?**pagesize=#no;pageno=#no**

Use this query-parameter to get the vpnpcoipvserverprofile resources in chunks.

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "vpnpcoipvserverprofile": [ {

"name":<String_value>,

"udpport":<Integer_value>,

"logindomain":<String_value>

}]}
```

get

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/vpnpcoipvserverprofile/name_value<String>

Query-parameters:

attrs

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/vpnpcoipvserverprofile/name_value<String>?**attrs=property-name1,property-name2**

Use this query parameter to specify the resource details that you want to retrieve.

view

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/vpnpcoipvserverprofile/name_value<String>?**view=summary**

Note: By default, the retrieved results are displayed in detail view (?view=detail).

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "vpnpcoipvserverprofile": [ {

"name":<String_value>,
```

```
"udpport":<Integer_value>,

"logindomain":<String_value>

}]}
```

count

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/vpnpcoipvserverprofile?**count=yes**

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "vpnpcoipvserverprofile": [ { "__count": "#no"} ] }
```

vpnportaltheme

Configuration for portaltheme resource.

Properties

(click to see Operations)

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
name	<String>	Read-write	Name of the uitheme. Minimum length = 1
basetheme	<String>	Read-write	. Minimum length = 1 Possible values = Default, Greenbubble, X1, RfWebUI
__count	<Double>	Read-only	count parameter

Operations

(click to see Properties)

ADD| DELETE| GET (ALL)| GET| COUNT

Some options that you can use for each operations:

- **Getting warnings in response:**NITRO allows you to get warnings in an operation by specifying the "warning" query parameter as "yes". For example, to get warnings while connecting to the Citrix ADC appliance, the URL is as follows:

  http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/login?warning=yes

  If any, the warnings are displayed in the response payload with the HTTP code "209 X-NITRO-WARNING".

- **Authenticated access for individual NITRO operations:**NITRO allows you to logon to the Citrix ADC appliance to perform individual operations. You can use this option instead of creating a NITRO session (using the login object) and then using that session to perform all operations,

  To do this, you must specify the username and password in the request header of the NITRO request as follows:

  X-NITRO-USER:<username>

X-NITRO-PASS:<password>

Note: In such cases, make sure that the request header DOES not include the following:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue>

`*Note:`*

Mandatory parameters are marked in redand placeholder content is marked in <green>.

add

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/vpnportaltheme

HTTP Method:POST

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Content-Type:application/json

Request Payload:

```
{"vpnportaltheme":{

<b>"name":<String_value>,

</b><b>"basetheme":<String_value>

</b>}}
```

Response:

HTTP Status Code on Success: 201 Created HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error

delete

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/vpnportaltheme/name_value<String>

HTTP Method:DELETE

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue>

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error

get (all)

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/vpnportaltheme

Query-parameters:

attrs

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/vpnportaltheme?**attrs=property-name1,property-name2**

Use this query parameter to specify the resource details that you want to retrieve.

filter

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/vpnportaltheme?**filter=property-name1:property-val1,property-name2:property-val2**

Use this query-parameter to get the filtered set of vpnportaltheme resources configured on Citrix ADC. Filtering can be done on any of the properties of the resource.

view

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/vpnportaltheme?**view=summary**

Note: By default, the retrieved results are displayed in detail view (?view=detail).

pagination

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/vpnportaltheme?**pagesize=#no;pageno=#no**

Use this query-parameter to get the vpnportaltheme resources in chunks.

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "vpnportaltheme": [ {

"name":<String_value>,

"basetheme":<String_value>

}]}
```

get

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/vpnportaltheme/name_value<String>

Query-parameters:

attrs

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/vpnportaltheme/name_value<String>?**attrs=property-name1,property-name2**

Use this query parameter to specify the resource details that you want to retrieve.

view

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/vpnportaltheme/name_value<String>?**view=summary**

Note: By default, the retrieved results are displayed in detail view (?view=detail).

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "vpnportaltheme": [ {

"name":<String_value>,

"basetheme":<String_value>

}]}
```

count

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/vpnportaltheme?**count=yes**

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "vpnportaltheme": [ { "__count": "#no"} ] }
```

vpnsamlssoprofile

Configuration for SAML sso action resource.

Properties

(click to see Operations)

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
name	<String>	Read-write	Name for the new saml single sign-on profile. Must begin with an ASCII alphanumeric or underscore (_) character, and must contain only ASCII alphanumeric, underscore, hash (#), period (.), space, colon (:), at (@), equals (=), and hyphen (-) characters. Cannot be changed after an SSO action is created. The following requirement applies only to the Citrix ADC CLI: If the name includes one or more spaces, enclose the name in double or

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
			single quotation marks (for example, "my action" or 'my action'). Minimum length = 1
samlsigningcertname	<String>	Read-write	Name of the signing authority as given in the SAML server's SSL certificate. Minimum length = 1
assertionconsumerserviceurl	<String>	Read-write	URL to which the assertion is to be sent. Minimum length = 1
relaystaterule	<String>	Read-write	Expression to extract relaystate to be sent along with assertion. Evaluation of this expression should return TEXT content. This is typically a target url to which user is redirected after the recipient validates SAML token.
sendpassword	<String>	Read-write	Option to send password in assertion. Default value: OFF Possible values = ON, OFF

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
samlissuername	<String>	Read-write	The name to be used in requests sent from Citrix ADC to IdP to uniquely identify Citrix ADC. Minimum length = 1
signaturealg	<String>	Read-write	Algorithm to be used to sign/verify SAML transactions. Default value: RSA-SHA256 Possible values = RSA-SHA1, RSA-SHA256
digestmethod	<String>	Read-write	Algorithm to be used to compute/verify digest for SAML transactions. Default value: SHA256 Possible values = SHA1, SHA256
audience	<String>	Read-write	Audience for which assertion sent by IdP is applicable. This is typically entity name or url that represents ServiceProvider.
nameidformat	<String>	Read-write	Format of Name Identifier sent in Assertion. Default value: transient

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
			Possible values = Unspecified, emailAddress, X509SubjectName, WindowsDomainQualifiedName, kerberos, entity, persistent, transient
nameidexpr	<String>	Read-write	Expression that will be evaluated to obtain NameIdentifier to be sent in assertion. Maximum length = 128
attribute1	<String>	Read-write	Name of attribute1 that needs to be sent in SAML Assertion.
attribute1expr	<String>	Read-write	Expression that will be evaluated to obtain attribute1's value to be sent in Assertion. Maximum length = 128
attribute1friendlyname	<String>	Read-write	User-Friendly Name of attribute1 that needs to be sent in SAML Assertion.
attribute1format	<String>	Read-write	Format of Attribute1 to be sent in Assertion. Possible values = URI, Basic

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
attribute2	<String>	Read-write	Name of attribute2 that needs to be sent in SAML Assertion.
attribute2expr	<String>	Read-write	Expression that will be evaluated to obtain attribute2's value to be sent in Assertion. Maximum length = 128
attribute2friendlyname	<String>	Read-write	User-Friendly Name of attribute2 that needs to be sent in SAML Assertion.
attribute2format	<String>	Read-write	Format of Attribute2 to be sent in Assertion. Possible values = URI, Basic
attribute3	<String>	Read-write	Name of attribute3 that needs to be sent in SAML Assertion.
attribute3expr	<String>	Read-write	Expression that will be evaluated to obtain attribute3's value to be sent in Assertion. Maximum length = 128
attribute3friendlyname	<String>	Read-write	User-Friendly Name of attribute3 that needs to be sent in SAML Assertion.

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
attribute3format	<String>	Read-write	Format of Attribute3 to be sent in Assertion. Possible values = URI, Basic
attribute4	<String>	Read-write	Name of attribute4 that needs to be sent in SAML Assertion.
attribute4expr	<String>	Read-write	Expression that will be evaluated to obtain attribute4's value to be sent in Assertion. Maximum length = 128
attribute4friendlyname	<String>	Read-write	User-Friendly Name of attribute4 that needs to be sent in SAML Assertion.
attribute4format	<String>	Read-write	Format of Attribute4 to be sent in Assertion. Possible values = URI, Basic
attribute5	<String>	Read-write	Name of attribute5 that needs to be sent in SAML Assertion.
attribute5expr	<String>	Read-write	Expression that will be evaluated to obtain attribute5's value to be sent in Assertion. Maximum length = 128

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
attribute5friendlyname	<String>	Read-write	User-Friendly Name of attribute5 that needs to be sent in SAML Assertion.
attribute5format	<String>	Read-write	Format of Attribute5 to be sent in Assertion. Possible values = URI, Basic
attribute6	<String>	Read-write	Name of attribute6 that needs to be sent in SAML Assertion.
attribute6expr	<String>	Read-write	Expression that will be evaluated to obtain attribute6's value to be sent in Assertion. Maximum length = 128
attribute6friendlyname	<String>	Read-write	User-Friendly Name of attribute6 that needs to be sent in SAML Assertion.
attribute6format	<String>	Read-write	Format of Attribute6 to be sent in Assertion. Possible values = URI, Basic
attribute7	<String>	Read-write	Name of attribute7 that needs to be sent in SAML Assertion.
attribute7expr	<String>	Read-write	Expression that will be evaluated to

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
			obtain attribute7's value to be sent in Assertion. Maximum length = 128
attribute7friendlyname	<String>	Read-write	User-Friendly Name of attribute7 that needs to be sent in SAML Assertion.
attribute7format	<String>	Read-write	Format of Attribute7 to be sent in Assertion. Possible values = URI, Basic
attribute8	<String>	Read-write	Name of attribute8 that needs to be sent in SAML Assertion.
attribute8expr	<String>	Read-write	Expression that will be evaluated to obtain attribute8's value to be sent in Assertion. Maximum length = 128
attribute8friendlyname	<String>	Read-write	User-Friendly Name of attribute8 that needs to be sent in SAML Assertion.
attribute8format	<String>	Read-write	Format of Attribute8 to be sent in Assertion. Possible values = URI, Basic

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
attribute9	<String>	Read-write	Name of attribute9 that needs to be sent in SAML Assertion.
attribute9expr	<String>	Read-write	Expression that will be evaluated to obtain attribute9's value to be sent in Assertion. Maximum length = 128
attribute9friendlyname	<String>	Read-write	User-Friendly Name of attribute9 that needs to be sent in SAML Assertion.
attribute9format	<String>	Read-write	Format of Attribute9 to be sent in Assertion. Possible values = URI, Basic
attribute10	<String>	Read-write	Name of attribute10 that needs to be sent in SAML Assertion.
attribute10expr	<String>	Read-write	Expression that will be evaluated to obtain attribute10's value to be sent in Assertion. Maximum length = 128
attribute10friendlyname	<String>	Read-write	User-Friendly Name of attribute10 that needs to be sent in SAML Assertion.

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
attribute10format	<String>	Read-write	Format of Attribute10 to be sent in Assertion. Possible values = URI, Basic
attribute11	<String>	Read-write	Name of attribute11 that needs to be sent in SAML Assertion.
attribute11expr	<String>	Read-write	Expression that will be evaluated to obtain attribute11's value to be sent in Assertion. Maximum length = 128
attribute11friendlyname	<String>	Read-write	User-Friendly Name of attribute11 that needs to be sent in SAML Assertion.
attribute11format	<String>	Read-write	Format of Attribute11 to be sent in Assertion. Possible values = URI, Basic
attribute12	<String>	Read-write	Name of attribute12 that needs to be sent in SAML Assertion.
attribute12expr	<String>	Read-write	Expression that will be evaluated to obtain attribute12's value to be sent in Assertion. Maximum length = 128

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
attribute12friendlyname	<String>	Read-write	User-Friendly Name of attribute12 that needs to be sent in SAML Assertion.
attribute12format	<String>	Read-write	Format of Attribute12 to be sent in Assertion. Possible values = URI, Basic
attribute13	<String>	Read-write	Name of attribute13 that needs to be sent in SAML Assertion.
attribute13expr	<String>	Read-write	Expression that will be evaluated to obtain attribute13's value to be sent in Assertion. Maximum length = 128
attribute13friendlyname	<String>	Read-write	User-Friendly Name of attribute13 that needs to be sent in SAML Assertion.
attribute13format	<String>	Read-write	Format of Attribute13 to be sent in Assertion. Possible values = URI, Basic
attribute14	<String>	Read-write	Name of attribute14 that needs to be sent in SAML Assertion.
attribute14expr	<String>	Read-write	Expression that will be evaluated to

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
			obtain attribute14's value to be sent in Assertion. Maximum length = 128
attribute14friendlyname	<String>	Read-write	User-Friendly Name of attribute14 that needs to be sent in SAML Assertion.
attribute14format	<String>	Read-write	Format of Attribute14 to be sent in Assertion. Possible values = URI, Basic
attribute15	<String>	Read-write	Name of attribute15 that needs to be sent in SAML Assertion.
attribute15expr	<String>	Read-write	Expression that will be evaluated to obtain attribute15's value to be sent in Assertion. Maximum length = 128
attribute15friendlyname	<String>	Read-write	User-Friendly Name of attribute15 that needs to be sent in SAML Assertion.
attribute15format	<String>	Read-write	Format of Attribute15 to be sent in Assertion. Possible values = URI, Basic

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
attribute16	<String>	Read-write	Name of attribute16 that needs to be sent in SAML Assertion.
attribute16expr	<String>	Read-write	Expression that will be evaluated to obtain attribute16's value to be sent in Assertion. Maximum length = 128
attribute16friendlyname	<String>	Read-write	User-Friendly Name of attribute16 that needs to be sent in SAML Assertion.
attribute16format	<String>	Read-write	Format of Attribute16 to be sent in Assertion. Possible values = URI, Basic
encryptassertion	<String>	Read-write	Option to encrypt assertion when Citrix ADC sends one. Default value: OFF Possible values = ON, OFF
samlspcertname	<String>	Read-write	Name of the SSL certificate of peer/ receving party using which Assertion is encrypted. Minimum length = 1

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
encryptionalgorithm	<String>	Read-write	Algorithm to be used to encrypt SAML assertion. Default value: AES256 Possible values = DES3, AES128, AES192, AES256
skewtime	<Double>	Read-write	This option specifies the number of minutes on either side of current time that the assertion would be valid. For example, if skewTime is 10, then assertion would be valid from (current time - 10) min to (current time + 10) min, ie 20min in all. Default value: 5
signassertion	<String>	Read-write	Option to sign portions of assertion when Citrix ADC IDP sends one. Based on the user selection, either Assertion or Response or Both or none can be signed. Default value: ASSERTION Possible values = NONE,

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
			ASSERTION, RESPONSE, BOTH
signatureservice	<String>	Read-write	Name of the service in cloud used to sign the data. Minimum length = 1
__count	<Double>	Read-only	count parameter

Operations

(click to see Properties)

ADD| DELETE| UPDATE| UNSET| GET (ALL)| GET| COUNT

Some options that you can use for each operations:

- **Getting warnings in response:**NITRO allows you to get warnings in an operation by specifying the "warning" query parameter as "yes". For example, to get warnings while connecting to the Citrix ADC appliance, the URL is as follows:

  http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/login?warning=yes

  If any, the warnings are displayed in the response payload with the HTTP code "209 X-NITRO-WARNING".

- **Authenticated access for individual NITRO operations:**NITRO allows you to logon to the Citrix ADC appliance to perform individual operations. You can use this option instead of creating a NITRO session (using the login object) and then using that session to perform all operations,

  To do this, you must specify the username and password in the request header of the NITRO request as follows:

  X-NITRO-USER:<username>

  X-NITRO-PASS:<password>

  Note: In such cases, make sure that the request header DOES not include the following:

  Cookie:NITRO_AUTH_TOKEN=<tokenvalue>

`*Note:` *

Mandatory parameters are marked in redand placeholder content is marked in <green>.

add

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/vpnsamlssoprofile

HTTP Method:POST

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Content-Type:application/json

Request Payload:

```
{"vpnsamlssoprofile":{

<b>"name":<String_value>,

</b>"samlsigningcertname":<String_value>,

<b>"assertionconsumerserviceurl":<String_value>,

</b>"relaystaterule":<String_value>,

"sendpassword":<String_value>,

"samlissuername":<String_value>,

"signaturealg":<String_value>,

"digestmethod":<String_value>,

"audience":<String_value>,

"nameidformat":<String_value>,

"nameidexpr":<String_value>,

"attribute1":<String_value>,

"attribute1expr":<String_value>,
```

```
"attribute1friendlyname":<String_value>,

"attribute1format":<String_value>,

"attribute2":<String_value>,

"attribute2expr":<String_value>,

"attribute2friendlyname":<String_value>,

"attribute2format":<String_value>,

"attribute3":<String_value>,

"attribute3expr":<String_value>,

"attribute3friendlyname":<String_value>,

"attribute3format":<String_value>,

"attribute4":<String_value>,

"attribute4expr":<String_value>,

"attribute4friendlyname":<String_value>,

"attribute4format":<String_value>,

"attribute5":<String_value>,

"attribute5expr":<String_value>,

"attribute5friendlyname":<String_value>,

"attribute5format":<String_value>,

"attribute6":<String_value>,

"attribute6expr":<String_value>,

"attribute6friendlyname":<String_value>,

"attribute6format":<String_value>,
```

```
"attribute7":<String_value>,

"attribute7expr":<String_value>,

"attribute7friendlyname":<String_value>,

"attribute7format":<String_value>,

"attribute8":<String_value>,

"attribute8expr":<String_value>,

"attribute8friendlyname":<String_value>,

"attribute8format":<String_value>,

"attribute9":<String_value>,

"attribute9expr":<String_value>,

"attribute9friendlyname":<String_value>,

"attribute9format":<String_value>,

"attribute10":<String_value>,

"attribute10expr":<String_value>,

"attribute10friendlyname":<String_value>,

"attribute10format":<String_value>,

"attribute11":<String_value>,

"attribute11expr":<String_value>,

"attribute11friendlyname":<String_value>,

"attribute11format":<String_value>,

"attribute12":<String_value>,

"attribute12expr":<String_value>,
```

```
"attribute12friendlyname":<String_value>,

"attribute12format":<String_value>,

"attribute13":<String_value>,

"attribute13expr":<String_value>,

"attribute13friendlyname":<String_value>,

"attribute13format":<String_value>,

"attribute14":<String_value>,

"attribute14expr":<String_value>,

"attribute14friendlyname":<String_value>,

"attribute14format":<String_value>,

"attribute15":<String_value>,

"attribute15expr":<String_value>,

"attribute15friendlyname":<String_value>,

"attribute15format":<String_value>,

"attribute16":<String_value>,

"attribute16expr":<String_value>,

"attribute16friendlyname":<String_value>,

"attribute16format":<String_value>,

"encryptassertion":<String_value>,

"samlspcertname":<String_value>,

"encryptionalgorithm":<String_value>,

"skewtime":<Double_value>,
```

```
"signassertion":<String_value>,

"signatureservice":<String_value>

}}
```

Response:

HTTP Status Code on Success: 201 Created HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error

delete

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/vpnsamlssoprofile/name_value<String>

HTTP Method:DELETE

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue>

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error

update

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/vpnsamlssoprofile

HTTP Method:PUT

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Content-Type:application/json

Request Payload:

```
{"vpnsamlssoprofile":{
```

```
<b>"name":<String_value>,

</b>"samlsigningcertname":<String_value>,

"assertionconsumerserviceurl":<String_value>,

"sendpassword":<String_value>,

"samlissuername":<String_value>,

"relaystaterule":<String_value>,

"signaturealg":<String_value>,

"digestmethod":<String_value>,

"audience":<String_value>,

"nameidformat":<String_value>,

"nameidexpr":<String_value>,

"attribute1":<String_value>,

"attribute1expr":<String_value>,

"attribute1friendlyname":<String_value>,

"attribute1format":<String_value>,

"attribute2":<String_value>,

"attribute2expr":<String_value>,

"attribute2friendlyname":<String_value>,

"attribute2format":<String_value>,

"attribute3":<String_value>,

"attribute3expr":<String_value>,

"attribute3friendlyname":<String_value>,
```

```
"attribute3format":<String_value>,

"attribute4":<String_value>,

"attribute4expr":<String_value>,

"attribute4friendlyname":<String_value>,

"attribute4format":<String_value>,

"attribute5":<String_value>,

"attribute5expr":<String_value>,

"attribute5friendlyname":<String_value>,

"attribute5format":<String_value>,

"attribute6":<String_value>,

"attribute6expr":<String_value>,

"attribute6friendlyname":<String_value>,

"attribute6format":<String_value>,

"attribute7":<String_value>,

"attribute7expr":<String_value>,

"attribute7friendlyname":<String_value>,

"attribute7format":<String_value>,

"attribute8":<String_value>,

"attribute8expr":<String_value>,

"attribute8friendlyname":<String_value>,

"attribute8format":<String_value>,

"attribute9":<String_value>,
```

```
"attribute9expr":<String_value>,

"attribute9friendlyname":<String_value>,

"attribute9format":<String_value>,

"attribute10":<String_value>,

"attribute10expr":<String_value>,

"attribute10friendlyname":<String_value>,

"attribute10format":<String_value>,

"attribute11":<String_value>,

"attribute11expr":<String_value>,

"attribute11friendlyname":<String_value>,

"attribute11format":<String_value>,

"attribute12":<String_value>,

"attribute12expr":<String_value>,

"attribute12friendlyname":<String_value>,

"attribute12format":<String_value>,

"attribute13":<String_value>,

"attribute13expr":<String_value>,

"attribute13friendlyname":<String_value>,

"attribute13format":<String_value>,

"attribute14":<String_value>,

"attribute14expr":<String_value>,

"attribute14friendlyname":<String_value>,
```

```
"attribute14format":<String_value>,

"attribute15":<String_value>,

"attribute15expr":<String_value>,

"attribute15friendlyname":<String_value>,

"attribute15format":<String_value>,

"attribute16":<String_value>,

"attribute16expr":<String_value>,

"attribute16friendlyname":<String_value>,

"attribute16format":<String_value>,

"encryptassertion":<String_value>,

"samlspcertname":<String_value>,

"encryptionalgorithm":<String_value>,

"skewtime":<Double_value>,

"signassertion":<String_value>,

"signatureservice":<String_value>

}}
```

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error

unset

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/vpnsamlssoprofile?**action=unset**

HTTP Method:POST

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Content-Type:application/json

Request Payload:

```
{"vpnsamlssoprofile":{

<b>"name":<String_value>,

</b>"samlsigningcertname":true,

"sendpassword":true,

"samlissuername":true,

"relaystaterule":true,

"signaturealg":true,

"digestmethod":true,

"audience":true,

"nameidformat":true,

"nameidexpr":true,

"attribute1":true,

"attribute1friendlyname":true,

"attribute1format":true,

"attribute2":true,

"attribute2friendlyname":true,

"attribute2format":true,
```

```
"attribute3":true,

"attribute3friendlyname":true,

"attribute3format":true,

"attribute4":true,

"attribute4friendlyname":true,

"attribute4format":true,

"attribute5":true,

"attribute5friendlyname":true,

"attribute5format":true,

"attribute6":true,

"attribute6friendlyname":true,

"attribute6format":true,

"attribute7":true,

"attribute7friendlyname":true,

"attribute7format":true,

"attribute8":true,

"attribute8friendlyname":true,

"attribute8format":true,

"attribute9":true,

"attribute9friendlyname":true,

"attribute9format":true,

"attribute10":true,
```

```
"attribute10friendlyname":true,

"attribute10format":true,

"attribute11":true,

"attribute11friendlyname":true,

"attribute11format":true,

"attribute12":true,

"attribute12friendlyname":true,

"attribute12format":true,

"attribute13":true,

"attribute13friendlyname":true,

"attribute13format":true,

"attribute14":true,

"attribute14friendlyname":true,

"attribute14format":true,

"attribute15":true,

"attribute15friendlyname":true,

"attribute15format":true,

"attribute16":true,

"attribute16friendlyname":true,

"attribute16format":true,

"encryptassertion":true,

"samlspcertname":true,
```

```
"encryptionalgorithm":true,

"skewtime":true,

"signassertion":true,

"signatureservice":true

}}
```

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error

get (all)

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/vpnsamlssoprofile

Query-parameters:

attrs

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/vpnsamlssoprofile?**attrs=property-name1,property-name2**

Use this query parameter to specify the resource details that you want to retrieve.

filter

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/vpnsamlssoprofile?**filter=property-name1:property-val1,property-name2:property-val2**

Use this query-parameter to get the filtered set of vpnsamlssoprofile resources configured on Citrix ADC. Filtering can be done on any of the properties of the resource.

view

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/vpnsamlssoprofile?**view=summary**

Note: By default, the retrieved results are displayed in detail view (?view=detail).

pagination

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/vpnsamlssoprofile?**pagesize=#no;pageno=#no**

Use this query-parameter to get the vpnsamlssoprofile resources in chunks.

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "vpnsamlssoprofile": [ {

"name":<String_value>,

"samlsigningcertname":<String_value>,

"assertionconsumerserviceurl":<String_value>,

"sendpassword":<String_value>,

"samlissuername":<String_value>,

"relaystaterule":<String_value>,

"signaturealg":<String_value>,

"digestmethod":<String_value>,

"audience":<String_value>,
```

```
"nameidformat":<String_value>,

"nameidexpr":<String_value>,

"attribute1":<String_value>,

"attribute2":<String_value>,

"attribute3":<String_value>,

"attribute4":<String_value>,

"attribute5":<String_value>,

"attribute6":<String_value>,

"attribute7":<String_value>,

"attribute8":<String_value>,

"attribute9":<String_value>,

"attribute10":<String_value>,

"attribute11":<String_value>,

"attribute12":<String_value>,

"attribute13":<String_value>,

"attribute14":<String_value>,

"attribute15":<String_value>,

"attribute16":<String_value>,

"attribute1friendlyname":<String_value>,

"attribute2friendlyname":<String_value>,

"attribute3friendlyname":<String_value>,

"attribute4friendlyname":<String_value>,
```

```
"attribute5friendlyname":<String_value>,

"attribute6friendlyname":<String_value>,

"attribute7friendlyname":<String_value>,

"attribute8friendlyname":<String_value>,

"attribute9friendlyname":<String_value>,

"attribute10friendlyname":<String_value>,

"attribute11friendlyname":<String_value>,

"attribute12friendlyname":<String_value>,

"attribute13friendlyname":<String_value>,

"attribute14friendlyname":<String_value>,

"attribute15friendlyname":<String_value>,

"attribute16friendlyname":<String_value>,

"attribute1format":<String_value>,

"attribute2format":<String_value>,

"attribute3format":<String_value>,

"attribute4format":<String_value>,

"attribute5format":<String_value>,

"attribute6format":<String_value>,

"attribute7format":<String_value>,

"attribute8format":<String_value>,

"attribute9format":<String_value>,

"attribute10format":<String_value>,
```

```
"attribute11format":<String_value>,

"attribute12format":<String_value>,

"attribute13format":<String_value>,

"attribute14format":<String_value>,

"attribute15format":<String_value>,

"attribute16format":<String_value>,

"attribute1expr":<String_value>,

"attribute2expr":<String_value>,

"attribute3expr":<String_value>,

"attribute4expr":<String_value>,

"attribute5expr":<String_value>,

"attribute6expr":<String_value>,

"attribute7expr":<String_value>,

"attribute8expr":<String_value>,

"attribute9expr":<String_value>,

"attribute10expr":<String_value>,

"attribute11expr":<String_value>,

"attribute12expr":<String_value>,

"attribute13expr":<String_value>,

"attribute14expr":<String_value>,

"attribute15expr":<String_value>,

"attribute16expr":<String_value>,
```

```
"encryptassertion":<String_value>,

"samlspcertname":<String_value>,

"encryptionalgorithm":<String_value>,

"skewtime":<Double_value>,

"signassertion":<String_value>,

"signatureservice":<String_value>

}]}
```

get

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/vpnsamlssoprofile/name_value<String>

Query-parameters:

attrs

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/vpnsamlssoprofile/name_value<String>?**attrs=property-name1,property-name2**

Use this query parameter to specify the resource details that you want to retrieve.

view

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/vpnsamlssoprofile/name_value<String>?**view=summary**

Note: By default, the retrieved results are displayed in detail view (?view=detail).

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "vpnsamlssoprofile": [ {

"name":<String_value>,

"samlsigningcertname":<String_value>,

"assertionconsumerserviceurl":<String_value>,

"sendpassword":<String_value>,

"samlissuername":<String_value>,

"relaystaterule":<String_value>,

"signaturealg":<String_value>,

"digestmethod":<String_value>,

"audience":<String_value>,

"nameidformat":<String_value>,

"nameidexpr":<String_value>,

"attribute1":<String_value>,

"attribute2":<String_value>,

"attribute3":<String_value>,

"attribute4":<String_value>,

"attribute5":<String_value>,
```

```
"attribute6":<String_value>,

"attribute7":<String_value>,

"attribute8":<String_value>,

"attribute9":<String_value>,

"attribute10":<String_value>,

"attribute11":<String_value>,

"attribute12":<String_value>,

"attribute13":<String_value>,

"attribute14":<String_value>,

"attribute15":<String_value>,

"attribute16":<String_value>,

"attribute1friendlyname":<String_value>,

"attribute2friendlyname":<String_value>,

"attribute3friendlyname":<String_value>,

"attribute4friendlyname":<String_value>,

"attribute5friendlyname":<String_value>,

"attribute6friendlyname":<String_value>,

"attribute7friendlyname":<String_value>,

"attribute8friendlyname":<String_value>,

"attribute9friendlyname":<String_value>,

"attribute10friendlyname":<String_value>,

"attribute11friendlyname":<String_value>,
```

```
"attribute12friendlyname":<String_value>,

"attribute13friendlyname":<String_value>,

"attribute14friendlyname":<String_value>,

"attribute15friendlyname":<String_value>,

"attribute16friendlyname":<String_value>,

"attribute1format":<String_value>,

"attribute2format":<String_value>,

"attribute3format":<String_value>,

"attribute4format":<String_value>,

"attribute5format":<String_value>,

"attribute6format":<String_value>,

"attribute7format":<String_value>,

"attribute8format":<String_value>,

"attribute9format":<String_value>,

"attribute10format":<String_value>,

"attribute11format":<String_value>,

"attribute12format":<String_value>,

"attribute13format":<String_value>,

"attribute14format":<String_value>,

"attribute15format":<String_value>,

"attribute16format":<String_value>,

"attribute1expr":<String_value>,
```

```
"attribute2expr":<String_value>,

"attribute3expr":<String_value>,

"attribute4expr":<String_value>,

"attribute5expr":<String_value>,

"attribute6expr":<String_value>,

"attribute7expr":<String_value>,

"attribute8expr":<String_value>,

"attribute9expr":<String_value>,

"attribute10expr":<String_value>,

"attribute11expr":<String_value>,

"attribute12expr":<String_value>,

"attribute13expr":<String_value>,

"attribute14expr":<String_value>,

"attribute15expr":<String_value>,

"attribute16expr":<String_value>,

"encryptassertion":<String_value>,

"samlspcertname":<String_value>,

"encryptionalgorithm":<String_value>,

"skewtime":<Double_value>,

"signassertion":<String_value>,

"signatureservice":<String_value>

}]}
```

count

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/vpnsamlssoprofile?**count=yes**

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "vpnsamlssoprofile": [ { "__count": "#no"} ] }
```

vpnsessionaction

Configuration for VPN session action resource.

Properties

(click to see Operations)

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
name	<String>	Read-write	Name for the Citrix Gateway profile (action). Must begin with an ASCII alphabetic or underscore (_) character, and must

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
			consist only of ASCII alphanumeric, underscore, hash (#), period (.), space, colon (:), at (@), equals (=), and hyphen (-) characters. Cannot be changed after the profile is created. The following requirement applies only to the Citrix ADC CLI: If the name includes one or more spaces, enclose the name in double or single quotation marks (for example, "my action" or 'my action'). Minimum length = 1
useraccounting	<String>	Read-write	The name of the radiusPolicy to use for RADIUS user accounting info on the session.
httpport	<Integer[]>	Read-write	Destination port numbers other than port 80, added as a comma-separated list. Traffic to these ports is processed as HTTP traffic, which allows functionality,

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
			such as HTTP authorization and single sign-on to a web application to work. Minimum value = 1
winsip	<String>	Read-write	WINS server IP address to add to Citrix Gateway for name resolution.
dnsvservername	<String>	Read-write	Name of the DNS virtual server for the user session. Minimum length = 1
splitdns	<String>	Read-write	Route the DNS requests to the local DNS server configured on the user device, or Citrix Gateway (remote), or both. Possible values = LOCAL, REMOTE, BOTH
sesstimeout	<Double>	Read-write	Number of minutes after which the session times out. Minimum value = 1
clientsecurity	<String>	Read-write	Specify the client security check for the user device to permit a Citrix Gateway session. The web address or IP address is not included in the

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
			expression for the client security check.
clientsecuritygroup	<String>	Read-write	The client security group that will be assigned on failure of the client security check. Users can in general be organized into Groups. In this case, the Client Security Group may have a more restrictive security policy. Minimum length = 1
clientsecuritymessage	<String>	Read-write	The client security message that will be displayed on failure of the client security check. Minimum length = 1 Maximum length = 127
clientsecuritylog	<String>	Read-write	Set the logging of client security checks. Possible values = ON, OFF
splittunnel	<String>	Read-write	Send, through the tunnel, traffic only for intranet applications that are defined in Citrix Gateway. Route all other traffic directly to the Internet. The OFF

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
			setting routes all traffic through Citrix Gateway. With the REVERSE setting, intranet applications define the network traffic that is not intercepted. All network traffic directed to internal IP addresses bypasses the VPN tunnel, while other traffic goes through Citrix Gateway. Reverse split tunneling can be used to log all non-local LAN traffic. For example, if users have a home network and are logged on through the Citrix Gateway Plug-in, network traffic destined to a printer or another device within the home network is not intercepted. Possible values = ON, OFF, REVERSE
locallanaccess	<String>	Read-write	Set local LAN access. If split tunneling is OFF, and you set local LAN access to ON, the local client can route traffic to its local interface. When the

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
			local area network switch is specified, this combination of switches is useful. The client can allow local LAN access to devices that commonly have non-routable addresses, such as local printers or local file servers. Possible values = ON, OFF
rfc1918	<String>	Read-write	As defined in the local area network, allow only the following local area network addresses to bypass the VPN tunnel when the local LAN access feature is enabled: * 10.*.*.*, * 172.16.*.*, * 192.168.*.*. Possible values = ON, OFF
spoofiip	<String>	Read-write	IP address that the intranet application uses to route the connection through the virtual adapter. Possible values = ON, OFF

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
killconnections	<String>	Read-write	Specify whether the Citrix Gateway Plug-in should disconnect all preexisting connections, such as the connections existing before the user logged on to Citrix Gateway, and prevent new incoming connections on the Citrix Gateway Plug-in for Windows and MAC when the user is connected to Citrix Gateway and split tunneling is disabled. Possible values = ON, OFF
transparentintercepti on	<String>	Read-write	Allow access to network resources by using a single IP address and subnet mask or a range of IP addresses. The OFF setting sets the mode to proxy, in which you configure destination and source IP addresses and port numbers. If you are using the Citrix Gateway Plug-in for Windows, set this parameter to ON, in which the mode is set to transparent. If you

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
			are using the Citrix Gateway Plug-in for Java, set this parameter to OFF. Possible values = ON, OFF
windowsclienttype	<String>	Read-write	Choose between two types of Windows Client a) Application Agent - which always runs in the task bar as a standalone application and also has a supporting service which runs permanently when installed b) Activex Control - ActiveX control run by Microsoft Internet Explorer. Possible values = AGENT, PLUGIN
defaultauthorizationaction	<String>	Read-write	Specify the network resources that users have access to when they log on to the internal network. The default setting for authorization is to deny access to all network resources. Citrix recommends using the default

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
			global setting and then creating authorization policies to define the network resources users can access. If you set the default authorization policy to DENY, you must explicitly authorize access to any network resource, which improves security. Possible values = ALLOW, DENY
authorizationgroup	<String>	Read-write	Comma-separated list of groups in which the user is placed when none of the groups that the user is a part of is configured on Citrix Gateway. The authorization policy can be bound to these groups to control access to the resources. Minimum length = 1
smartgroup	<String>	Read-write	This is the default group that is chosen when the authentication succeeds in addition to extracted groups. Minimum length = 1

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
			Maximum length = 64
clientidletimeout	<Double>	Read-write	Time, in minutes, after which to time out the user session if Citrix Gateway does not detect mouse or keyboard activity. Minimum value = 1 Maximum value = 9999
proxy	<String>	Read-write	Set options to apply proxy for accessing the internal resources. Available settings function as follows: * BROWSER - Proxy settings are configured only in Internet Explorer and Firefox browsers. * NS - Proxy settings are configured on the Citrix ADC. * OFF - Proxy settings are not configured.

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
			Possible values = BROWSER, NS, OFF
allprotocolproxy	<String>	Read-write	IP address of the proxy server to use for all protocols supported by Citrix Gateway. Minimum length = 1
httpproxy	<String>	Read-write	IP address of the proxy server to be used for HTTP access for all subsequent connections to the internal network. Minimum length = 1
ftpproxy	<String>	Read-write	IP address of the proxy server to be used for FTP access for all subsequent connections to the internal network. Minimum length = 1
socksproxy	<String>	Read-write	IP address of the proxy server to be used for SOCKS access for all subsequent connections to the internal network. Minimum length = 1
gopherproxy	<String>	Read-write	IP address of the proxy server to be used for GOPHER access for all

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
			subsequent connections to the internal network. Minimum length = 1
sslproxy	<String>	Read-write	IP address of the proxy server to be used for SSL access for all subsequent connections to the internal network. Minimum length = 1
proxyexception	<String>	Read-write	Proxy exception string that will be configured in the browser for bypassing the previously configured proxies. Allowed only if proxy type is Browser. Minimum length = 1
proxylocalbypass	<String>	Read-write	Bypass proxy server for local addresses option in Internet Explorer and Firefox proxy server settings. Possible values = ENABLED, DISABLED
clientcleanupprompt	<String>	Read-write	Prompt for client-side cache clean-up when a client-initiated session closes. Possible values = ON, OFF

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
forcecleanup	<String[]>	Read-write	Force cache clean-up when the user closes a session. You can specify all, none, or any combination of the client-side items. Possible values = none, all, cookie, addressbar, plugin, filesystemapplicatio n, application, applicationdata, clientcertificate, autocomplete, cache
clientoptions	<String>	Read-write	Display only the configured menu options when you select the "Configure Citrix Gateway" option in the Citrix Gateway Plug-in system tray icon for Windows. Possible values = none, all, services, filetransfer, configuration
clientconfiguration	<String[]>	Read-write	Allow users to change client Debug logging level in Configuration tab of the Citrix Gateway Plug-in for Windows.

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
			Possible values = none, trace
sso	<String>	Read-write	Set single sign-on (SSO) for the session. When the user accesses a server, the user's logon credentials are passed to the server for authentication. NOTE : Authentication mechanisms like Basic-authentication require the user credentials to be sent in plaintext which is not secure if the server is running on HTTP (instead of HTTPS). By default, Single Sign On is not attempted to a Public IP. Possible values = ON, OFF
ssocredential	<String>	Read-write	Specify whether to use the primary or secondary authentication credentials for single sign-on to the server.

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
			Possible values = PRIMARY, SECONDARY
windowsautologon	<String>	Read-write	Enable or disable the Windows Auto Logon for the session. If a VPN session is established after this setting is enabled, the user is automatically logged on by using Windows credentials after the system is restarted. Possible values = ON, OFF
usemip	<String>	Read-write	Enable or disable the use of a unique IP address alias, or a mapped IP address, as the client IP address for each client session. Allow Citrix Gateway to use the mapped IP address as an intranet IP address when all other IP addresses are not available. When IP pooling is configured and the mapped IP is used as an intranet IP address, the mapped IP address is used when an

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
			intranet IP address cannot be assigned. Possible values = NS, OFF
useiip	<String>	Read-write	Define IP address pool options. Available settings function as follows: * SPILLOVER - When an address pool is configured and the mapped IP is used as an intranet IP address, the mapped IP address is used when an intranet IP address cannot be assigned. * NOSPILLOVER - When intranet IP addresses are enabled and the mapped IP address is not used, the Transfer Login page appears for users who have used all available intranet IP addresses. * OFF - Address pool is not configured.

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
			Possible values = NOSPILLOVER, SPILLOVER, OFF
clientdebug	<String>	Read-write	Set the trace level on Citrix Gateway. Technical support technicians use these debug logs for in-depth debugging and troubleshooting purposes. Available settings function as follows: * DEBUG - Detailed debug messages are collected and written into the specified file. * STATS - Application audit level error messages and debug statistic counters are written into the specified file. * EVENTS - Application audit-level error messages are written into the specified file. * OFF - Only critical events are logged

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
			into the Windows Application Log. Possible values = debug, stats, events, OFF
loginscript	<String>	Read-write	Path to the logon script that is run when a session is established. Separate multiple scripts by using comma. A "$" in the path signifies that the word following the "$" is an environment variable. Minimum length = 1
logoutscript	<String>	Read-write	Path to the logout script. Separate multiple scripts by using comma. A "$" in the path signifies that the word following the "$" is an environment variable. Minimum length = 1
homepage	<String>	Read-write	Web address of the home page that appears when users log on. Otherwise, users receive the default home page for Citrix Gateway, which is the Access Interface.

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
icaproxy	<String>	Read-write	Enable ICA proxy to configure secure Internet access to servers running Citrix XenApp or XenDesktop by using Citrix Receiver instead of the Citrix Gateway Plug-in. Possible values = ON, OFF
wihome	<String>	Read-write	Web address of the Web Interface server, such as http://<ipAddress>/Citrix/XenApp, or Receiver for Web, which enumerates the virtualized resources, such as XenApp, XenDesktop, and cloud applications. This web address is used as the home page in ICA proxy mode. If Client Choices is ON, you must configure this setting. Because the user can choose between FullClient and ICAProxy, the user may see a different home page. An Internet web site may

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
			appear if the user gets the FullClient option, or a Web Interface site if the user gets the ICAProxy option. If the setting is not configured, the XenApp option does not appear as a client choice.
wihomeaddresstype	<String>	Read-write	Type of the wihome address(IPV4/V6). Possible values = IPV4, IPV6
citrixreceiverhome	<String>	Read-write	Web address for the Citrix Receiver home page. Configure Citrix Gateway so that when users log on to the appliance, the Citrix Gateway Plug-in opens a web browser that allows single sign-on to the Citrix Receiver home page.
wiportalmode	<String>	Read-write	Layout on the Access Interface. The COMPACT value indicates the use of small icons. Possible values = NORMAL, COMPACT

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
clientchoices	<String>	Read-write	Provide users with multiple logon options. With client choices, users have the option of logging on by using the Citrix Gateway Plug-in for Windows, Citrix Gateway Plug-in for Java, the Web Interface, or clientless access from one location. Depending on how Citrix Gateway is configured, users are presented with up to three icons for logon choices. The most common are the Citrix Gateway Plug-in for Windows, Web Interface, and clientless access. Possible values = ON, OFF
epaclienttype	<String>	Read-write	Choose between two types of End point Windows Client a) Application Agent - which always runs in the task bar as a standalone application and also has a supporting service which runs

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
			permanently when installed b) Activex Control - ActiveX control run by Microsoft Internet Explorer. Possible values = AGENT, PLUGIN
iipdnssuffix	<String>	Read-write	An intranet IP DNS suffix. When a user logs on to Citrix Gateway and is assigned an IP address, a DNS record for the user name and IP address combination is added to the Citrix Gateway DNS cache. You can configure a DNS suffix to append to the user name when the DNS record is added to the cache. You can reach to the host from where the user is logged on by using the user's name, which can be easier to remember than an IP address. When the user logs off from Citrix Gateway, the record is removed from the DNS cache.

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
			Minimum length = 1
forcedtimeout	<Double>	Read-write	Force a disconnection from the Citrix Gateway Plug-in with Citrix Gateway after a specified number of minutes. If the session closes, the user must log on again. Minimum value = 1 Maximum value = 65535
forcedtimeoutwarning	<Double>	Read-write	Number of minutes to warn a user before the user session is disconnected. Minimum value = 1 Maximum value = 255
ntdomain	<String>	Read-write	Single sign-on domain to use for single sign-on to applications in the internal network. This setting can be overwritten by the domain that users specify at the time of logon or by the domain that the authentication server returns. Minimum length = 1

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
			Maximum length = 32
clientlessvpnmode	<String>	Read-write	Enable clientless access for web, XenApp or XenDesktop, and FileShare resources without installing the Citrix Gateway Plug-in. Available settings function as follows: * ON - Allow only clientless access. * OFF - Allow clientless access after users log on with the Citrix Gateway Plug-in. * DISABLED - Do not allow clientless access. Possible values = ON, OFF, DISABLED
emailhome	<String>	Read-write	Web address for the web-based email, such as Outlook Web Access.
clientlessmodeurlencoding	<String>	Read-write	When clientless access is enabled, you can choose to encode the addresses of internal web applications or

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
			to leave the address as clear text. Available settings function as follows: * OPAQUE - Use standard encoding mechanisms to make the domain and protocol part of the resource unclear to users. * CLEAR - Do not encode the web address and make it visible to users. * ENCRYPT - Allow the domain and protocol to be encrypted using a session key. When the web address is encrypted, the URL is different for each user session for the same web resource. If users bookmark the encoded web address, save it in the web browser and then log off, they cannot connect to the web address when they log on and use the bookmark. If users

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
			save the encrypted bookmark in the Access Interface during their session, the bookmark works each time the user logs on. Possible values = TRANSPARENT, OPAQUE, ENCRYPT
clientlesspersistentcookie	<String>	Read-write	State of persistent cookies in clientless access mode. Persistent cookies are required for accessing certain features of SharePoint, such as opening and editing Microsoft Word, Excel, and PowerPoint documents hosted on the SharePoint server. A persistent cookie remains on the user device and is sent with each HTTP request. Citrix Gateway encrypts the persistent cookie before sending it to the plug-in on the user device, and refreshes the cookie periodically as long as the session

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
			exists. The cookie becomes stale if the session ends. Available settings function as follows: * ALLOW - Enable persistent cookies. Users can open and edit Microsoft documents stored in SharePoint. * DENY - Disable persistent cookies. Users cannot open and edit Microsoft documents stored in SharePoint. * PROMPT - Prompt users to allow or deny persistent cookies during the session. Persistent cookies are not required for clientless access if users do not connect to SharePoint. Possible values = ALLOW, DENY, PROMPT
allowedlogingroups	<String>	Read-write	Specify groups that have permission to log on to Citrix

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
			Gateway. Users who do not belong to this group or groups are denied access even if they have valid credentials. Minimum length = 1 Maximum length = 511
securebrowse	<String>	Read-write	Allow users to connect through Citrix Gateway to network resources from iOS and Android mobile devices with Citrix Receiver. Users do not need to establish a full VPN tunnel to access resources in the secure network. Possible values = ENABLED, DISABLED
storefronturl	<String>	Read-write	Web address for StoreFront to be used in this session for enumeration of resources from XenApp or XenDesktop. Minimum length = 1 Maximum length = 255

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
sfgatewayauthtype	<String>	Read-write	The authentication type configured for the Citrix Gateway on StoreFront. Possible values = domain, RSA, domainAndRSA, SMS, smartCard, sfAuth, sfAuthAndRSA
kcdaccount	<String>	Read-write	The kcd account details to be used in SSO. Minimum length = 1 Maximum length = 32
rdpclientprofilename	<String>	Read-write	Name of the RDP profile associated with the vserver. Minimum length = 1 Maximum length = 31
windowspluginupgrade	<String>	Read-write	Option to set plugin upgrade behaviour for Win. Possible values = Always, Essential, Never
macpluginupgrade	<String>	Read-write	Option to set plugin upgrade behaviour for Mac.

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
			Possible values = Always, Essential, Never
linuxpluginupgrade	<String>	Read-write	Option to set plugin upgrade behaviour for Linux. Possible values = Always, Essential, Never
iconwithreceiver	<String>	Read-write	Option to decide whether to show plugin icon along with receiver. Possible values = ON, OFF
alwaysonprofilename	<String>	Read-write	Name of the AlwaysON profile associated with the session action. The builtin profile named none can be used to explicitly disable AlwaysON for the session action. Minimum length = 1 Maximum length = 31
autoproxyurl	<String>	Read-write	URL to auto proxy config file.
advancedclientlessvpnmode	<String>	Read-write	Option to enable/disable Advanced ClientlessVpnMode.

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
			Default value: DISABLED Possible values = ENABLED, DISABLED
pcoipprofilename	<String>	Read-write	Name of the PCOIP profile associated with the session action. The builtin profile named none can be used to explicitly disable PCOIP for the session action. Minimum length = 1 Maximum length = 31
fqdnspoofedip	<String>	Read-write	Spoofed IP address range that can be used by client for FQDN based split tunneling. Minimum length = 1
netmask	<String>	Read-write	The netmask for the spoofed ip address. Minimum length = 1
clientidletimeoutwarning	<Double>	Read-only	The time after which the client gets a timeout warning, in minutes.
__count	<Double>	Read-only	count parameter

Operations

(click to see Properties)

ADD| DELETE| UPDATE| UNSET| GET (ALL)| GET| COUNT

Some options that you can use for each operations:

- **Getting warnings in response:**NITRO allows you to get warnings in an operation by specifying the "warning" query parameter as "yes". For example, to get warnings while connecting to the Citrix ADC appliance, the URL is as follows:

  http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/login?warning=yes

  If any, the warnings are displayed in the response payload with the HTTP code "209 X-NITRO-WARNING".

- **Authenticated access for individual NITRO operations:**NITRO allows you to logon to the Citrix ADC appliance to perform individual operations. You can use this option instead of creating a NITRO session (using the login object) and then using that session to perform all operations,

  To do this, you must specify the username and password in the request header of the NITRO request as follows:

  X-NITRO-USER:<username>

  X-NITRO-PASS:<password>

  Note: In such cases, make sure that the request header DOES not include the following:

  Cookie:NITRO_AUTH_TOKEN=<tokenvalue>

`*Note:`*

Mandatory parameters are marked in redand placeholder content is marked in <green>.

add

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/vpnsessionaction

HTTP Method:POST

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Content-Type:application/json

Request Payload:

```
{"vpnsessionaction":{

<b>"name":<String_value>,

</b>"useraccounting":<String_value>,

"httpport":<Integer[]_value>,

"winsip":<String_value>,

"dnsvservername":<String_value>,

"splitdns":<String_value>,

"sesstimeout":<Double_value>,

"clientsecurity":<String_value>,

"clientsecuritygroup":<String_value>,

"clientsecuritymessage":<String_value>,

"clientsecuritylog":<String_value>,

"splittunnel":<String_value>,

"locallanaccess":<String_value>,

"rfc1918":<String_value>,

"spoofiip":<String_value>,

"killconnections":<String_value>,

"transparentinterception":<String_value>,

"windowsclienttype":<String_value>,

"defaultauthorizationaction":<String_value>,
```

```
"authorizationgroup":<String_value>,

"smartgroup":<String_value>,

"clientidletimeout":<Double_value>,

"proxy":<String_value>,

"allprotocolproxy":<String_value>,

"httpproxy":<String_value>,

"ftpproxy":<String_value>,

"socksproxy":<String_value>,

"gopherproxy":<String_value>,

"sslproxy":<String_value>,

"proxyexception":<String_value>,

"proxylocalbypass":<String_value>,

"clientcleanupprompt":<String_value>,

"forcecleanup":<String[]_value>,

"clientoptions":<String_value>,

"clientconfiguration":<String[]_value>,

"sso":<String_value>,

"ssocredential":<String_value>,

"windowsautologon":<String_value>,

"usemip":<String_value>,

"useiip":<String_value>,

"clientdebug":<String_value>,
```

```
"loginscript":<String_value>,

"logoutscript":<String_value>,

"homepage":<String_value>,

"icaproxy":<String_value>,

"wihome":<String_value>,

"wihomeaddresstype":<String_value>,

"citrixreceiverhome":<String_value>,

"wiportalmode":<String_value>,

"clientchoices":<String_value>,

"epaclienttype":<String_value>,

"iipdnssuffix":<String_value>,

"forcedtimeout":<Double_value>,

"forcedtimeoutwarning":<Double_value>,

"ntdomain":<String_value>,

"clientlessvpnmode":<String_value>,

"emailhome":<String_value>,

"clientlessmodeurlencoding":<String_value>,

"clientlesspersistentcookie":<String_value>,

"allowedlogingroups":<String_value>,

"securebrowse":<String_value>,

"storefronturl":<String_value>,

"sfgatewayauthtype":<String_value>,
```

```
"kcdaccount":<String_value>,

"rdpclientprofilename":<String_value>,

"windowspluginupgrade":<String_value>,

"macpluginupgrade":<String_value>,

"linuxpluginupgrade":<String_value>,

"iconwithreceiver":<String_value>,

"alwaysonprofilename":<String_value>,

"autoproxyurl":<String_value>,

"advancedclientlessvpnmode":<String_value>,

"pcoipprofilename":<String_value>,

"fqdnspoofedip":<String_value>,

"netmask":<String_value>

}}
```

Response:

HTTP Status Code on Success: 201 Created HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error

delete

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/vpnsessionaction/name_value<String>

HTTP Method:DELETE

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue>

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error

update

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/vpnsessionaction

HTTP Method:PUT

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Content-Type:application/json

Request Payload:

```
{"vpnsessionaction":{

<b>"name":<String_value>,

</b>"useraccounting":<String_value>,

"httpport":<Integer[]_value>,

"winsip":<String_value>,

"dnsvservername":<String_value>,

"splitdns":<String_value>,

"sesstimeout":<Double_value>,

"clientsecurity":<String_value>,

"clientsecuritygroup":<String_value>,

"clientsecuritymessage":<String_value>,

"clientsecuritylog":<String_value>,
```

```
"splittunnel":<String_value>,

"locallanaccess":<String_value>,

"rfc1918":<String_value>,

"spoofiip":<String_value>,

"killconnections":<String_value>,

"transparentinterception":<String_value>,

"windowsclienttype":<String_value>,

"defaultauthorizationaction":<String_value>,

"authorizationgroup":<String_value>,

"smartgroup":<String_value>,

"clientidletimeout":<Double_value>,

"proxy":<String_value>,

"allprotocolproxy":<String_value>,

"httpproxy":<String_value>,

"ftpproxy":<String_value>,

"socksproxy":<String_value>,

"gopherproxy":<String_value>,

"sslproxy":<String_value>,

"proxyexception":<String_value>,

"proxylocalbypass":<String_value>,

"clientcleanupprompt":<String_value>,

"forcecleanup":<String[]_value>,
```

```
"clientoptions":<String_value>,

"clientconfiguration":<String[]_value>,

"sso":<String_value>,

"ssocredential":<String_value>,

"windowsautologon":<String_value>,

"usemip":<String_value>,

"useiip":<String_value>,

"clientdebug":<String_value>,

"loginscript":<String_value>,

"logoutscript":<String_value>,

"homepage":<String_value>,

"icaproxy":<String_value>,

"wihome":<String_value>,

"wihomeaddresstype":<String_value>,

"citrixreceiverhome":<String_value>,

"wiportalmode":<String_value>,

"clientchoices":<String_value>,

"epaclienttype":<String_value>,

"iipdnssuffix":<String_value>,

"forcedtimeout":<Double_value>,

"forcedtimeoutwarning":<Double_value>,

"ntdomain":<String_value>,
```

```
"clientlessvpnmode":<String_value>,

"emailhome":<String_value>,

"clientlessmodeurlencoding":<String_value>,

"clientlesspersistentcookie":<String_value>,

"allowedlogingroups":<String_value>,

"securebrowse":<String_value>,

"storefronturl":<String_value>,

"sfgatewayauthtype":<String_value>,

"kcdaccount":<String_value>,

"rdpclientprofilename":<String_value>,

"windowspluginupgrade":<String_value>,

"macpluginupgrade":<String_value>,

"linuxpluginupgrade":<String_value>,

"iconwithreceiver":<String_value>,

"alwaysonprofilename":<String_value>,

"autoproxyurl":<String_value>,

"advancedclientlessvpnmode":<String_value>,

"pcoipprofilename":<String_value>,

"fqdnspoofedip":<String_value>,

"netmask":<String_value>

}}
```

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error

unset

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/vpnsessionaction?**action=unset**

HTTP Method:POST

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Content-Type:application/json

Request Payload:

```
{"vpnsessionaction":{

<b>"name":<String_value>,

</b>"useraccounting":true,

"httpport":true,

"winsip":true,

"dnsvservername":true,

"splitdns":true,

"sesstimeout":true,

"clientsecurity":true,

"clientsecuritygroup":true,

"clientsecuritymessage":true,
```

```
"clientsecuritylog":true,

"splittunnel":true,

"locallanaccess":true,

"rfc1918":true,

"spoofiip":true,

"killconnections":true,

"transparentinterception":true,

"windowsclienttype":true,

"defaultauthorizationaction":true,

"authorizationgroup":true,

"smartgroup":true,

"clientidletimeout":true,

"proxy":true,

"allprotocolproxy":true,

"httpproxy":true,

"ftpproxy":true,

"socksproxy":true,

"gopherproxy":true,

"sslproxy":true,

"proxyexception":true,

"proxylocalbypass":true,

"clientcleanupprompt":true,
```

```
"forcecleanup":true,

"clientoptions":true,

"clientconfiguration":true,

"sso":true,

"ssocredential":true,

"windowsautologon":true,

"usemip":true,

"useiip":true,

"clientdebug":true,

"loginscript":true,

"logoutscript":true,

"homepage":true,

"icaproxy":true,

"wihome":true,

"citrixreceiverhome":true,

"wiportalmode":true,

"clientchoices":true,

"epaclienttype":true,

"iipdnssuffix":true,

"forcedtimeout":true,

"forcedtimeoutwarning":true,

"ntdomain":true,
```

```
"clientlessvpnmode":true,

"emailhome":true,

"clientlessmodeurlencoding":true,

"clientlesspersistentcookie":true,

"allowedlogingroups":true,

"securebrowse":true,

"storefronturl":true,

"sfgatewayauthtype":true,

"kcdaccount":true,

"rdpclientprofilename":true,

"windowspluginupgrade":true,

"macpluginupgrade":true,

"linuxpluginupgrade":true,

"iconwithreceiver":true,

"alwaysonprofilename":true,

"autoproxyurl":true,

"advancedclientlessvpnmode":true,

"pcoipprofilename":true,

"fqdnspoofedip":true,

"netmask":true

}}
```

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error

get (all)

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/vpnsessionaction

Query-parameters:

attrs

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/vpnsessionaction?**attrs=property-name1,property-name2**

Use this query parameter to specify the resource details that you want to retrieve.

filter

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/vpnsessionaction?**filter=property-name1:property-val1,property-name2:property-val2**

Use this query-parameter to get the filtered set of vpnsessionaction resources configured on Citrix ADC. Filtering can be done on any of the properties of the resource.

view

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/vpnsessionaction?**view=summary**

Note: By default, the retrieved results are displayed in detail view (?view=detail).

pagination

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/vpnsessionaction?**pagesize=#no;pageno=#no**

Use this query-parameter to get the vpnsessionaction resources in chunks.

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "vpnsessionaction": [ {

"name":<String_value>,

"useraccounting":<String_value>,

"httpport":<Integer[]_value>,

"winsip":<String_value>,

"dnsvservername":<String_value>,

"splitdns":<String_value>,

"sesstimeout":<Double_value>,

"clientsecurity":<String_value>,

"clientsecuritygroup":<String_value>,

"clientsecuritymessage":<String_value>,

"clientsecuritylog":<String_value>,

"splittunnel":<String_value>,

"locallanaccess":<String_value>,

"rfc1918":<String_value>,
```

```
"spoofiip":<String_value>,

"killconnections":<String_value>,

"transparentinterception":<String_value>,

"windowsclienttype":<String_value>,

"defaultauthorizationaction":<String_value>,

"authorizationgroup":<String_value>,

"smartgroup":<String_value>,

"clientidletimeout":<Double_value>,

"clientidletimeoutwarning":<Double_value>,

"proxy":<String_value>,

"allprotocolproxy":<String_value>,

"httpproxy":<String_value>,

"ftpproxy":<String_value>,

"socksproxy":<String_value>,

"gopherproxy":<String_value>,

"sslproxy":<String_value>,

"proxyexception":<String_value>,

"proxylocalbypass":<String_value>,

"clientcleanupprompt":<String_value>,

"forcecleanup":<String[]_value>,

"clientoptions":<String_value>,

"clientconfiguration":<String[]_value>,
```

```
"sso":<String_value>,

"ssocredential":<String_value>,

"windowsautologon":<String_value>,

"usemip":<String_value>,

"useiip":<String_value>,

"clientdebug":<String_value>,

"loginscript":<String_value>,

"logoutscript":<String_value>,

"homepage":<String_value>,

"icaproxy":<String_value>,

"wihome":<String_value>,

"wihomeaddresstype":<String_value>,

"citrixreceiverhome":<String_value>,

"wiportalmode":<String_value>,

"clientchoices":<String_value>,

"epaclienttype":<String_value>,

"iipdnssuffix":<String_value>,

"forcedtimeout":<Double_value>,

"forcedtimeoutwarning":<Double_value>,

"ntdomain":<String_value>,

"clientlessvpnmode":<String_value>,

"clientlessmodeurlencoding":<String_value>,
```

```
"clientlesspersistentcookie":<String_value>,

"emailhome":<String_value>,

"allowedlogingroups":<String_value>,

"securebrowse":<String_value>,

"storefronturl":<String_value>,

"sfgatewayauthtype":<String_value>,

"kcdaccount":<String_value>,

"rdpclientprofilename":<String_value>,

"windowspluginupgrade":<String_value>,

"macpluginupgrade":<String_value>,

"linuxpluginupgrade":<String_value>,

"iconwithreceiver":<String_value>,

"alwaysonprofilename":<String_value>,

"autoproxyurl":<String_value>,

"advancedclientlessvpnmode":<String_value>,

"pcoipprofilename":<String_value>,

"fqdnspoofedip":<String_value>,

"netmask":<String_value>

}]}
```

get

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/vpnsessionaction/name_value<String>

Query-parameters:

attrs

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/vpnsessionaction/name_value<String>?**attrs=property-name1,property-name2**

Use this query parameter to specify the resource details that you want to retrieve.

view

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/vpnsessionaction/name_value<String>?**view=summary**

Note: By default, the retrieved results are displayed in detail view (?view=detail).

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "vpnsessionaction": [ {

"name":<String_value>,

"useraccounting":<String_value>,

"httpport":<Integer[]_value>,

"winsip":<String_value>,

"dnsvservername":<String_value>,
```

```
"splitdns":<String_value>,

"sesstimeout":<Double_value>,

"clientsecurity":<String_value>,

"clientsecuritygroup":<String_value>,

"clientsecuritymessage":<String_value>,

"clientsecuritylog":<String_value>,

"splittunnel":<String_value>,

"locallanaccess":<String_value>,

"rfc1918":<String_value>,

"spoofiip":<String_value>,

"killconnections":<String_value>,

"transparentinterception":<String_value>,

"windowsclienttype":<String_value>,

"defaultauthorizationaction":<String_value>,

"authorizationgroup":<String_value>,

"smartgroup":<String_value>,

"clientidletimeout":<Double_value>,

"clientidletimeoutwarning":<Double_value>,

"proxy":<String_value>,

"allprotocolproxy":<String_value>,

"httpproxy":<String_value>,

"ftpproxy":<String_value>,
```

```
"socksproxy":<String_value>,

"gopherproxy":<String_value>,

"sslproxy":<String_value>,

"proxyexception":<String_value>,

"proxylocalbypass":<String_value>,

"clientcleanupprompt":<String_value>,

"forcecleanup":<String[]_value>,

"clientoptions":<String_value>,

"clientconfiguration":<String[]_value>,

"sso":<String_value>,

"ssocredential":<String_value>,

"windowsautologon":<String_value>,

"usemip":<String_value>,

"useiip":<String_value>,

"clientdebug":<String_value>,

"loginscript":<String_value>,

"logoutscript":<String_value>,

"homepage":<String_value>,

"icaproxy":<String_value>,

"wihome":<String_value>,

"wihomeaddresstype":<String_value>,

"citrixreceiverhome":<String_value>,
```

```
"wiportalmode":<String_value>,

"clientchoices":<String_value>,

"epaclienttype":<String_value>,

"iipdnssuffix":<String_value>,

"forcedtimeout":<Double_value>,

"forcedtimeoutwarning":<Double_value>,

"ntdomain":<String_value>,

"clientlessvpnmode":<String_value>,

"clientlessmodeurlencoding":<String_value>,

"clientlesspersistentcookie":<String_value>,

"emailhome":<String_value>,

"allowedlogingroups":<String_value>,

"securebrowse":<String_value>,

"storefronturl":<String_value>,

"sfgatewayauthtype":<String_value>,

"kcdaccount":<String_value>,

"rdpclientprofilename":<String_value>,

"windowspluginupgrade":<String_value>,

"macpluginupgrade":<String_value>,

"linuxpluginupgrade":<String_value>,

"iconwithreceiver":<String_value>,

"alwaysonprofilename":<String_value>,
```

```
"autoproxyurl":<String_value>,

"advancedclientlessvpnmode":<String_value>,

"pcoipprofilename":<String_value>,

"fqdnspoofedip":<String_value>,

"netmask":<String_value>

}]}
```

count

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/vpnsessionaction?**count=yes**

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "vpnsessionaction": [ { "__count": "#no"} ] }
```

vpnsessionpolicy

Configuration for VPN session policy resource.

Properties

(click to see Operations)

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
name	<String>	Read-write	Name for the new session policy that is applied after the user logs on to Citrix Gateway. Minimum length = 1
rule	<String>	Read-write	Expression, or name of a named expression, specifying the traffic that matches the policy. The following requirements apply only to the Citrix ADC CLI: * If the expression includes one or more spaces, enclose the entire expression in double quotation marks. * If the expression itself includes double quotation marks, escape the quotations by using the \ character. * Alternatively, you can use single

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
			quotation marks to enclose the rule, in which case you do not have to escape the double quotation marks.
action	<String>	Read-write	Action to be applied by the new session policy if the rule criteria are met. Minimum length = 1
expressiontype	<String>	Read-only	Type of policy (Classic/Advanced) . Possible values = Classic Policy, Advanced Policy
hits	<Double>	Read-only	Number of hits.
__count	<Double>	Read-only	count parameter

Operations

(click to see Properties)

ADD| DELETE| UPDATE| UNSET| GET (ALL)| GET| COUNT

Some options that you can use for each operations:

- **Getting warnings in response:**NITRO allows you to get warnings in an operation by specifying the "warning" query parameter as "yes". For example, to get warnings while connecting to the Citrix ADC appliance, the URL is as follows:

  http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/login?warning=yes

  If any, the warnings are displayed in the response payload with the HTTP code "209 X-NITRO-WARNING".

- **Authenticated access for individual NITRO operations:**NITRO allows you to logon to the Citrix ADC appliance to perform individual operations. You can use this option instead of creating a NITRO session (using the login object) and then using that session to perform all operations,

  To do this, you must specify the username and password in the request header of the NITRO request as follows:

  X-NITRO-USER:<username>

  X-NITRO-PASS:<password>

  Note: In such cases, make sure that the request header DOES not include the following:

  Cookie:NITRO_AUTH_TOKEN=<tokenvalue>

`*Note:`*

Mandatory parameters are marked in redand placeholder content is marked in <green>.

add

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/vpnsessionpolicy

HTTP Method:POST

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Content-Type:application/json

Request Payload:

```
{"vpnsessionpolicy":{

<b>"name":<String_value>,

</b><b>"rule":<String_value>,

</b><b>"action":<String_value>

</b>}}
```

Response:

HTTP Status Code on Success: 201 Created HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error

delete

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/vpnsessionpolicy/name_value<String>

HTTP Method:DELETE

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue>

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error

update

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/vpnsessionpolicy

HTTP Method:PUT

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Content-Type:application/json

Request Payload:

```
{"vpnsessionpolicy":{

<b>"name":<String_value>,

</b>"rule":<String_value>,

"action":<String_value>

}}
```

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error

unset

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/vpnsessionpolicy?**action=unset**

HTTP Method:POST

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Content-Type:application/json

Request Payload:

```
{"vpnsessionpolicy":{

<b>"name":<String_value>,

</b>"rule":true,

"action":true

}}
```

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error

get (all)

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/vpnsessionpolicy

Query-parameters:

attrs

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/vpnsessionpolicy?**attrs=property-name1,property-name2**

Use this query parameter to specify the resource details that you want to retrieve.

filter

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/vpnsessionpolicy?**filter=property-name1:property-val1,property-name2:property-val2**

Use this query-parameter to get the filtered set of vpnsessionpolicy resources configured on Citrix ADC. Filtering can be done on any of the properties of the resource.

view

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/vpnsessionpolicy?**view=summary**

Note: By default, the retrieved results are displayed in detail view (?view=detail).

pagination

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/vpnsessionpolicy?**pagesize=#no;pageno=#no**

Use this query-parameter to get the vpnsessionpolicy resources in chunks.

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "vpnsessionpolicy": [ {

"name":<String_value>,

"rule":<String_value>,

"action":<String_value>,

"expressiontype":<String_value>,

"hits":<Double_value>

}]}
```

get

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/vpnsessionpolicy/name_value<String>

Query-parameters:

attrs

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/vpnsessionpolicy/name_value<String>?**attrs=property-name1,property-name2**

Use this query parameter to specify the resource details that you want to retrieve.

view

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/vpnsessionpolicy/name_value<String>?**view=summary**

Note: By default, the retrieved results are displayed in detail view (?view=detail).

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "vpnsessionpolicy": [ {

"name":<String_value>,

"rule":<String_value>,

"action":<String_value>,

"expressiontype":<String_value>,

"hits":<Double_value>

}]}
```

count

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/vpnsessionpolicy?**count=yes**

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "vpnsessionpolicy": [ { "__count": "#no"} ] }
```

vpnsessionpolicy_aaagroup_binding

Binding object showing the aaagroup that can be bound to vpnsessionpolicy.

Properties

(click to see Operations)

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
boundto	<String>	Read-write	The entity name to which policy is bound.
name	<String>	Read-write	Name of the session policy to display. Minimum length = 1
priority	<Double>	Read-only	.
activepolicy	<Double>	Read-only	.
__count	<Double>	Read-only	count parameter

Operations

(click to see Properties)

GET| GET (ALL)| COUNT

Some options that you can use for each operations:

- **Getting warnings in response:**NITRO allows you to get warnings in an operation by specifying the "warning" query parameter as "yes". For example, to get warnings while connecting to the Citrix ADC appliance, the URL is as follows:

  http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/login?warning=yes

If any, the warnings are displayed in the response payload with the HTTP code "209 X-NITRO-WARNING".

- **Authenticated access for individual NITRO operations:**NITRO allows you to logon to the Citrix ADC appliance to perform individual operations. You can use this option instead of creating a NITRO session (using the login object) and then using that session to perform all operations,

  To do this, you must specify the username and password in the request header of the NITRO request as follows:

  X-NITRO-USER:<username>

  X-NITRO-PASS:<password>

  Note: In such cases, make sure that the request header DOES not include the following:

  Cookie:NITRO_AUTH_TOKEN=<tokenvalue>

`*Note:`*

Mandatory parameters are marked in redand placeholder content is marked in <green>.

get

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/vpnsessionpolicy_aaagroup_binding/name_value<String>

Query-parameters:

filter

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/vpnsessionpolicy_aaagroup_binding/name_value<String>?**filter=property-name1:property-value1,property-name2:property-value2**

Use this query-parameter to get the filtered set of vpnsessionpolicy_aaagroup_binding resources configured on Citrix ADC. Filtering can be done on any of the properties of the resource.

pagination

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/vpnsessionpolicy_aaagroup_binding/name_value<String>?**pagesize=#no;pageno=#no**

Use this query-parameter to get the vpnsessionpolicy_aaagroup_binding resources in chunks.

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "vpnsessionpolicy_aaagroup_binding": [ {

"boundto":<String_value>,

"name":<String_value>,

"priority":<Double_value>,

"activepolicy":<Double_value>

}]}
```

get (all)

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/vpnsessionpolicy_aaagroup_binding

Query-parameters:

bulkbindings

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/vpnsessionpolicy_aaagroup_binding?**bulkbindings=yes**

NITRO allows you to fetch bindings in bulk.

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "vpnsessionpolicy_aaagroup_binding": [ {

"boundto":<String_value>,

"name":<String_value>,

"priority":<Double_value>,

"activepolicy":<Double_value>

}]}
```

count

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/vpnsessionpolicy_aaagroup_binding/name_value<String>?**count=yes**

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{"vpnsessionpolicy_aaagroup_binding": [ { "__count": "#no"} ] }
```

vpnsessionpolicy_aaauser_binding

Binding object showing the aaauser that can be bound to vpnsessionpolicy.

Properties

(click to see Operations)

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
boundto	<String>	Read-write	The entity name to which policy is bound.
name	<String>	Read-write	Name of the session policy to display. Minimum length = 1
priority	<Double>	Read-only	.
activepolicy	<Double>	Read-only	.
__count	<Double>	Read-only	count parameter

Operations

(click to see Properties)

GET| GET (ALL)| COUNT

Some options that you can use for each operations:

- **Getting warnings in response:**NITRO allows you to get warnings in an operation by specifying the "warning" query parameter as "yes". For example, to get warnings while connecting to the Citrix ADC appliance, the URL is as follows:

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/login?warning=yes

If any, the warnings are displayed in the response payload with the HTTP code "209 X-NITRO-WARNING".

- **Authenticated access for individual NITRO operations:**NITRO allows you to logon to the Citrix ADC appliance to perform individual operations. You can use this option instead of creating a NITRO session (using the login object) and then using that session to perform all operations,

  To do this, you must specify the username and password in the request header of the NITRO request as follows:

  X-NITRO-USER:<username>

  X-NITRO-PASS:<password>

  Note: In such cases, make sure that the request header DOES not include the following:

  Cookie:NITRO_AUTH_TOKEN=<tokenvalue>

`*Note:`*

Mandatory parameters are marked in redand placeholder content is marked in <green>.

get

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/vpnsessionpolicy_aaauser_binding/name_value<String>

Query-parameters:

filter

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/vpnsessionpolicy_aaauser_binding/name_value<String>?**filter=property-name1:property-value1,property-name2:property-value2**

Use this query-parameter to get the filtered set of vpnsessionpolicy_aaauser_binding resources configured on Citrix ADC. Filtering can be done on any of the properties of the resource.

pagination

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/vpnsessionpolicy_aaauser_binding/name_value<String>?**pagesize=#no;pageno=#no**

Use this query-parameter to get the vpnsessionpolicy_aaauser_binding resources in chunks.

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "vpnsessionpolicy_aaauser_binding": [ {

"boundto":<String_value>,

"name":<String_value>,

"priority":<Double_value>,

"activepolicy":<Double_value>

}]}
```

get (all)

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/vpnsessionpolicy_aaauser_binding

Query-parameters:

bulkbindings

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/vpnsessionpolicy_aaauser_binding?**bulkbindings=yes**

NITRO allows you to fetch bindings in bulk.

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "vpnsessionpolicy_aaauser_binding": [ {

"boundto":<String_value>,

"name":<String_value>,

"priority":<Double_value>,

"activepolicy":<Double_value>

}]}
```

count

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/vpnsessionpolicy_aaauser_binding/name_value<String>?**count=yes**

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{"vpnsessionpolicy_aaauser_binding": [ { "__count": "#no"} ] }
```

vpnsessionpolicy_binding

Binding object which returns the resources bound to vpnsessionpolicy.

Properties

(click to see Operations)

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
name	<String>	Read-write	Name of the session policy to display. Minimum length = 1
vpnsessionpolicy_vpnvserver_binding	<vpnsessionpolicy_vpnvserver_binding[]>	Read-only	vpnvserver that can be bound to vpnsessionpolicy.
vpnsessionpolicy_aaagroup_binding	<vpnsessionpolicy_aaagroup_binding[]>	Read-only	aaagroup that can be bound to vpnsessionpolicy.
vpnsessionpolicy_aaauser_binding	<vpnsessionpolicy_aaauser_binding[]>	Read-only	aaauser that can be bound to vpnsessionpolicy.
vpnsessionpolicy_vpnglobal_binding	<vpnsessionpolicy_vpnglobal_binding[]>	Read-only	vpnglobal that can be bound to vpnsessionpolicy.

Operations

(click to see Properties)

GET| GET (ALL)

Some options that you can use for each operations:

- **Getting warnings in response:**NITRO allows you to get warnings in an operation by specifying the "warning" query parameter as "yes". For example, to get warnings while connecting to the Citrix ADC appliance, the URL is as follows:

  http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/login?warning=yes

  If any, the warnings are displayed in the response payload with the HTTP code "209 X-NITRO-WARNING".

- **Authenticated access for individual NITRO operations:**NITRO allows you to logon to the Citrix ADC appliance to perform individual operations. You can use this option instead of creating a NITRO session (using the login object) and then using that session to perform all operations,

  To do this, you must specify the username and password in the request header of the NITRO request as follows:

  X-NITRO-USER:<username>

  X-NITRO-PASS:<password>

  Note: In such cases, make sure that the request header DOES not include the following:

  Cookie:NITRO_AUTH_TOKEN=<tokenvalue>

`*Note:`*

Mandatory parameters are marked in redand placeholder content is marked in <green>.

get

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/vpnsessionpolicy_binding/name_value<String>

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "vpnsessionpolicy_binding": [ {

"name":<String_value>,

"vpnsessionpolicy_aaauser_binding":<vpnsessionpolicy_aaauser_binding[]_val
ue>,

"vpnsessionpolicy_vpnglobal_binding":<vpnsessionpolicy_vpnglobal_binding[]
_value>,

"vpnsessionpolicy_vpnvserver_binding":<vpnsessionpolicy_vpnvserver_binding
[]_value>,

"vpnsessionpolicy_aaagroup_binding":<vpnsessionpolicy_aaagroup_binding[]_v
alue>

}]}
```

get (all)

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/vpnsessionpolicy_binding

Query-parameters:

bulkbindings

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/vpnsessionpolicy_binding?**bulkbindings=yes**

NITRO allows you to fetch bindings in bulk.

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "vpnsessionpolicy_binding": [ {

"name":<String_value>,

"vpnsessionpolicy_aaauser_binding":<vpnsessionpolicy_aaauser_binding[]_val
ue>,

"vpnsessionpolicy_vpnglobal_binding":<vpnsessionpolicy_vpnglobal_binding[]
_value>,

"vpnsessionpolicy_vpnvserver_binding":<vpnsessionpolicy_vpnvserver_binding
[]_value>,

"vpnsessionpolicy_aaagroup_binding":<vpnsessionpolicy_aaagroup_binding[]_v
alue>

}]}
```

vpnsessionpolicy_vpnglobal_binding

Binding object showing the vpnglobal that can be bound to vpnsessionpolicy.

Properties

(click to see Operations)

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
boundto	<String>	Read-write	The entity name to which policy is bound.
name	<String>	Read-write	Name of the session policy to display. Minimum length = 1
priority	<Double>	Read-only	.
activepolicy	<Double>	Read-only	.
__count	<Double>	Read-only	count parameter

Operations

(click to see Properties)

GET| GET (ALL)| COUNT

Some options that you can use for each operations:

- **Getting warnings in response:**NITRO allows you to get warnings in an operation by specifying the "warning" query parameter as "yes". For example, to get warnings while connecting to the Citrix ADC appliance, the URL is as follows:

  http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/login?warning=yes

  If any, the warnings are displayed in the response payload with the HTTP code "209 X-NITRO-WARNING".

- **Authenticated access for individual NITRO operations:**NITRO allows you to logon to the Citrix ADC appliance to perform individual operations. You can use this option instead of creating a NITRO session (using the login object) and then using that session to perform all operations,

  To do this, you must specify the username and password in the request header of the NITRO request as follows:

  X-NITRO-USER:<username>

  X-NITRO-PASS:<password>

Note: In such cases, make sure that the request header DOES not include the following:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue>

`*Note:` *

Mandatory parameters are marked in redand placeholder content is marked in <green>.

get

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/vpnsessionpolicy_vpnglobal_binding/name_value<String>

Query-parameters:

filter

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/vpnsessionpolicy_vpnglobal_binding/name_value<String>?**filter=property-name1:property-value1,property-name2:property-value2**

Use this query-parameter to get the filtered set of vpnsessionpolicy_vpnglobal_binding resources configured on Citrix ADC. Filtering can be done on any of the properties of the resource.

pagination

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/vpnsessionpolicy_vpnglobal_binding/name_value<String>?**pagesize=#no;pageno=#no**

Use this query-parameter to get the vpnsessionpolicy_vpnglobal_binding resources in chunks.

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "vpnsessionpolicy_vpnglobal_binding": [ {

"boundto":<String_value>,

"name":<String_value>,

"priority":<Double_value>,

"activepolicy":<Double_value>

}]}
```

get (all)

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/vpnsessionpolicy_vpnglobal_binding

Query-parameters:

bulkbindings

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/vpnsessionpolicy_vpnglobal_binding?**bulkbindings=yes**

NITRO allows you to fetch bindings in bulk.

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "vpnsessionpolicy_vpnglobal_binding": [ {

"boundto":<String_value>,

"name":<String_value>,

"priority":<Double_value>,

"activepolicy":<Double_value>

}]}
```

count

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/vpnsessionpolicy_vpnglobal_binding/name_value<String>?**count=yes**

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{"vpnsessionpolicy_vpnglobal_binding": [ { "__count": "#no"} ] }
```

vpnsessionpolicy_vpnvserver_binding

Binding object showing the vpnvserver that can be bound to vpnsessionpolicy.

Properties

(click to see Operations)

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
boundto	<String>	Read-write	The entity name to which policy is bound.
name	<String>	Read-write	Name of the session policy to display. Minimum length = 1
priority	<Double>	Read-only	.
activepolicy	<Double>	Read-only	.
__count	<Double>	Read-only	count parameter

Operations

(click to see Properties)

GET| GET (ALL)| COUNT

Some options that you can use for each operations:

- **Getting warnings in response:**NITRO allows you to get warnings in an operation by specifying the "warning" query parameter as "yes". For example, to get warnings while connecting to the Citrix ADC appliance, the URL is as follows:

  http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/login?warning=yes

  If any, the warnings are displayed in the response payload with the HTTP code "209 X-NITRO-WARNING".

- **Authenticated access for individual NITRO operations:**NITRO allows you to logon to the Citrix ADC appliance to perform individual operations. You can use this option instead of

creating a NITRO session (using the login object) and then using that session to perform all operations,

To do this, you must specify the username and password in the request header of the NITRO request as follows:

X-NITRO-USER:<username>

X-NITRO-PASS:<password>

Note: In such cases, make sure that the request header DOES not include the following:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue>

`*Note:`*

Mandatory parameters are marked in redand placeholder content is marked in <green>.

get

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/vpnsessionpolicy_vpnvserver_binding/name_value<String>

Query-parameters:

filter

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/vpnsessionpolicy_vpnvserver_binding/name_value<String>?**filter=property-name1:property-value1,property-name2:property-value2**

Use this query-parameter to get the filtered set of vpnsessionpolicy_vpnvserver_binding resources configured on Citrix ADC. Filtering can be done on any of the properties of the resource.

pagination

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/vpnsessionpolicy_vpnvserver_binding/name_value<String>?**pagesize=#no;pageno=#no**

Use this query-parameter to get the vpnsessionpolicy_vpnvserver_binding resources in chunks.

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "vpnsessionpolicy_vpnvserver_binding": [ {

"boundto":<String_value>,

"name":<String_value>,

"priority":<Double_value>,

"activepolicy":<Double_value>

}]}
```

get (all)

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/vpnsessionpolicy_vpnvserver_binding

Query-parameters:

bulkbindings

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/vpnsessionpolicy_vpnvserver_binding?**bulkbindings=yes**

NITRO allows you to fetch bindings in bulk.

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "vpnsessionpolicy_vpnvserver_binding": [ {

"boundto":<String_value>,

"name":<String_value>,

"priority":<Double_value>,

"activepolicy":<Double_value>

}]}
```

count

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/vpnsessionpolicy_vpnvserver_binding/name_value<String>?**count=yes**

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{"vpnsessionpolicy_vpnvserver_binding": [ { "__count": "#no"} ] }
```

vpnsfconfig

Configuration for Create a StoreFront configuration file resource.

Properties

(click to see Operations)

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
vserver	<String[]>	Read-write	Name of Gateway virtual server.
filename	<String>	Read-only	Path where file was created.
__count	<Double>	Read-only	count parameter

Operations

(click to see Properties)

GET (ALL)| COUNT

Some options that you can use for each operations:

- **Getting warnings in response:**NITRO allows you to get warnings in an operation by specifying the "warning" query parameter as "yes". For example, to get warnings while connecting to the Citrix ADC appliance, the URL is as follows:

  http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/login?warning=yes

  If any, the warnings are displayed in the response payload with the HTTP code "209 X-NITRO-WARNING".

- **Authenticated access for individual NITRO operations:**NITRO allows you to logon to the Citrix ADC appliance to perform individual operations. You can use this option instead of

creating a NITRO session (using the login object) and then using that session to perform all operations,

To do this, you must specify the username and password in the request header of the NITRO request as follows:

X-NITRO-USER:<username>

X-NITRO-PASS:<password>

Note: In such cases, make sure that the request header DOES not include the following:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue>

`*Note:`*

Mandatory parameters are marked in redand placeholder content is marked in <green>.

get (all)

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/vpnsfconfig

Query-parameters:

args

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/vpnsfconfig?**args=vserver:<String_value>**

Use this query-parameter to get vpnsfconfig resources based on additional properties.

attrs

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/vpnsfconfig?**attrs=property-name1,property-name2**

Use this query parameter to specify the resource details that you want to retrieve.

filter

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/vpnsfconfig?**filter=property-name1:property-val1,property-name2:property-val2**

Use this query-parameter to get the filtered set of vpnsfconfig resources configured on Citrix ADC. Filtering can be done on any of the properties of the resource.

view

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/vpnsfconfig?**view=summary**

Note: By default, the retrieved results are displayed in detail view (?view=detail).

pagination

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/vpnsfconfig?**pagesize=#no;pageno=#no**

Use this query-parameter to get the vpnsfconfig resources in chunks.

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "vpnsfconfig": [ {

vserver:<String[]_value>"filename":<String_value>

}]}
```

count

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/vpnsfconfig?**count=yes**

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "vpnsfconfig": [ { "__count": "#no"} ] }
```

vpnstoreinfo

Configuration for Store information for a URL resource.

Properties

(click to see Operations)

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
url	<String>	Read-write	StoreFront URL to be scanned.
storeserverstatus	<String>	Read-only	Availability of the server for TCP connections. Possible values = UP, DOWN
storeserverissf	<String>	Read-only	Indicates if the server being scanned is running StoreFront. Possible values = YES, NO

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
storeapisupport	<String>	Read-only	Indicates StoreFront API support status. Possible values = YES, NO
storelist	<String>	Read-only	List of available stores return by StoreFront API.
storestatus	<String>	Read-only	Availability of the specified StoreFront store. Possible values = UP, DOWN
__count	<Double>	Read-only	count parameter

Operations

(click to see Properties)

GET (ALL)| COUNT

Some options that you can use for each operations:

- **Getting warnings in response:**NITRO allows you to get warnings in an operation by specifying the "warning" query parameter as "yes". For example, to get warnings while connecting to the Citrix ADC appliance, the URL is as follows:

  http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/login?warning=yes

  If any, the warnings are displayed in the response payload with the HTTP code "209 X-NITRO-WARNING".

- **Authenticated access for individual NITRO operations:**NITRO allows you to logon to the Citrix ADC appliance to perform individual operations. You can use this option instead of creating a NITRO session (using the login object) and then using that session to perform all operations,

  To do this, you must specify the username and password in the request header of the NITRO request as follows:

X-NITRO-USER:<username>

X-NITRO-PASS:<password>

Note: In such cases, make sure that the request header DOES not include the following:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue>

`*Note:`*

Mandatory parameters are marked in redand placeholder content is marked in <green>.

get (all)

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/vpnstoreinfo

Query-parameters:

args

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/vpnstoreinfo?**args=url:<String_value>,**

Use this query-parameter to get vpnstoreinfo resources based on additional properties.

attrs

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/vpnstoreinfo?**attrs=property-name1,property-name2**

Use this query parameter to specify the resource details that you want to retrieve.

filter

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/vpnstoreinfo?**filter=property-name1:property-val1,property-name2:property-val2**

Use this query-parameter to get the filtered set of vpnstoreinfo resources configured on Citrix ADC. Filtering can be done on any of the properties of the resource.

view

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/vpnstoreinfo?**view=summary**

Note: By default, the retrieved results are displayed in detail view (?view=detail).

pagination

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/vpnstoreinfo?**pagesize=#no;pageno=#no**

Use this query-parameter to get the vpnstoreinfo resources in chunks.

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "vpnstoreinfo": [ {

<b>url:<String_value>,</b>"storeserverstatus":<String_value>,

"storeserverissf":<String_value>,

"storeapisupport":<String_value>,

"storelist":<String_value>,

"storestatus":<String_value>

}]}
```

count

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/vpnstoreinfo?**args=url:<String_value>,;count=yes**

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "vpnstoreinfo": [ { "__count": "#no"} ] }
```

vpntrafficaction

Configuration for VPN traffic action resource.

Properties

(click to see Operations)

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
name	<String>	Read-write	Name for the traffic action. Must begin with an ASCII alphabetic or underscore (_) character, and must contain only ASCII alphanumeric, underscore, hash (#), period (.), space, colon (:), at (@), equals (=), and hyphen (-) characters. Cannot

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
			be changed after a traffic action is created. The following requirement applies only to the Citrix ADC CLI: If the name includes one or more spaces, enclose the name in double or single quotation marks (for example, "my action" or 'my action'). Minimum length = 1
qual	<String>	Read-write	Protocol, either HTTP or TCP, to be used with the action. Possible values = http, tcp
apptimeout	<Double>	Read-write	Maximum amount of time, in minutes, a user can stay logged on to the web application. Minimum value = 1 Maximum value = 715827
sso	<String>	Read-write	Provide single sign-on to the web application.

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
			NOTE : Authentication mechanisms like Basic-authentication require the user credentials to be sent in plaintext which is not secure if the server is running on HTTP (instead of HTTPS). Possible values = ON, OFF
hdx	<String>	Read-write	Provide hdx proxy to the ICA traffic. Possible values = ON, OFF
formssoaction	<String>	Read-write	Name of the form-based single sign-on profile. Form-based single sign-on allows users to log on one time to all protected applications in your network, instead of requiring them to log on separately to access each one.
fta	<String>	Read-write	Specify file type association, which is a list of file extensions that users are allowed to open. Possible values = ON, OFF

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
wanscaler	<String>	Read-write	Use the Repeater Plug-in to optimize network traffic. Possible values = ON, OFF
kcdaccount	<String>	Read-write	Kerberos constrained delegation account name. Default value: "Default" Minimum length = 1 Maximum length = 32
samlssoprofile	<String>	Read-write	Profile to be used for doing SAML SSO to remote relying party. Minimum length = 1
proxy	<String>	Read-write	IP address and Port of the proxy server to be used for HTTP access for this request. Minimum length = 1
userexpression	<String>	Read-write	expression that will be evaluated to obtain username for SingleSignOn. Maximum length = 256
passwdexpression	<String>	Read-write	expression that will be evaluated to

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
			obtain password for SingleSignOn. Maximum length = 256
__count	<Double>	Read-only	count parameter

Operations

(click to see Properties)

ADD| DELETE| UPDATE| UNSET| GET (ALL)| GET| COUNT

Some options that you can use for each operations:

- **Getting warnings in response:**NITRO allows you to get warnings in an operation by specifying the "warning" query parameter as "yes". For example, to get warnings while connecting to the Citrix ADC appliance, the URL is as follows:

  http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/login?warning=yes

  If any, the warnings are displayed in the response payload with the HTTP code "209 X-NITRO-WARNING".

- **Authenticated access for individual NITRO operations:**NITRO allows you to logon to the Citrix ADC appliance to perform individual operations. You can use this option instead of creating a NITRO session (using the login object) and then using that session to perform all operations,

  To do this, you must specify the username and password in the request header of the NITRO request as follows:

  X-NITRO-USER:<username>

  X-NITRO-PASS:<password>

  Note: In such cases, make sure that the request header DOES not include the following:

  Cookie:NITRO_AUTH_TOKEN=<tokenvalue>

`*Note:` *

Mandatory parameters are marked in redand placeholder content is marked in <green>.

add

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/vpntrafficaction

HTTP Method:POST

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Content-Type:application/json

Request Payload:

```
{"vpntrafficaction":{

<b>"name":<String_value>,

</b><b>"qual":<String_value>,

</b>"apptimeout":<Double_value>,

"sso":<String_value>,

"hdx":<String_value>,

"formssoaction":<String_value>,

"fta":<String_value>,

"wanscaler":<String_value>,

"kcdaccount":<String_value>,

"samlssoprofile":<String_value>,

"proxy":<String_value>,

"userexpression":<String_value>,

"passwdexpression":<String_value>

}}
```

Response:

HTTP Status Code on Success: 201 Created HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error

delete

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/vpntrafficaction/name_value<String>

HTTP Method:DELETE

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue>

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error

update

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/vpntrafficaction

HTTP Method:PUT

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Content-Type:application/json

Request Payload:

```
{"vpntrafficaction":{

<b>"name":<String_value>,

</b>"apptimeout":<Double_value>,

"sso":<String_value>,
```

```
"hdx":<String_value>,

"formssoaction":<String_value>,

"fta":<String_value>,

"wanscaler":<String_value>,

"kcdaccount":<String_value>,

"samlssoprofile":<String_value>,

"proxy":<String_value>,

"userexpression":<String_value>,

"passwdexpression":<String_value>

}}
```

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error

unset

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/vpntrafficaction?**action=unset**

HTTP Method:POST

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Content-Type:application/json

Request Payload:

```
{"vpntrafficaction":{
```

```
<b>"name":<String_value>,

</b>"wanscaler":true,

"kcdaccount":true,

"proxy":true,

"userexpression":true,

"passwdexpression":true

}}
```

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error

get (all)

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/vpntrafficaction

Query-parameters:

attrs

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/vpntrafficaction?**attrs=property-name1,property-name2**

Use this query parameter to specify the resource details that you want to retrieve.

filter

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/vpntrafficaction?**filter=property-name1:property-val1,property-name2:property-val2**

Use this query-parameter to get the filtered set of vpntrafficaction resources configured on Citrix ADC. Filtering can be done on any of the properties of the resource.

view

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/vpntrafficaction?**view=summary**

Note: By default, the retrieved results are displayed in detail view (?view=detail).

pagination

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/vpntrafficaction?**pagesize=#no;pageno=#no**

Use this query-parameter to get the vpntrafficaction resources in chunks.

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "vpntrafficaction": [ {

"name":<String_value>,

"qual":<String_value>,

"apptimeout":<Double_value>,

"sso":<String_value>,

"formssoaction":<String_value>,

"hdx":<String_value>,

"fta":<String_value>,

"wanscaler":<String_value>,
```

```
"kcdaccount":<String_value>,

"samlssoprofile":<String_value>,

"proxy":<String_value>,

"userexpression":<String_value>,

"passwdexpression":<String_value>

}]}
```

get

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/vpntrafficaction/name_value<String>

Query-parameters:

attrs

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/vpntrafficaction/name_value<String>?**attrs=property-name1,property-name2**

Use this query parameter to specify the resource details that you want to retrieve.

view

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/vpntrafficaction/name_value<String>?**view=summary**

Note: By default, the retrieved results are displayed in detail view (?view=detail).

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "vpntrafficaction": [ {

"name":<String_value>,

"qual":<String_value>,

"apptimeout":<Double_value>,

"sso":<String_value>,

"formssoaction":<String_value>,

"hdx":<String_value>,

"fta":<String_value>,

"wanscaler":<String_value>,

"kcdaccount":<String_value>,

"samlssoprofile":<String_value>,

"proxy":<String_value>,

"userexpression":<String_value>,

"passwdexpression":<String_value>

}]}
```

count

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/vpntrafficaction?**count=yes**

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "vpntrafficaction": [ { "__count": "#no"} ] }
```

vpntrafficpolicy

Configuration for VPN traffic policy resource.

Properties

(click to see Operations)

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
name	<String>	Read-write	Name for the traffic policy. Must begin with an ASCII alphabetic or underscore (_) character, and must contain only ASCII alphanumeric, underscore, hash (#), period (.), space, colon (:), at (@), equals (=), and hyphen (-) characters. Cannot be changed after the policy is created.

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
			The following requirement applies only to the Citrix ADC CLI: If the name includes one or more spaces, enclose the name in double or single quotation marks (for example, "my policy" or 'my policy'). Minimum length = 1
rule	<String>	Read-write	Expression, or name of a named expression, against which traffic is evaluated. The following requirements apply only to the Citrix ADC CLI: * If the expression includes one or more spaces, enclose the entire expression in double quotation marks. * If the expression itself includes double quotation marks, escape the

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
			quotations by using the \ character. * Alternatively, you can use single quotation marks to enclose the rule, in which case you do not have to escape the double quotation marks.
action	<String>	Read-write	Action to apply to traffic that matches the policy. Minimum length = 1
expressiontype	<String>	Read-only	Type of policy (Classic/Advanced). Possible values = Classic Policy, Advanced Policy
hits	<Double>	Read-only	Number of hits.
__count	<Double>	Read-only	count parameter

Operations

(click to see Properties)

ADD| DELETE| UPDATE| UNSET| GET (ALL)| GET| COUNT

Some options that you can use for each operations:

- **Getting warnings in response:**NITRO allows you to get warnings in an operation by specifying the "warning" query parameter as "yes". For example, to get warnings while connecting to the Citrix ADC appliance, the URL is as follows:

  http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/login?warning=yes

If any, the warnings are displayed in the response payload with the HTTP code "209 X-NITRO-WARNING".

- **Authenticated access for individual NITRO operations:**NITRO allows you to logon to the Citrix ADC appliance to perform individual operations. You can use this option instead of creating a NITRO session (using the login object) and then using that session to perform all operations,

  To do this, you must specify the username and password in the request header of the NITRO request as follows:

  X-NITRO-USER:<username>

  X-NITRO-PASS:<password>

  Note: In such cases, make sure that the request header DOES not include the following:

  Cookie:NITRO_AUTH_TOKEN=<tokenvalue>

`*Note:`*

Mandatory parameters are marked in redand placeholder content is marked in <green>.

add

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/vpntrafficpolicy

HTTP Method:POST

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Content-Type:application/json

Request Payload:

```
{"vpntrafficpolicy":{

<b>"name":<String_value>,

</b><b>"rule":<String_value>,

</b><b>"action":<String_value>
```

```
</b>}}
```

Response:

HTTP Status Code on Success: 201 Created HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error

delete

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/vpntrafficpolicy/name_value<String>

HTTP Method:DELETE

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue>

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error

update

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/vpntrafficpolicy

HTTP Method:PUT

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Content-Type:application/json

Request Payload:

```
{"vpntrafficpolicy":{

<b>"name":<String_value>,

</b>"rule":<String_value>,
```

```
"action":<String_value>

}}
```

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error

unset

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/vpntrafficpolicy?**action=unset**

HTTP Method:POST

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Content-Type:application/json

Request Payload:

```
{"vpntrafficpolicy":{

<b>"name":<String_value>,

</b>"rule":true,

"action":true

}}
```

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error

get (all)

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/vpntrafficpolicy

Query-parameters:

attrs

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/vpntrafficpolicy?**attrs=property-name1,property-name2**

Use this query parameter to specify the resource details that you want to retrieve.

filter

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/vpntrafficpolicy?**filter=property-name1:property-val1,property-name2:property-val2**

Use this query-parameter to get the filtered set of vpntrafficpolicy resources configured on Citrix ADC. Filtering can be done on any of the properties of the resource.

view

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/vpntrafficpolicy?**view=summary**

Note: By default, the retrieved results are displayed in detail view (?view=detail).

pagination

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/vpntrafficpolicy?**pagesize=#no;pageno=#no**

Use this query-parameter to get the vpntrafficpolicy resources in chunks.

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "vpntrafficpolicy": [ {

"name":<String_value>,

"rule":<String_value>,

"action":<String_value>,

"expressiontype":<String_value>,

"hits":<Double_value>

}]}
```

get

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/vpntrafficpolicy/name_value<String>

Query-parameters:

attrs

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/vpntrafficpolicy/name_value<String>?**attrs=property-name1,property-name2**

Use this query parameter to specify the resource details that you want to retrieve.

view

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/vpntrafficpolicy/name_value<String>?**view=summary**

Note: By default, the retrieved results are displayed in detail view (?view=detail).

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "vpntrafficpolicy": [ {

"name":<String_value>,

"rule":<String_value>,

"action":<String_value>,

"expressiontype":<String_value>,

"hits":<Double_value>

}]}
```

count

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/vpntrafficpolicy?**count=yes**

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "vpntrafficpolicy": [ { "__count": "#no"} ] }
```

vpntrafficpolicy_aaagroup_binding

Binding object showing the aaagroup that can be bound to vpntrafficpolicy.

Properties

(click to see Operations)

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
name	<String>	Read-write	Name of the traffic policy for which to display detailed information. Minimum length = 1
boundto	<String>	Read-write	The entity name to which policy is bound.
priority	<Double>	Read-only	.
activepolicy	<Double>	Read-only	.
__count	<Double>	Read-only	count parameter

Operations

(click to see Properties)

GET| GET (ALL)| COUNT

Some options that you can use for each operations:

- **Getting warnings in response:**NITRO allows you to get warnings in an operation by specifying the "warning" query parameter as "yes". For example, to get warnings while connecting to the Citrix ADC appliance, the URL is as follows:

  http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/login?warning=yes

  If any, the warnings are displayed in the response payload with the HTTP code "209 X-NITRO-WARNING".

- **Authenticated access for individual NITRO operations:**NITRO allows you to logon to the Citrix ADC appliance to perform individual operations. You can use this option instead of creating a NITRO session (using the login object) and then using that session to perform all operations,

  To do this, you must specify the username and password in the request header of the NITRO request as follows:

  X-NITRO-USER:<username>

  X-NITRO-PASS:<password>

  Note: In such cases, make sure that the request header DOES not include the following:

  Cookie:NITRO_AUTH_TOKEN=<tokenvalue>

`*Note:`*

Mandatory parameters are marked in redand placeholder content is marked in <green>.

get

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/vpntrafficpolicy_aaagroup_binding/name_value<String>

Query-parameters:

filter

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/vpntrafficpolicy_aaagroup_binding/name_value<String>?**filter=property-name1:property-value1,property-name2:property-value2**

Use this query-parameter to get the filtered set of vpntrafficpolicy_aaagroup_binding resources configured on Citrix ADC. Filtering can be done on any of the properties of the resource.

pagination

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/vpntrafficpolicy_aaagroup_binding/name_value<String>?**pagesize=#no;pageno=#no**

Use this query-parameter to get the vpntrafficpolicy_aaagroup_binding resources in chunks.

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "vpntrafficpolicy_aaagroup_binding": [ {

"name":<String_value>,

"boundto":<String_value>,

"priority":<Double_value>,

"activepolicy":<Double_value>

}]}
```

get (all)

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/vpntrafficpolicy_aaagroup_binding

Query-parameters:

bulkbindings

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/vpntrafficpolicy_aaagroup_binding?**bulkbindings=yes**

NITRO allows you to fetch bindings in bulk.

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "vpntrafficpolicy_aaagroup_binding": [ {

"name":<String_value>,

"boundto":<String_value>,

"priority":<Double_value>,

"activepolicy":<Double_value>

}]}
```

count

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/vpntrafficpolicy_aaagroup_binding/name_value<String>?**count=yes**

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{"vpntrafficpolicy_aaagroup_binding": [ { "__count": "#no"} ] }
```

vpntrafficpolicy_aaauser_binding

Binding object showing the aaauser that can be bound to vpntrafficpolicy.

Properties

(click to see Operations)

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
name	<String>	Read-write	Name of the traffic policy for which to display detailed information. Minimum length = 1
boundto	<String>	Read-write	The entity name to which policy is bound.
priority	<Double>	Read-only	.
activepolicy	<Double>	Read-only	.
__count	<Double>	Read-only	count parameter

Operations

(click to see Properties)

GET| GET (ALL)| COUNT

Some options that you can use for each operations:

- **Getting warnings in response:**NITRO allows you to get warnings in an operation by specifying the "warning" query parameter as "yes". For example, to get warnings while connecting to the Citrix ADC appliance, the URL is as follows:

  http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/login?warning=yes

  If any, the warnings are displayed in the response payload with the HTTP code "209 X-NITRO-WARNING".

- **Authenticated access for individual NITRO operations:**NITRO allows you to logon to the Citrix ADC appliance to perform individual operations. You can use this option instead of creating a NITRO session (using the login object) and then using that session to perform all operations,

  To do this, you must specify the username and password in the request header of the NITRO request as follows:

  X-NITRO-USER:<username>

  X-NITRO-PASS:<password>

  Note: In such cases, make sure that the request header DOES not include the following:

  Cookie:NITRO_AUTH_TOKEN=<tokenvalue>

`*Note:`*

Mandatory parameters are marked in redand placeholder content is marked in <green>.

get

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/vpntrafficpolicy_aaauser_binding/name_value<String>

Query-parameters:

filter

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/vpntrafficpolicy_aaauser_binding/name_value<String>?**filter=property-name1:property-value1,property-name2:property-value2**

Use this query-parameter to get the filtered set of vpntrafficpolicy_aaauser_binding resources configured on Citrix ADC. Filtering can be done on any of the properties of the resource.

pagination

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/vpntrafficpolicy_aaauser_binding/name_value<String>?**pagesize=#no;pageno=#no**

Use this query-parameter to get the vpntrafficpolicy_aaauser_binding resources in chunks.

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "vpntrafficpolicy_aaauser_binding": [ {

"name":<String_value>,

"boundto":<String_value>,

"priority":<Double_value>,

"activepolicy":<Double_value>

}]}
```

get (all)

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/vpntrafficpolicy_aaauser_binding

Query-parameters:

bulkbindings

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/vpntrafficpolicy_aaauser_binding?**bulkbindings=yes**

NITRO allows you to fetch bindings in bulk.

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "vpntrafficpolicy_aaauser_binding": [ {

"name":<String_value>,

"boundto":<String_value>,

"priority":<Double_value>,

"activepolicy":<Double_value>

}]}
```

count

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/vpntrafficpolicy_aaauser_binding/name_value<String>?**count=yes**

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{"vpntrafficpolicy_aaauser_binding": [ { "__count": "#no"} ] }
```

vpntrafficpolicy_binding

Binding object which returns the resources bound to vpntrafficpolicy.

Properties

(click to see Operations)

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
name	<String>	Read-write	Name of the traffic policy for which to display detailed information. Minimum length = 1

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
vpntrafficpolicy_aaauser_binding	<vpntrafficpolicy_aaauser_binding[]>	Read-only	aaauser that can be bound to vpntrafficpolicy.
vpntrafficpolicy_vpnglobal_binding	<vpntrafficpolicy_vpnglobal_binding[]>	Read-only	vpnglobal that can be bound to vpntrafficpolicy.
vpntrafficpolicy_vpnvserver_binding	<vpntrafficpolicy_vpnvserver_binding[]>	Read-only	vpnvserver that can be bound to vpntrafficpolicy.
vpntrafficpolicy_aaagroup_binding	<vpntrafficpolicy_aaagroup_binding[]>	Read-only	aaagroup that can be bound to vpntrafficpolicy.

Operations

(click to see Properties)

GET| GET (ALL)

Some options that you can use for each operations:

- **Getting warnings in response:**NITRO allows you to get warnings in an operation by specifying the "warning" query parameter as "yes". For example, to get warnings while connecting to the Citrix ADC appliance, the URL is as follows:

  http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/login?warning=yes

  If any, the warnings are displayed in the response payload with the HTTP code "209 X-NITRO-WARNING".

- **Authenticated access for individual NITRO operations:**NITRO allows you to logon to the Citrix ADC appliance to perform individual operations. You can use this option instead of creating a NITRO session (using the login object) and then using that session to perform all operations,

  To do this, you must specify the username and password in the request header of the NITRO request as follows:

  X-NITRO-USER:<username>

X-NITRO-PASS:<password>

Note: In such cases, make sure that the request header DOES not include the following:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue>

`*Note:`*

Mandatory parameters are marked in redand placeholder content is marked in <green>.

get

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/vpntrafficpolicy_binding/name_value<String>

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "vpntrafficpolicy_binding": [ {

"name":<String_value>,

"vpntrafficpolicy_aaagroup_binding":<vpntrafficpolicy_aaagroup_binding[]_value>,

"vpntrafficpolicy_aaauser_binding":<vpntrafficpolicy_aaauser_binding[]_value>,

"vpntrafficpolicy_vpnvserver_binding":<vpntrafficpolicy_vpnvserver_binding[]_value>,
```

```
"vpntrafficpolicy_vpnglobal_binding":<vpntrafficpolicy_vpnglobal_binding[]
_value>

}]}
```

get (all)

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/vpntrafficpolicy_binding

Query-parameters:

bulkbindings

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/vpntrafficpolicy_binding?**bulkbindings=yes**

NITRO allows you to fetch bindings in bulk.

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "vpntrafficpolicy_binding": [ {

"name":<String_value>,

"vpntrafficpolicy_aaagroup_binding":<vpntrafficpolicy_aaagroup_binding[]_v
alue>,

"vpntrafficpolicy_aaauser_binding":<vpntrafficpolicy_aaauser_binding[]_val
```

```
ue>,

"vpntrafficpolicy_vpnvserver_binding":<vpntrafficpolicy_vpnvserver_binding
[]_value>,

"vpntrafficpolicy_vpnglobal_binding":<vpntrafficpolicy_vpnglobal_binding[]
_value>

}]}
```

vpntrafficpolicy_vpnglobal_binding

Binding object showing the vpnglobal that can be bound to vpntrafficpolicy.

Properties

(click to see Operations)

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
name	<String>	Read-write	Name of the traffic policy for which to display detailed information. Minimum length = 1
boundto	<String>	Read-write	The entity name to which policy is bound.
priority	<Double>	Read-only	.
activepolicy	<Double>	Read-only	.
__count	<Double>	Read-only	count parameter

Operations

(click to see Properties)

GET| GET (ALL)| COUNT

Some options that you can use for each operations:

- **Getting warnings in response:**NITRO allows you to get warnings in an operation by specifying the "warning" query parameter as "yes". For example, to get warnings while connecting to the Citrix ADC appliance, the URL is as follows:

  http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/login?warning=yes

  If any, the warnings are displayed in the response payload with the HTTP code "209 X-NITRO-WARNING".

- **Authenticated access for individual NITRO operations:**NITRO allows you to logon to the Citrix ADC appliance to perform individual operations. You can use this option instead of creating a NITRO session (using the login object) and then using that session to perform all operations,

  To do this, you must specify the username and password in the request header of the NITRO request as follows:

  X-NITRO-USER:<username>

  X-NITRO-PASS:<password>

  Note: In such cases, make sure that the request header DOES not include the following:

  Cookie:NITRO_AUTH_TOKEN=<tokenvalue>

`*Note:`*

Mandatory parameters are marked in redand placeholder content is marked in <green>.

get

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/vpntrafficpolicy_vpnglobal_binding/name_value<String>

Query-parameters:

filter

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/vpntrafficpolicy_vpnglobal_binding/name_value<String>?**filter=property-name1:property-value1,property-name2:property-value2**

Use this query-parameter to get the filtered set of vpntrafficpolicy_vpnglobal_binding resources configured on Citrix ADC. Filtering can be done on any of the properties of the resource.

pagination

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/vpntrafficpolicy_vpnglobal_binding/name_value<String>?**pagesize=#no;pageno=#no**

Use this query-parameter to get the vpntrafficpolicy_vpnglobal_binding resources in chunks.

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "vpntrafficpolicy_vpnglobal_binding": [ {

"name":<String_value>,

"boundto":<String_value>,

"priority":<Double_value>,

"activepolicy":<Double_value>

}]}
```

get (all)

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/vpntrafficpolicy_vpnglobal_binding

Query-parameters:

bulkbindings

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/vpntrafficpolicy_vpnglobal_binding?**bulkbindings=yes**

NITRO allows you to fetch bindings in bulk.

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "vpntrafficpolicy_vpnglobal_binding": [ {

"name":<String_value>,

"boundto":<String_value>,

"priority":<Double_value>,

"activepolicy":<Double_value>

}]}
```

count

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/vpntrafficpolicy_vpnglobal_binding/name_value<String>?**count=yes**

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{"vpntrafficpolicy_vpnglobal_binding": [ { "__count": "#no"} ] }
```

vpntrafficpolicy_vpnvserver_binding

Binding object showing the vpnvserver that can be bound to vpntrafficpolicy.

Properties

(click to see Operations)

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
name	<String>	Read-write	Name of the traffic policy for which to display detailed information. Minimum length = 1
boundto	<String>	Read-write	The entity name to which policy is bound.
priority	<Double>	Read-only	.
activepolicy	<Double>	Read-only	.
__count	<Double>	Read-only	count parameter

Operations

(click to see Properties)

GET| GET (ALL)| COUNT

Some options that you can use for each operations:

- **Getting warnings in response:**NITRO allows you to get warnings in an operation by specifying the "warning" query parameter as "yes". For example, to get warnings while connecting to the Citrix ADC appliance, the URL is as follows:

  http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/login?warning=yes

  If any, the warnings are displayed in the response payload with the HTTP code "209 X-NITRO-WARNING".

- **Authenticated access for individual NITRO operations:**NITRO allows you to logon to the Citrix ADC appliance to perform individual operations. You can use this option instead of creating a NITRO session (using the login object) and then using that session to perform all operations,

  To do this, you must specify the username and password in the request header of the NITRO request as follows:

  X-NITRO-USER:<username>

  X-NITRO-PASS:<password>

  Note: In such cases, make sure that the request header DOES not include the following:

  Cookie:NITRO_AUTH_TOKEN=<tokenvalue>

`*Note:`*

Mandatory parameters are marked in redand placeholder content is marked in <green>.

get

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/vpntrafficpolicy_vpnvserver_binding/name_value<String>

Query-parameters:

filter

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/vpntrafficpolicy_vpnvserver_binding/name_value<String>?**filter=property-name1:property-value1,property-name2:property-value2**

Use this query-parameter to get the filtered set of vpntrafficpolicy_vpnvserver_binding resources configured on Citrix ADC. Filtering can be done on any of the properties of the resource.

pagination

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/vpntrafficpolicy_vpnvserver_binding/name_value<String>?**pagesize=#no;pageno=#no**

Use this query-parameter to get the vpntrafficpolicy_vpnvserver_binding resources in chunks.

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "vpntrafficpolicy_vpnvserver_binding": [ {

"name":<String_value>,

"boundto":<String_value>,

"priority":<Double_value>,

"activepolicy":<Double_value>

}]}
```

get (all)

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/vpntrafficpolicy_vpnvserver_binding

Query-parameters:

bulkbindings

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/vpntrafficpolicy_vpnvserver_binding?**bulkbindings=yes**

NITRO allows you to fetch bindings in bulk.

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "vpntrafficpolicy_vpnvserver_binding": [ {

"name":<String_value>,

"boundto":<String_value>,

"priority":<Double_value>,

"activepolicy":<Double_value>

}]}
```

count

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/vpntrafficpolicy_vpnvserver_binding/name_value<String>?**count=yes**

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{"vpntrafficpolicy_vpnvserver_binding": [ { "__count": "#no"} ] }
```

vpnurl

Configuration for VPN URL resource.

Properties

(click to see Operations)

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
urlname	<String>	Read-write	Name of the bookmark link. Minimum length = 1
linkname	<String>	Read-write	Description of the bookmark link. The description appears

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
			in the Access Interface. Minimum length = 1
actualurl	<String>	Read-write	Web address for the bookmark link. Minimum length = 1
vservername	<String>	Read-write	Name of the associated LB/CS vserver.
clientlessaccess	<String>	Read-write	If clientless access to the resource hosting the link is allowed, also use clientless access for the bookmarked web address in the Secure Client Access based session. Allows single sign-on and other HTTP processing on Citrix Gateway for HTTPS resources. Default value: OFF Possible values = ON, OFF
comment	<String>	Read-write	Any comments associated with the bookmark link.
iconurl	<String>	Read-write	URL to fetch icon file for displaying this resource.
ssotype	<String>	Read-write	Single sign on type for unified gateway.

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
			Possible values = unifiedgateway, selfauth, samlauth
applicationtype	<String>	Read-write	The type of application this VPN URL represents. Possible values are CVPN/SaaS/VPN. Possible values = CVPN, VPN, SaaS
samlssoprofile	<String>	Read-write	Profile to be used for doing SAML SSO.
appjson	<String>	Read-write	To store the template details in the json format.
__count	<Double>	Read-only	count parameter

Operations

(click to see Properties)

ADD| DELETE| UPDATE| UNSET| GET (ALL)| GET| COUNT

Some options that you can use for each operations:

- **Getting warnings in response:**NITRO allows you to get warnings in an operation by specifying the "warning" query parameter as "yes". For example, to get warnings while connecting to the Citrix ADC appliance, the URL is as follows:

  http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/login?warning=yes

  If any, the warnings are displayed in the response payload with the HTTP code "209 X-NITRO-WARNING".

- **Authenticated access for individual NITRO operations:**NITRO allows you to logon to the Citrix ADC appliance to perform individual operations. You can use this option instead of

creating a NITRO session (using the login object) and then using that session to perform all operations,

To do this, you must specify the username and password in the request header of the NITRO request as follows:

X-NITRO-USER:<username>

X-NITRO-PASS:<password>

Note: In such cases, make sure that the request header DOES not include the following:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue>

`*Note:`*

Mandatory parameters are marked in redand placeholder content is marked in <green>.

add

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/vpnurl

HTTP Method:POST

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Content-Type:application/json

Request Payload:

```
{"vpnurl":{

<b>"urlname":<String_value>,

</b><b>"linkname":<String_value>,

</b><b>"actualurl":<String_value>,

</b>"vservername":<String_value>,

"clientlessaccess":<String_value>,

"comment":<String_value>,
```

```
"iconurl":<String_value>,

"ssotype":<String_value>,

"applicationtype":<String_value>,

"samlssoprofile":<String_value>,

"appjson":<String_value>

}}
```

Response:

HTTP Status Code on Success: 201 Created HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error

delete

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/vpnurl/urlname_value<String>

HTTP Method:DELETE

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue>

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error

update

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/vpnurl

HTTP Method:PUT

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Content-Type:application/json

Request Payload:

```
{"vpnurl":{

<b>"urlname":<String_value>,

</b>"linkname":<String_value>,

"actualurl":<String_value>,

"vservername":<String_value>,

"clientlessaccess":<String_value>,

"comment":<String_value>,

"iconurl":<String_value>,

"ssotype":<String_value>,

"applicationtype":<String_value>,

"samlssoprofile":<String_value>,

"appjson":<String_value>

}}
```

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error

unset

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/vpnurl?**action=unset**

HTTP Method:POST

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Content-Type:application/json

Request Payload:

```
{"vpnurl":{

<b>"urlname":<String_value>,

</b>"vservername":true,

"clientlessaccess":true,

"comment":true,

"iconurl":true,

"ssotype":true,

"applicationtype":true,

"samlssoprofile":true,

"appjson":true

}}
```

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error

get (all)

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/vpnurl

Query-parameters:

attrs

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/vpnurl?**attrs=property-name1,property-name2**

Use this query parameter to specify the resource details that you want to retrieve.

filter

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/vpnurl?**filter=property-name1:property-val1,property-name2:property-val2**

Use this query-parameter to get the filtered set of vpnurl resources configured on Citrix ADC. Filtering can be done on any of the properties of the resource.

view

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/vpnurl?**view=summary**

Note: By default, the retrieved results are displayed in detail view (?view=detail).

pagination

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/vpnurl?**pagesize=#no;pageno=#no**

Use this query-parameter to get the vpnurl resources in chunks.

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "vpnurl": [ {
```

```
"urlname":<String_value>,

"linkname":<String_value>,

"vservername":<String_value>,

"actualurl":<String_value>,

"clientlessaccess":<String_value>,

"comment":<String_value>,

"iconurl":<String_value>,

"ssotype":<String_value>,

"applicationtype":<String_value>,

"samlssoprofile":<String_value>,

"appjson":<String_value>

}]}
```

get

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/vpnurl/urlname_value<String>

Query-parameters:

attrs

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/vpnurl/urlname_value<String>?**attrs=property-name1,property-name2**

Use this query parameter to specify the resource details that you want to retrieve.

view

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/vpnurl/urlname_value<String>?**view=summary**

Note: By default, the retrieved results are displayed in detail view (?view=detail).

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "vpnurl": [ {

"urlname":<String_value>,

"linkname":<String_value>,

"vservername":<String_value>,

"actualurl":<String_value>,

"clientlessaccess":<String_value>,

"comment":<String_value>,

"iconurl":<String_value>,

"ssotype":<String_value>,

"applicationtype":<String_value>,

"samlssoprofile":<String_value>,

"appjson":<String_value>

}]}
```

count

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/vpnurl?**count=yes**

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "vpnurl": [ { "__count": "#no"} ] }
```

vpnvserver

Configuration for VPN virtual server resource.

Properties

(click to see Operations)

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
name	<String>	Read-write	Name for the Citrix Gateway virtual server. Must begin with an ASCII alphabetic or underscore (_) character, and must

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
			contain only ASCII alphanumeric, underscore, hash (#), period (.), space, colon (:), at (@), equals (=), and hyphen (-) characters. Can be changed after the virtual server is created. The following requirement applies only to the Citrix ADC CLI: If the name includes one or more spaces, enclose the name in double or single quotation marks (for example, "my server" or 'my server'). Minimum length = 1
servicetype	<String>	Read-write	Protocol used by the Citrix Gateway virtual server. Default value: SSL Possible values = SSL
ipv46	<String>	Read-write	IPv4 or IPv6 address of the Citrix Gateway virtual server. Usually a public IP address. User

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
			devices send connection requests to this IP address. Minimum length = 1
range	<Double>	Read-write	Range of Citrix Gateway virtual server IP addresses. The consecutively numbered range of IP addresses begins with the address specified by the IP Address parameter. In the configuration utility, select Network VServer to enter a range. Default value: 1 Minimum value = 1
port	<Integer>	Read-write	TCP port on which the virtual server listens. Range 1 - 65535 * in CLI is represented as 65535 in NITRO API
ipset	<String>	Read-write	The list of IPv4/IPv6 addresses bound to ipset would form a part of listening service on the current vpn vserver. Minimum length = 1

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
state	<String>	Read-write	State of the virtual server. If the virtual server is disabled, requests are not processed. Default value: ENABLED Possible values = ENABLED, DISABLED
authentication	<String>	Read-write	Require authentication for users connecting to Citrix Gateway. Default value: ON Possible values = ON, OFF
doublehop	<String>	Read-write	Use the Citrix Gateway appliance in a double-hop configuration. A double-hop deployment provides an extra layer of security for the internal network by using three firewalls to divide the DMZ into two stages. Such a deployment can have one appliance in the DMZ and one appliance in the secure network.

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
			Default value: DISABLED Possible values = ENABLED, DISABLED
maxaaausers	<Double>	Read-write	Maximum number of concurrent user sessions allowed on this virtual server. The actual number of users allowed to log on to this virtual server depends on the total number of user licenses.
icaonly	<String>	Read-write	- When set to ON, it implies Basic mode where the user can log on using either Citrix Receiver or a browser and get access to the published apps configured at the XenApp/XenDEsktop environment pointed out by the WIHome parameter. Users are not allowed to connect using the Citrix Gateway Plug-in and end point scans cannot be configured. Number of users that can log in and access the apps are not limited

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
			by the license in this mode. - When set to OFF, it implies Smart Access mode where the user can log on using either Citrix Receiver or a browser or a Citrix Gateway Plug-in. The admin can configure end point scans to be run on the client systems and then use the results to control access to the published apps. In this mode, the client can connect to the gateway in other client modes namely VPN and CVPN. Number of users that can log in and access the resources are limited by the CCU licenses in this mode. Default value: OFF Possible values = ON, OFF

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
icaproxysessionmigration	<String>	Read-write	This option determines if an existing ICA Proxy session is transferred when the user logs on from another device. Default value: OFF Possible values = ON, OFF
dtls	<String>	Read-write	This option starts/stops the turn service on the vserver. Default value: ON Possible values = ON, OFF
loginonce	<String>	Read-write	This option enables/disables seamless SSO for this Vserver. Default value: OFF Possible values = ON, OFF
advancedepa	<String>	Read-write	This option tells whether advanced EPA is enabled on this virtual server. Default value: OFF Possible values = ON, OFF
devicecert	<String>	Read-write	Indicates whether device certificate

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
			check as a part of EPA is on or off. Default value: OFF Possible values = ON, OFF
certkeynames	<String>	Read-write	Name of the certificate key that was bound to the corresponding SSL virtual server as the Certificate Authority for the device certificate. Minimum length = 1 Maximum length = 127
downstateflush	<String>	Read-write	Close existing connections when the virtual server is marked DOWN, which means the server might have timed out. Disconnecting existing connections frees resources and in certain cases speeds recovery of overloaded load balancing setups. Enable this setting on servers in which the connections can safely be closed when they are marked DOWN. Do not enable DOWN

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
			state flush on servers that must complete their transactions. Default value: ENABLED Possible values = ENABLED, DISABLED
listenpolicy	<String>	Read-write	String specifying the listen policy for the Citrix Gateway virtual server. Can be either a named expression or an expression. The Citrix Gateway virtual server processes only the traffic for which the expression evaluates to true. Default value: "none"
listenpriority	<Double>	Read-write	Integer specifying the priority of the listen policy. A higher number specifies a lower priority. If a request matches the listen policies of more than one virtual server, the virtual server whose listen policy has the highest priority (the lowest priority

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
			number) accepts the request. Default value: 101 Minimum value = 0 Maximum value = 100
tcpprofilename	<String>	Read-write	Name of the TCP profile to assign to this virtual server. Minimum length = 1 Maximum length = 127
httpprofilename	<String>	Read-write	Name of the HTTP profile to assign to this virtual server. Default value: "nshttp_default_strict_validation" Minimum length = 1 Maximum length = 127
comment	<String>	Read-write	Any comments associated with the virtual server.
appflowlog	<String>	Read-write	Log AppFlow records that contain standard NetFlow or IPFIX information, such as time stamps for the beginning and end of a flow, packet count, and byte count. Also

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
			log records that contain application-level information, such as HTTP web addresses, HTTP request methods and response status codes, server response time, and latency. Default value: ENABLED Possible values = ENABLED, DISABLED
icmpvsrresponse	<String>	Read-write	Criterion for responding to PING requests sent to this virtual server. If this parameter is set to ACTIVE, respond only if the virtual server is available. With the PASSIVE setting, respond even if the virtual server is not available. Default value: PASSIVE Possible values = PASSIVE, ACTIVE
rhistate	<String>	Read-write	A host route is injected according to the setting on the virtual servers.

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
			* If set to PASSIVE on all the virtual servers that share the IP address, the appliance always injects the hostroute. * If set to ACTIVE on all the virtual servers that share the IP address, the appliance injects even if one virtual server is UP. * If set to ACTIVE on some virtual servers and PASSIVE on the others, the appliance injects even if one virtual server set to ACTIVE is UP. Default value: PASSIVE Possible values = PASSIVE, ACTIVE
netprofile	<String>	Read-write	The name of the network profile. Minimum length = 1 Maximum length = 127

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
cginfrahomepageredirect	<String>	Read-write	When client requests ShareFile resources and Citrix Gateway detects that the user is unauthenticated or the user session has expired, disabling this option takes the user to the originally requested ShareFile resource after authentication (instead of taking the user to the default VPN home page). Default value: ENABLED Possible values = ENABLED, DISABLED
maxloginattempts	<Double>	Read-write	Maximum number of logon attempts. Minimum value = 1 Maximum value = 255
failedlogintimeout	<Double>	Read-write	Number of minutes an account will be locked if user exceeds maximum permissible attempts. Minimum value = 1
l2conn	<String>	Read-write	Use Layer 2 parameters (channel number, MAC address, and VLAN

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
			ID) in addition to the 4-tuple (<source IP>:<source port>::<destination IP>:<destination port>) that is used to identify a connection. Allows multiple TCP and non-TCP connections with the same 4-tuple to coexist on the Citrix ADC. Possible values = ON, OFF
deploymenttype	<String>	Read-write	. Default value: 5 Possible values = NONE, ICA_WEBINTERFACE, ICA_STOREFRONT, MOBILITY, WIONNS
rdpserverprofilename	<String>	Read-write	Name of the RDP server profile associated with the vserver. Minimum length = 1 Maximum length = 31
windowsepapluginupgrade	<String>	Read-write	Option to set plugin upgrade behaviour for Win.

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
			Possible values = Always, Essential, Never
linuxepapluginupgrade	<String>	Read-write	Option to set plugin upgrade behaviour for Linux. Possible values = Always, Essential, Never
macepapluginupgrade	<String>	Read-write	Option to set plugin upgrade behaviour for Mac. Possible values = Always, Essential, Never
logoutonsmartcardremoval	<String>	Read-write	Option to VPN plugin behavior when smartcard or its reader is removed. Default value: OFF Possible values = ON, OFF
userdomains	<String>	Read-write	List of user domains specified as comma seperated value.
authnprofile	<String>	Read-write	Authentication Profile entity on virtual server. This entity can be used to offload authentication to AAA vserver for multi-factor(nFactor) authentication.

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
vserverfqdn	<String>	Read-write	Fully qualified domain name for a VPN virtual server. This is used during StoreFront configuration generation.
pcoipvserverprofilename	<String>	Read-write	Name of the PCoIP vserver profile associated with the vserver. Minimum length = 1 Maximum length = 31
newname	<String>	Read-write	New name for the Citrix Gateway virtual server. Must begin with an ASCII alphabetic or underscore (_) character, and must contain only ASCII alphanumeric, underscore, hash (#), period (.), space, colon (:), at (@), equals (=), and hyphen (-) characters. The following requirement applies only to the Citrix ADC CLI: If the name includes one or more spaces, enclose the

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
			name in double or single quotation marks (for example, "my server" or 'my server'). Minimum length = 1
ip	<String>	Read-only	The Virtual IP address of the VPN virtual server.
value	<String>	Read-only	Indicates whether or not the certificate is bound or if SSL offload is disabled. Possible values = Certkey not bound, SSL feature disabled
type	<String>	Read-only	The type of virtual server; for example, CONTENT based or ADDRESS based. Possible values = CONTENT, ADDRESS
curstate	<String>	Read-only	The current state of the virtual server, as UP, DOWN, BUSY, and so on. Possible values = UP, DOWN, UNKNOWN, BUSY, OUT OF SERVICE, GOING OUT OF SERVICE, DOWN

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
			WHEN GOING OUT OF SERVICE, NS_EMPTY_STR, Unknown, DISABLED
status	<Integer>	Read-only	Whether or not this virtual server responds to ARPs and whether or not round-robin selection is temporarily in effect.
cachetype	<String>	Read-only	Virtual server cache type. The options are: TRANSPARENT, REVERSE, and FORWARD. Possible values = TRANSPARENT, REVERSE, FORWARD
redirect	<String>	Read-only	The cache redirect policy. The valid redirect policies are: l. CACHE - Directs all requests to the cache. 2. POLICY - Applies cache redirection policy to determine whether the request should be directed

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
			to the cache or origin. This is the default setting. 3. ORIGIN - Directs all requests to the origin server. Possible values = CACHE, POLICY, ORIGIN
precedence	<String>	Read-only	This argument is used only when configuring content switching on the specified virtual server. This is applicable only if both the URL and RULE-based policies have been configured on the same virtual server. It specifies the type of policy (URL or RULE) that takes precedence on the content switching virtual server. The default setting is RULE. l URL - In this case, the incoming request is matched against the URL-based policies

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
			before the rule-based policies. l RULE - In this case, the incoming request is matched against the rule-based policies before the URL-based policies. For all URL-based policies, the precedence hierarchy is: 1. Domain and exact URL 2. Domain, prefix, and suffix 3. Domain and suffix 4. Domain and prefix 5. Domain only 6. Exact URL 7. Prefix and suffix 8. Suffix only 9. Prefix only 10. Default.

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
			Possible values = RULE, URL
redirecturl	<String>	Read-only	The URL where traffic is redirected if the virtual server in system becomes unavailable. WARNING! Make sure that the domain you specify in the URL does not match the domain specified in the -d domainName argument of the ###add cs policy### command. If the same domain is specified in both arguments, the request will be continuously redirected to the same unavailable virtual server in the system. If so, the user may not get the requested content.
curaaausers	<Double>	Read-only	The number of current users logged on to this virtual server.
curtotalusers	<Double>	Read-only	The total number of current users connected through this virtual server.

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
domain	<String>	Read-only	The domain name of the server for which a service needs to be added. If the IP address has been specified, the domain name does not need to be specified.
rule	<String>	Read-only	The name of the rule, or expression, if any, that policy for the VPN server is to use. Rules are combinations of expressions. Expressions are simple conditions, such as a test for equality, applied to operands, such as a URL string or an IP address. Expression syntax is described in the Installation and Configuration Guide. The default rule is true. Minimum length = 1
servicename	<String>	Read-only	The name of the service, if any, to which the virtual server policy is bound.
weight	<Double>	Read-only	Weight for this service, if any. This weight is used when the system performs load balancing,

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
			giving greater priority to a specific service. It is useful when the services bound to a virtual server are of different capacity.
cachevserver	<String>	Read-only	The name of the default target cache virtual server, if any, to which requests are redirected.
backupvserver	<String>	Read-only	The name of the backup VPN virtual server for this VPN virtual server.
clttimeout	<Double>	Read-only	The idle time, if any, in seconds after which the client connection is terminated.
somethod	<String>	Read-only	VPN client applications are allocated from a block of intranet IP addresses. That block may be exhausted after a certain number of connections. This switch specifies the method used to determine whether or not a new connection will spill over, or exhaust,

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
			the allocated block of intranet IP addresses for that application. Possible values are CONNECTION or DYNAMICCONNECTION. CONNECTION means that a static integer value is the hard limit for the spillover threshold. The spillover threshold is described below. DYNAMICCONNECTION means that the spillover threshold is set according to the maximum number of connections defined for the VPN virtual server. Possible values = CONNECTION, DYNAMICCONNECTION,

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
			BANDWIDTH, HEALTH, NONE
sothreshold	<Double>	Read-only	VPN client applications are allocated from a block of intranet IP addresses. That block may be exhausted after a certain number of connections. The value of this option is the number of client connections after which the mapped IP address is used as the client source IP address instead of an address from the allocated block of intranet IP addresses.
sopersistence	<String>	Read-only	Whether or not cookie-based site persistance is enabled for this VPN vserver. Possible values are 'ConnectionProxy', HTTPRedirect, or NONE.

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
			Possible values = ENABLED, DISABLED
sopersistencetimeout	<Double>	Read-only	The timeout, if any, for cookie-based site persistance of this VPN vserver.
usemip	<String>	Read-only	Deprecated. See 'map' below. Possible values = ENABLED, DISABLED
map	<String>	Read-only	Whether or not mapped IP addresses are ON or OFF. Mapped IP addresses are source IP addresses for the virtual servers running on the Citrix ADC. Mapped IP addresses are used by the system to connect to the backend servers. Possible values = ON, OFF
bindpoint	<String>	Read-only	Bindpoint to which the policy is bound. Possible values = REQUEST, RESPONSE, ICA_REQUEST,

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
			OTHERTCP_REQUEST
disableprimaryondown	<String>	Read-only	Tells whether traffic will continue reaching backup virtual servers even after the primary virtual server comes UP from DOWN state. Possible values = ENABLED, DISABLED
secondary	<Boolean>	Read-only	Binds the authentication policy as the secondary policy to use in a two-factor configuration. A user must then authenticate not only via a primary authentication method but also via a secondary authentication method. User groups are aggregated across both. The user name must be exactly the same for both authentication methods, but they can require different passwords.
groupextraction	<Boolean>	Read-only	Binds the authentication policy to a tertiary chain

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
			which will be used only for group extraction. The user will not authenticate against this server, and this will only be called if primary and/or secondary authentication has succeeded.
epaprofileoptional	<Boolean>	Read-only	Mark the EPA profile optional for preauthentication EPA profile. User would be shown a logon page even if the EPA profile fails to evaluate.
ngname	<String>	Read-only	Node group devno to which this authentication virtual sever belongs.
csvserver	<String>	Read-only	Name of the CS vserver to which the VPN vserver is bound.
analyticsprofile	<String>	Read-only	Name of the analytics profile bound to the VPN Vserver.
nodefaultbindings	<String>	Read-only	to determine if the configuration will have default ssl CIPHER and ECC curve bindings.

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
			Default value: NO Possible values = YES, NO
response	<String>	Read-only	.
__count	<Double>	Read-only	count parameter

Operations

(click to see Properties)

ADD| DELETE| UPDATE| UNSET| ENABLE| DISABLE| RENAME| CHECK| GET (ALL)| GET| COUNT

Some options that you can use for each operations:

- **Getting warnings in response:**NITRO allows you to get warnings in an operation by specifying the "warning" query parameter as "yes". For example, to get warnings while connecting to the Citrix ADC appliance, the URL is as follows:

  http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/login?warning=yes

  If any, the warnings are displayed in the response payload with the HTTP code "209 X-NITRO-WARNING".

- **Authenticated access for individual NITRO operations:**NITRO allows you to logon to the Citrix ADC appliance to perform individual operations. You can use this option instead of creating a NITRO session (using the login object) and then using that session to perform all operations,

  To do this, you must specify the username and password in the request header of the NITRO request as follows:

  X-NITRO-USER:<username>

  X-NITRO-PASS:<password>

  Note: In such cases, make sure that the request header DOES not include the following:

  Cookie:NITRO_AUTH_TOKEN=<tokenvalue>

`*Note:` *

Mandatory parameters are marked in redand placeholder content is marked in <green>.

add

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/vpnvserver

HTTP Method:POST

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Content-Type:application/json

Request Payload:

```
{"vpnvserver":{

<b>"name":<String_value>,

</b><b>"servicetype":<String_value>,

</b>"ipv46":<String_value>,

"range":<Double_value>,

"port":<Integer_value>,

"ipset":<String_value>,

"state":<String_value>,

"authentication":<String_value>,

"doublehop":<String_value>,

"maxaaausers":<Double_value>,

"icaonly":<String_value>,

"icaproxysessionmigration":<String_value>,

"dtls":<String_value>,
```

```
"loginonce":<String_value>,

"advancedepa":<String_value>,

"devicecert":<String_value>,

"certkeynames":<String_value>,

"downstateflush":<String_value>,

"listenpolicy":<String_value>,

"listenpriority":<Double_value>,

"tcpprofilename":<String_value>,

"httpprofilename":<String_value>,

"comment":<String_value>,

"appflowlog":<String_value>,

"icmpvsrresponse":<String_value>,

"rhistate":<String_value>,

"netprofile":<String_value>,

"cginfrahomepageredirect":<String_value>,

"maxloginattempts":<Double_value>,

"failedlogintimeout":<Double_value>,

"l2conn":<String_value>,

"deploymenttype":<String_value>,

"rdpserverprofilename":<String_value>,

"windowsepapluginupgrade":<String_value>,

"linuxepapluginupgrade":<String_value>,
```

```
"macepapluginupgrade":<String_value>,

"logoutonsmartcardremoval":<String_value>,

"userdomains":<String_value>,

"authnprofile":<String_value>,

"vserverfqdn":<String_value>,

"pcoipvserverprofilename":<String_value>

}}
```

Response:

HTTP Status Code on Success: 201 Created HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error

delete

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/vpnvserver/name_value<String>

HTTP Method:DELETE

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue>

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error

update

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/vpnvserver

HTTP Method:PUT

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Content-Type:application/json

Request Payload:

```
{"vpnvserver":{

<b>"name":<String_value>,

</b>"ipv46":<String_value>,

"ipset":<String_value>,

"authentication":<String_value>,

"doublehop":<String_value>,

"icaonly":<String_value>,

"icaproxysessionmigration":<String_value>,

"dtls":<String_value>,

"loginonce":<String_value>,

"advancedepa":<String_value>,

"devicecert":<String_value>,

"certkeynames":<String_value>,

"maxaaausers":<Double_value>,

"downstateflush":<String_value>,

"listenpolicy":<String_value>,

"listenpriority":<Double_value>,

"tcpprofilename":<String_value>,

"httpprofilename":<String_value>,
```

```
"comment":<String_value>,

"appflowlog":<String_value>,

"icmpvsrresponse":<String_value>,

"rhistate":<String_value>,

"netprofile":<String_value>,

"cginfrahomepageredirect":<String_value>,

"maxloginattempts":<Double_value>,

"rdpserverprofilename":<String_value>,

"failedlogintimeout":<Double_value>,

"l2conn":<String_value>,

"windowsepapluginupgrade":<String_value>,

"macepapluginupgrade":<String_value>,

"linuxepapluginupgrade":<String_value>,

"logoutonsmartcardremoval":<String_value>,

"userdomains":<String_value>,

"authnprofile":<String_value>,

"vserverfqdn":<String_value>,

"pcoipvserverprofilename":<String_value>

}}
```

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error

unset

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/vpnvserver?**action=unset**

HTTP Method:POST

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Content-Type:application/json

Request Payload:

```
{"vpnvserver":{

<b>"name":<String_value>,

</b>"ipset":true,

"authentication":true,

"doublehop":true,

"icaonly":true,

"icaproxysessionmigration":true,

"dtls":true,

"loginonce":true,

"advancedepa":true,

"devicecert":true,

"certkeynames":true,

"maxaaausers":true,
```

```
"downstateflush":true,

"listenpolicy":true,

"listenpriority":true,

"tcpprofilename":true,

"httpprofilename":true,

"comment":true,

"appflowlog":true,

"icmpvsrresponse":true,

"rhistate":true,

"netprofile":true,

"cginfrahomepageredirect":true,

"maxloginattempts":true,

"rdpserverprofilename":true,

"l2conn":true,

"windowsepapluginupgrade":true,

"macepapluginupgrade":true,

"linuxepapluginupgrade":true,

"logoutonsmartcardremoval":true,

"userdomains":true,

"authnprofile":true,

"vserverfqdn":true,

"pcoipvserverprofilename":true
```

```
}}
```

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error

enable

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/vpnvserver?**action=enable**

HTTP Method:POST

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Content-Type:application/json

Request Payload:

```
{"vpnvserver":{

<b>"name":<String_value>

</b>}}
```

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error

disable

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/vpnvserver?**action=disable**

HTTP Method:POST

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Content-Type:application/json

Request Payload:

```
{"vpnvserver":{

<b>"name":<String_value>

</b>}}
```

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error

rename

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/vpnvserver?**action=rename**

HTTP Method:POST

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Content-Type:application/json

Request Payload:

```
{"vpnvserver":{

<b>"name":<String_value>,

</b><b>"newname":<String_value>

</b>}}
```

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error

check

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/vpnvserver?**action=check**

HTTP Method:POST

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Content-Type:application/json

Request Payload:

```
{"vpnvserver":{

<b>"name":<String_value>

</b>}}
```

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Payload:**

```
{ "vpnvserver": [ {

<b>"name":<String_value>,

</b>"response":<String_value>



}]}
```

get (all)

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/vpnvserver

Query-parameters:

attrs

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/vpnvserver?**attrs=property-name1,property-name2**

Use this query parameter to specify the resource details that you want to retrieve.

filter

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/vpnvserver?**filter=property-name1:property-val1,property-name2:property-val2**

Use this query-parameter to get the filtered set of vpnvserver resources configured on Citrix ADC. Filtering can be done on any of the properties of the resource.

view

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/vpnvserver?**view=summary**

Note: By default, the retrieved results are displayed in detail view (?view=detail).

pagination

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/vpnvserver?**pagesize=#no;pageno=#no**

Use this query-parameter to get the vpnvserver resources in chunks.

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "vpnvserver": [ {

"name":<String_value>,

"ip":<String_value>,

"ipv46":<String_value>,

"value":<String_value>,

"port":<Integer_value>,

"range":<Double_value>,

"ipset":<String_value>,

"servicetype":<String_value>,

"type":<String_value>,

"curstate":<String_value>,

"status":<Integer_value>,

"cachetype":<String_value>,

"redirect":<String_value>,

"precedence":<String_value>,

"redirecturl":<String_value>,

"authentication":<String_value>,

"doublehop":<String_value>,

"icaonly":<String_value>,
```

```
"icaproxysessionmigration":<String_value>,

"dtls":<String_value>,

"loginonce":<String_value>,

"advancedepa":<String_value>,

"devicecert":<String_value>,

"certkeynames":<String_value>,

"maxaaausers":<Double_value>,

"curaaausers":<Double_value>,

"curtotalusers":<Double_value>,

"domain":<String_value>,

"rule":<String_value>,

"policyname":<String_value>,

"servicename":<String_value>,

"weight":<Double_value>,

"cachevserver":<String_value>,

"backupvserver":<String_value>,

"clttimeout":<Double_value>,

"somethod":<String_value>,

"sothreshold":<Double_value>,

"sopersistence":<String_value>,

"sopersistencetimeout":<Double_value>,

"usemip":<String_value>,
```

```
"map":<String_value>,

"downstateflush":<String_value>,

"bindpoint":<String_value>,

"disableprimaryondown":<String_value>,

"listenpolicy":<String_value>,

"listenpriority":<Double_value>,

"tcpprofilename":<String_value>,

"httpprofilename":<String_value>,

"comment":<String_value>,

"appflowlog":<String_value>,

"icmpvsrresponse":<String_value>,

"rhistate":<String_value>,

"netprofile":<String_value>,

"cginfrahomepageredirect":<String_value>,

"maxloginattempts":<Double_value>,

"failedlogintimeout":<Double_value>,

"secondary":<Boolean_value>,

"groupextraction":<Boolean_value>,

"deploymenttype":<String_value>,

"windowsepapluginupgrade":<String_value>,

"linuxepapluginupgrade":<String_value>,

"macepapluginupgrade":<String_value>,
```

```
"logoutonsmartcardremoval":<String_value>,

"epaprofileoptional":<Boolean_value>,

"rdpserverprofilename":<String_value>,

"ngname":<String_value>,

"state":<String_value>,

"l2conn":<String_value>,

"userdomains":<String_value>,

"csvserver":<String_value>,

"authnprofile":<String_value>,

"vserverfqdn":<String_value>,

"pcoipvserverprofilename":<String_value>,

"analyticsprofile":<String_value>,

"nodefaultbindings":<String_value>

}]}
```

get

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/vpnvserver/name_value<String>

Query-parameters:

attrs

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/vpnvserver/name_value<String>?**attrs=property-name1,property-name2**

Use this query parameter to specify the resource details that you want to retrieve.

view

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/vpnvserver/name_value<String>?**view=summary**

Note: By default, the retrieved results are displayed in detail view (?view=detail).

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "vpnvserver": [ {

"name":<String_value>,

"ip":<String_value>,

"ipv46":<String_value>,

"value":<String_value>,

"port":<Integer_value>,

"range":<Double_value>,

"ipset":<String_value>,

"servicetype":<String_value>,

"type":<String_value>,

"curstate":<String_value>,
```

```
"status":<Integer_value>,

"cachetype":<String_value>,

"redirect":<String_value>,

"precedence":<String_value>,

"redirecturl":<String_value>,

"authentication":<String_value>,

"doublehop":<String_value>,

"icaonly":<String_value>,

"icaproxysessionmigration":<String_value>,

"dtls":<String_value>,

"loginonce":<String_value>,

"advancedepa":<String_value>,

"devicecert":<String_value>,

"certkeynames":<String_value>,

"maxaaausers":<Double_value>,

"curaaausers":<Double_value>,

"curtotalusers":<Double_value>,

"domain":<String_value>,

"rule":<String_value>,

"policyname":<String_value>,

"servicename":<String_value>,

"weight":<Double_value>,
```

```
"cachevserver":<String_value>,

"backupvserver":<String_value>,

"clttimeout":<Double_value>,

"somethod":<String_value>,

"sothreshold":<Double_value>,

"sopersistence":<String_value>,

"sopersistencetimeout":<Double_value>,

"usemip":<String_value>,

"map":<String_value>,

"downstateflush":<String_value>,

"bindpoint":<String_value>,

"disableprimaryondown":<String_value>,

"listenpolicy":<String_value>,

"listenpriority":<Double_value>,

"tcpprofilename":<String_value>,

"httpprofilename":<String_value>,

"comment":<String_value>,

"appflowlog":<String_value>,

"icmpvsrresponse":<String_value>,

"rhistate":<String_value>,

"netprofile":<String_value>,

"cginfrahomepageredirect":<String_value>,
```

```
"maxloginattempts":<Double_value>,

"failedlogintimeout":<Double_value>,

"secondary":<Boolean_value>,

"groupextraction":<Boolean_value>,

"deploymenttype":<String_value>,

"windowsepapluginupgrade":<String_value>,

"linuxepapluginupgrade":<String_value>,

"macepapluginupgrade":<String_value>,

"logoutonsmartcardremoval":<String_value>,

"epaprofileoptional":<Boolean_value>,

"rdpserverprofilename":<String_value>,

"ngname":<String_value>,

"state":<String_value>,

"l2conn":<String_value>,

"userdomains":<String_value>,

"csvserver":<String_value>,

"authnprofile":<String_value>,

"vserverfqdn":<String_value>,

"pcoipvserverprofilename":<String_value>,

"analyticsprofile":<String_value>,

"nodefaultbindings":<String_value>

}]}
```

count

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/vpnvserver?**count=yes**

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "vpnvserver": [ { "__count": "#no"} ] }
```

vpnvserver_aaapreauthenticationpolicy_binding

Binding object showing the aaapreauthenticationpolicy that can be bound to vpnvserver.

Properties

(click to see Operations)

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
priority	<Double>	Read-write	Integer specifying the policy's priority. The lower the number, the higher the priority. Policies are evaluated in the order of their priority

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
			numbers. Maximum value for default syntax policies is 2147483647 and for classic policies is 64000. Minimum value = 0 Maximum value = 2147483647
gotopriorityexpression	<String>	Read-write	Applicable only to advance vpn session policy. Expression or other value specifying the next policy to evaluate if the current policy evaluates to TRUE. Specify one of the following values: * NEXT - Evaluate the policy with the next higher priority number. * END - End policy evaluation. * An expression that evaluates to a number. If you specify an expression, the number to which it evaluates determines the next policy to evaluate, as follows: * If the expression evaluates to a higher numbered priority, the policy with that priority is evaluated

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
			next. * If the expression evaluates to the priority of the current policy, the policy with the next higher numbered priority is evaluated next. * If the expression evaluates to a number that is larger than the largest numbered priority, policy evaluation ends. An UNDEF event is triggered if: * The expression is invalid. * The expression evaluates to a priority number that is numerically lower than the current policy's priority. * The expression evaluates to a priority number that is between the current policy's priority number (say, 30) and the highest priority number (say, 100), but does not match any configured priority number (for example, the expression evaluates to the number 85). This example assumes that the priority number increments by 10 for

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
			every successive policy, and therefore a priority number of 85 does not exist in the policy label.
policy	<String>	Read-write	The name of the policy, if any, bound to the VPN virtual server.
groupextraction	<Boolean>	Read-write	Binds the authentication policy to a tertiary chain which will be used only for group extraction. The user will not authenticate against this server, and this will only be called if primary and/or secondary authentication has succeeded.
name	<String>	Read-write	Name of the virtual server. Minimum length = 1
secondary	<Boolean>	Read-write	Binds the authentication policy as the secondary policy to use in a two-factor configuration. A user must then authenticate not only via a primary authentication method but also via a secondary

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
			authentication method. User groups are aggregated across both. The user name must be exactly the same for both authentication methods, but they can require different passwords.
bindpoint	<String>	Read-write	Bind point to which to bind the policy. Applies only to rewrite and cache policies. If you do not set this parameter, the policy is bound to REQ_DEFAULT or RES_DEFAULT, depending on whether the policy rule is a response-time or a request-time expression. Possible values = REQUEST, RESPONSE, ICA_REQUEST, OTHERTCP_REQUEST
acttype	<Double>	Read-only	.
__count	<Double>	Read-only	count parameter

Operations

(click to see Properties)

ADD:| DELETE:| GET| GET (ALL)| COUNT

Some options that you can use for each operations:

- **Getting warnings in response:**NITRO allows you to get warnings in an operation by specifying the "warning" query parameter as "yes". For example, to get warnings while connecting to the Citrix ADC appliance, the URL is as follows:

  http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/login?warning=yes

  If any, the warnings are displayed in the response payload with the HTTP code "209 X-NITRO-WARNING".

- **Authenticated access for individual NITRO operations:**NITRO allows you to logon to the Citrix ADC appliance to perform individual operations. You can use this option instead of creating a NITRO session (using the login object) and then using that session to perform all operations,

  To do this, you must specify the username and password in the request header of the NITRO request as follows:

  X-NITRO-USER:<username>

  X-NITRO-PASS:<password>

  Note: In such cases, make sure that the request header DOES not include the following:

  Cookie:NITRO_AUTH_TOKEN=<tokenvalue>

`*Note:`*

Mandatory parameters are marked in redand placeholder content is marked in <green>.

add

URL:http://<Citrix-ADC-IP-address(NSIP)/nitro/v1/config/vpnvserver_aaapreauthenticationpolicy_binding

HTTP Method:PUT

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Content-Type:application/json

Request Payload:

```
{

"vpnvserver_aaapreauthenticationpolicy_binding":{

<b>"name":<String_value>,

</b>"policy":<String_value>,

"priority":<Double_value>,

"secondary":<Boolean_value>,

"groupextraction":<Boolean_value>,

"gotopriorityexpression":<String_value>,

"bindpoint":<String_value>

}}
```

Response:

HTTP Status Code on Success: 201 Created HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error

delete

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/vpnvserver_aaapreauthenticationpolicy_binding/name_value<String>

Query-parameters:

args

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/vpnvserver_aaapreauthenticationpolicy_binding/name_value<String>?**args=policy:<String_value>,secondary:<Boolean_value>,groupextraction:<Boolean_value>,bindpoint:<**

HTTP Method:DELETE

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue>

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error

get

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/vpnvserver_aaapreauthenticationpolicy_binding/name_value<String>

Query-parameters:

filter

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/vpnvserver_aaapreauthenticationpolicy_binding/name_value<String>?**filter=property-name1:property-value1,property-name2:property-value2**

Use this query-parameter to get the filtered set of vpnvserver_aaapreauthenticationpolicy_binding resources configured on Citrix ADC. Filtering can be done on any of the properties of the resource.

pagination

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/vpnvserver_aaapreauthenticationpolicy_binding/name_value<String>?**pagesize=#no;pageno=#no**

Use this query-parameter to get the vpnvserver_aaapreauthenticationpolicy_binding resources in chunks.

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "vpnvserver_aaapreauthenticationpolicy_binding": [ {

"priority":<Double_value>,

"gotopriorityexpression":<String_value>,

"policy":<String_value>,

"groupextraction":<Boolean_value>,

"name":<String_value>,

"secondary":<Boolean_value>,

"bindpoint":<String_value>,

"acttype":<Double_value>

}]}
```

get (all)

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/vpnvserver_aaapreauthenticationpolicy_binding

Query-parameters:

bulkbindings

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/vpnvserver_aaapreauthenticationpolicy_binding?**bulkbindings=yes**

NITRO allows you to fetch bindings in bulk.

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "vpnvserver_aaapreauthenticationpolicy_binding": [ {

"priority":<Double_value>,

"gotopriorityexpression":<String_value>,

"policy":<String_value>,

"groupextraction":<Boolean_value>,

"name":<String_value>,

"secondary":<Boolean_value>,

"bindpoint":<String_value>,

"acttype":<Double_value>

}]}
```

count

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/vpnvserver_aaapreauthenticationpolicy_binding/name_value<String>?**count=yes**

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{"vpnvserver_aaapreauthenticationpolicy_binding": [ { "__count":
"#no"} ] }
```

vpnvserver_appcontroller_binding

Binding object showing the appcontroller that can be bound to vpnvserver.

Properties

(click to see Operations)

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
appcontroller	<String>	Read-write	Configured App Controller server in XenMobile deployment.
name	<String>	Read-write	Name of the virtual server. Minimum length = 1
acttype	<Double>	Read-only	.

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
__count	<Double>	Read-only	count parameter

Operations

(click to see Properties)

ADD:| DELETE:| GET| GET (ALL)| COUNT

Some options that you can use for each operations:

- **Getting warnings in response:**NITRO allows you to get warnings in an operation by specifying the "warning" query parameter as "yes". For example, to get warnings while connecting to the Citrix ADC appliance, the URL is as follows:

  http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/login?warning=yes

  If any, the warnings are displayed in the response payload with the HTTP code "209 X-NITRO-WARNING".
- **Authenticated access for individual NITRO operations:**NITRO allows you to logon to the Citrix ADC appliance to perform individual operations. You can use this option instead of creating a NITRO session (using the login object) and then using that session to perform all operations,

  To do this, you must specify the username and password in the request header of the NITRO request as follows:

  X-NITRO-USER:<username>

  X-NITRO-PASS:<password>

  Note: In such cases, make sure that the request header DOES not include the following:

  Cookie:NITRO_AUTH_TOKEN=<tokenvalue>

`*Note:`*

Mandatory parameters are marked in redand placeholder content is marked in <green>.

add

URL:http://<Citrix-ADC-IP-address(NSIP)/nitro/v1/config/vpnvserver_appcontroller_binding

HTTP Method:PUT

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Content-Type:application/json

Request Payload:

```
{

"vpnvserver_appcontroller_binding":{

<b>"name":<String_value>,

</b>"appcontroller":<String_value>

}}
```

Response:

HTTP Status Code on Success: 201 Created HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error

delete

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/vpnvserver_appcontroller_binding/name_value<String>

Query-parameters:

args

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/vpnvserver_appcontroller_binding/name_value<String>?**args=appcontroller:<String_value>**

HTTP Method:DELETE

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue>

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error

get

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/vpnvserver_appcontroller_binding/name_value<String>

Query-parameters:

filter

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/vpnvserver_appcontroller_binding/name_value<String>?**filter=property-name1:property-value1,property-name2:property-value2**

Use this query-parameter to get the filtered set of vpnvserver_appcontroller_binding resources configured on Citrix ADC. Filtering can be done on any of the properties of the resource.

pagination

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/vpnvserver_appcontroller_binding/name_value<String>?**pagesize=#no;pageno=#no**

Use this query-parameter to get the vpnvserver_appcontroller_binding resources in chunks.

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "vpnvserver_appcontroller_binding": [ {
```

```
"appcontroller":<String_value>,

"name":<String_value>,

"acttype":<Double_value>

}]}
```

get (all)

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/vpnvserver_appcontroller_binding

Query-parameters:

bulkbindings

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/vpnvserver_appcontroller_binding?**bulkbindings=yes**

NITRO allows you to fetch bindings in bulk.

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "vpnvserver_appcontroller_binding": [ {

"appcontroller":<String_value>,
```

```
"name":<String_value>,

"acttype":<Double_value>

}]}
```

count

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/vpnvserver_appcontroller_binding/name_value<String>?**count=yes**

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{"vpnvserver_appcontroller_binding": [ { "__count": "#no"} ] }
```

vpnvserver_appflowpolicy_binding

Binding object showing the appflowpolicy that can be bound to vpnvserver.

Properties

(click to see Operations)

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
priority	<Double>	Read-write	Integer specifying the policy's priority. The lower the number, the higher the priority. Policies are evaluated in the order of their priority numbers. Maximum value for default syntax policies is 2147483647 and for classic policies is 64000. Minimum value = 0 Maximum value = 2147483647
gotopriorityexpression	<String>	Read-write	Next priority expression.
policy	<String>	Read-write	The name of the policy, if any, bound to the VPN virtual server.
groupextraction	<Boolean>	Read-write	Binds the authentication policy to a tertiary chain which will be used only for group extraction. The user will not authenticate against this server, and this will only be called if primary and/or secondary authentication has succeeded.

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
name	<String>	Read-write	Name of the virtual server. Minimum length = 1
secondary	<Boolean>	Read-write	Binds the authentication policy as the secondary policy to use in a two-factor configuration. A user must then authenticate not only via a primary authentication method but also via a secondary authentication method. User groups are aggregated across both. The user name must be exactly the same for both authentication methods, but they can require different passwords.
bindpoint	<String>	Read-write	Bindpoint to which the policy is bound. Possible values = REQUEST, RESPONSE, ICA_REQUEST, OTHERTCP_REQUEST
__count	<Double>	Read-write	count parameter

Operations

(click to see Properties)

ADD:| DELETE:| GET| GET (ALL)| COUNT

Some options that you can use for each operations:

- **Getting warnings in response:**NITRO allows you to get warnings in an operation by specifying the "warning" query parameter as "yes". For example, to get warnings while connecting to the Citrix ADC appliance, the URL is as follows:

  http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/login?warning=yes

  If any, the warnings are displayed in the response payload with the HTTP code "209 X-NITRO-WARNING".

- **Authenticated access for individual NITRO operations:**NITRO allows you to logon to the Citrix ADC appliance to perform individual operations. You can use this option instead of creating a NITRO session (using the login object) and then using that session to perform all operations,

  To do this, you must specify the username and password in the request header of the NITRO request as follows:

  X-NITRO-USER:<username>

  X-NITRO-PASS:<password>

  Note: In such cases, make sure that the request header DOES not include the following:

  Cookie:NITRO_AUTH_TOKEN=<tokenvalue>

`*Note:`*

Mandatory parameters are marked in redand placeholder content is marked in <green>.

add

URL:http://<Citrix-ADC-IP-address(NSIP)/nitro/v1/config/vpnvserver_appflowpolicy_binding

HTTP Method:PUT

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Content-Type:application/json

Request Payload:

```
{

"vpnvserver_appflowpolicy_binding":{

<b>"name":<String_value>,

</b>"policy":<String_value>,

"priority":<Double_value>,

"secondary":<Boolean_value>,

"groupextraction":<Boolean_value>,

"gotopriorityexpression":<String_value>,

"bindpoint":<String_value>

}}
```

Response:

HTTP Status Code on Success: 201 Created HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error

delete

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/vpnvserver_appflowpolicy_binding/name_value<String>

Query-parameters:

args

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/vpnvserver_appflowpolicy_binding/name_value<String>?
args=policy:<String_value>,secondary:<Boolean_value>,groupextraction:<Boolean_value>,bindpoint:<

HTTP Method:DELETE

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue>

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error

get

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/vpnvserver_appflowpolicy_binding/name_value<String>

Query-parameters:

filter

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/vpnvserver_appflowpolicy_binding/name_value<String>?**filter=property-name1:property-value1,property-name2:property-value2**

Use this query-parameter to get the filtered set of vpnvserver_appflowpolicy_binding resources configured on Citrix ADC. Filtering can be done on any of the properties of the resource.

pagination

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/vpnvserver_appflowpolicy_binding/name_value<String>?**pagesize=#no;pageno=#no**

Use this query-parameter to get the vpnvserver_appflowpolicy_binding resources in chunks.

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "vpnvserver_appflowpolicy_binding": [ {

"priority":<Double_value>,

"gotopriorityexpression":<String_value>,

"policy":<String_value>,

"groupextraction":<Boolean_value>,

"name":<String_value>,

"secondary":<Boolean_value>,

"bindpoint":<String_value>

}]}
```

get (all)

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/vpnvserver_appflowpolicy_binding

Query-parameters:

bulkbindings

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/vpnvserver_appflowpolicy_binding?**bulkbindings=yes**

NITRO allows you to fetch bindings in bulk.

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "vpnvserver_appflowpolicy_binding": [ {

"priority":<Double_value>,

"gotopriorityexpression":<String_value>,

"policy":<String_value>,

"groupextraction":<Boolean_value>,

"name":<String_value>,

"secondary":<Boolean_value>,

"bindpoint":<String_value>

}]}
```

count

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/vpnvserver_appflowpolicy_binding/name_value<String>?**count=yes**

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{"vpnvserver_appflowpolicy_binding": [ { "__count": "#no"} ] }
```

vpnvserver_auditnslogpolicy_binding

Binding object showing the auditnslogpolicy that can be bound to vpnvserver.

Properties

(click to see Operations)

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
priority	<Double>	Read-write	Integer specifying the policy's priority. The lower the number, the higher the priority. Policies are evaluated in the order of their priority numbers. Maximum value for default syntax policies is 2147483647 and for classic policies is 64000. Minimum value = 0 Maximum value = 2147483647

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
gotopriorityexpression	<String>	Read-write	Applicable only to advance vpn session policy. Expression or other value specifying the next policy to evaluate if the current policy evaluates to TRUE. Specify one of the following values: * NEXT - Evaluate the policy with the next higher priority number. * END - End policy evaluation. * An expression that evaluates to a number. If you specify an expression, the number to which it evaluates determines the next policy to evaluate, as follows: * If the expression evaluates to a higher numbered priority, the policy with that priority is evaluated next. * If the expression evaluates to the priority of the current policy, the policy with the next higher numbered priority is evaluated next. * If the expression evaluates to a number that is larger than the

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
			largest numbered priority, policy evaluation ends. An UNDEF event is triggered if: * The expression is invalid. * The expression evaluates to a priority number that is numerically lower than the current policy's priority. * The expression evaluates to a priority number that is between the current policy's priority number (say, 30) and the highest priority number (say, 100), but does not match any configured priority number (for example, the expression evaluates to the number 85). This example assumes that the priority number increments by 10 for every successive policy, and therefore a priority number of 85 does not exist in the policy label.
policy	<String>	Read-write	The name of the policy, if any, bound to the VPN virtual server.

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
groupextraction	<Boolean>	Read-write	Binds the authentication policy to a tertiary chain which will be used only for group extraction. The user will not authenticate against this server, and this will only be called if primary and/or secondary authentication has succeeded.
name	<String>	Read-write	Name of the virtual server. Minimum length = 1
secondary	<Boolean>	Read-write	Binds the authentication policy as the secondary policy to use in a two-factor configuration. A user must then authenticate not only via a primary authentication method but also via a secondary authentication method. User groups are aggregated across both. The user name must be exactly the same for both authentication methods, but they can require different passwords.

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
bindpoint	<String>	Read-write	Bind point to which to bind the policy. Applies only to rewrite and cache policies. If you do not set this parameter, the policy is bound to REQ_DEFAULT or RES_DEFAULT, depending on whether the policy rule is a response-time or a request-time expression. Possible values = REQUEST, RESPONSE, ICA_REQUEST, OTHERTCP_REQUEST
acttype	<Double>	Read-only	.
__count	<Double>	Read-only	count parameter

Operations

(click to see Properties)

ADD:| DELETE:| GET| GET (ALL)| COUNT

Some options that you can use for each operations:

- **Getting warnings in response:**NITRO allows you to get warnings in an operation by specifying the "warning" query parameter as "yes". For example, to get warnings while connecting to the Citrix ADC appliance, the URL is as follows:

  http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/login?warning=yes

  If any, the warnings are displayed in the response payload with the HTTP code "209 X-NITRO-WARNING".

- **Authenticated access for individual NITRO operations:**NITRO allows you to logon to the Citrix ADC appliance to perform individual operations. You can use this option instead of creating a NITRO session (using the login object) and then using that session to perform all operations,

  To do this, you must specify the username and password in the request header of the NITRO request as follows:

  X-NITRO-USER:<username>

  X-NITRO-PASS:<password>

  Note: In such cases, make sure that the request header DOES not include the following:

  Cookie:NITRO_AUTH_TOKEN=<tokenvalue>

`*Note:`*

Mandatory parameters are marked in redand placeholder content is marked in <green>.

add

URL:http://<Citrix-ADC-IP-address(NSIP)/nitro/v1/config/vpnvserver_auditnslogpolicy_binding

HTTP Method:PUT

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Content-Type:application/json

Request Payload:

```
{

"vpnvserver_auditnslogpolicy_binding":{

<b>"name":<String_value>,

</b>"policy":<String_value>,

"priority":<Double_value>,

"secondary":<Boolean_value>,
```

```
"groupextraction":<Boolean_value>,

"gotopriorityexpression":<String_value>,

"bindpoint":<String_value>

}}
```

Response:

HTTP Status Code on Success: 201 Created HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error

delete

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/vpnvserver_auditnslogpolicy_binding/name_value<String>

Query-parameters:

args

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/vpnvserver_auditnslogpolicy_binding/name_value<String>?
args=policy:<String_value>,secondary:<Boolean_value>,groupextraction:<Boolean_value>,bindpoint:<

HTTP Method:DELETE

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue>

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error

get

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/vpnvserver_auditnslogpolicy_binding/name_value<String>

Query-parameters:

filter

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/vpnvserver_auditnslogpolicy_binding/name_value<String>?**filter=property-name1:property-value1,property-name2:property-value2**

Use this query-parameter to get the filtered set of vpnvserver_auditnslogpolicy_binding resources configured on Citrix ADC. Filtering can be done on any of the properties of the resource.

pagination

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/vpnvserver_auditnslogpolicy_binding/name_value<String>?**pagesize=#no;pageno=#no**

Use this query-parameter to get the vpnvserver_auditnslogpolicy_binding resources in chunks.

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "vpnvserver_auditnslogpolicy_binding": [ {

"priority":<Double_value>,

"gotopriorityexpression":<String_value>,

"policy":<String_value>,

"groupextraction":<Boolean_value>,
```

```
"name":<String_value>,

"secondary":<Boolean_value>,

"bindpoint":<String_value>,

"acttype":<Double_value>

}]}
```

get (all)

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/vpnvserver_auditnslogpolicy_binding

Query-parameters:

bulkbindings

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/vpnvserver_auditnslogpolicy_binding?**bulkbindings=yes**

NITRO allows you to fetch bindings in bulk.

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "vpnvserver_auditnslogpolicy_binding": [ {
```

```
"priority":<Double_value>,

"gotopriorityexpression":<String_value>,

"policy":<String_value>,

"groupextraction":<Boolean_value>,

"name":<String_value>,

"secondary":<Boolean_value>,

"bindpoint":<String_value>,

"acttype":<Double_value>

}]}
```

count

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/vpnvserver_auditnslogpolicy_binding/name_value<String>?**count=yes**

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{"vpnvserver_auditnslogpolicy_binding": [ { "__count": "#no"} ] }
```

vpnvserver_auditsyslogpolicy_binding

Binding object showing the auditsyslogpolicy that can be bound to vpnvserver.

Properties

(click to see Operations)

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
priority	<Double>	Read-write	Integer specifying the policy's priority. The lower the number, the higher the priority. Policies are evaluated in the order of their priority numbers. Maximum value for default syntax policies is 2147483647 and for classic policies is 64000. Minimum value = 0 Maximum value = 2147483647
gotopriorityexpression	<String>	Read-write	Applicable only to advance vpn session policy. Expression or other value specifying the next policy to evaluate if the current policy evaluates to TRUE. Specify one of the following values: * NEXT - Evaluate the policy with the next higher priority

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
			number. * END - End policy evaluation. * An expression that evaluates to a number. If you specify an expression, the number to which it evaluates determines the next policy to evaluate, as follows: * If the expression evaluates to a higher numbered priority, the policy with that priority is evaluated next. * If the expression evaluates to the priority of the current policy, the policy with the next higher numbered priority is evaluated next. * If the expression evaluates to a number that is larger than the largest numbered priority, policy evaluation ends. An UNDEF event is triggered if: * The expression is invalid. * The expression evaluates to a priority number that is numerically lower than the current policy's priority. * The expression evaluates

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
			to a priority number that is between the current policy's priority number (say, 30) and the highest priority number (say, 100), but does not match any configured priority number (for example, the expression evaluates to the number 85). This example assumes that the priority number increments by 10 for every successive policy, and therefore a priority number of 85 does not exist in the policy label.
policy	<String>	Read-write	The name of the policy, if any, bound to the VPN virtual server.
groupextraction	<Boolean>	Read-write	Binds the authentication policy to a tertiary chain which will be used only for group extraction. The user will not authenticate against this server, and this will only be called if primary and/or secondary authentication has succeeded.

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
name	<String>	Read-write	Name of the virtual server. Minimum length = 1
secondary	<Boolean>	Read-write	Binds the authentication policy as the secondary policy to use in a two-factor configuration. A user must then authenticate not only via a primary authentication method but also via a secondary authentication method. User groups are aggregated across both. The user name must be exactly the same for both authentication methods, but they can require different passwords.
bindpoint	<String>	Read-write	Bind point to which to bind the policy. Applies only to rewrite and cache policies. If you do not set this parameter, the policy is bound to REQ_DEFAULT or RES_DEFAULT, depending on whether the policy rule is a response-

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
			time or a request-time expression. Possible values = REQUEST, RESPONSE, ICA_REQUEST, OTHERTCP_REQUEST
acttype	<Double>	Read-only	.
__count	<Double>	Read-only	count parameter

Operations

(click to see Properties)

ADD:| DELETE:| GET| GET (ALL)| COUNT

Some options that you can use for each operations:

- **Getting warnings in response:**NITRO allows you to get warnings in an operation by specifying the "warning" query parameter as "yes". For example, to get warnings while connecting to the Citrix ADC appliance, the URL is as follows:

  http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/login?warning=yes

  If any, the warnings are displayed in the response payload with the HTTP code "209 X-NITRO-WARNING".
- **Authenticated access for individual NITRO operations:**NITRO allows you to logon to the Citrix ADC appliance to perform individual operations. You can use this option instead of creating a NITRO session (using the login object) and then using that session to perform all operations,

  To do this, you must specify the username and password in the request header of the NITRO request as follows:

  X-NITRO-USER:<username>

  X-NITRO-PASS:<password>

Note: In such cases, make sure that the request header DOES not include the following:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue>

`*Note:` *

Mandatory parameters are marked in redand placeholder content is marked in <green>.

add

URL:http://<Citrix-ADC-IP-address(NSIP)/nitro/v1/config/vpnvserver_auditsyslogpolicy_binding

HTTP Method:PUT

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Content-Type:application/json

Request Payload:

```
{

"vpnvserver_auditsyslogpolicy_binding":{

<b>"name":<String_value>,

</b>"policy":<String_value>,

"priority":<Double_value>,

"secondary":<Boolean_value>,

"groupextraction":<Boolean_value>,

"gotopriorityexpression":<String_value>,

"bindpoint":<String_value>

}}
```

Response:

HTTP Status Code on Success: 201 Created HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error

delete

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/vpnvserver_auditsyslogpolicy_binding/name_value<String>

Query-parameters:

args

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/vpnvserver_auditsyslogpolicy_binding/name_value<String>?**args=policy:<String_value>,secondary:<Boolean_value>,groupextraction:<Boolean_value>,bindpoint:<**

HTTP Method:DELETE

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue>

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error

get

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/vpnvserver_auditsyslogpolicy_binding/name_value<String>

Query-parameters:

filter

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/vpnvserver_auditsyslogpolicy_binding/name_value<String>?**filter=property-name1:property-value1,property-name2:property-value2**

Use this query-parameter to get the filtered set of vpnvserver_auditsyslogpolicy_binding resources configured on Citrix ADC. Filtering can be done on any of the properties of the resource.

pagination

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/vpnvserver_auditsyslogpolicy_binding/name_value<String>?**pagesize=#no;pageno=#no**

Use this query-parameter to get the vpnvserver_auditsyslogpolicy_binding resources in chunks.

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "vpnvserver_auditsyslogpolicy_binding": [ {

"priority":<Double_value>,

"gotopriorityexpression":<String_value>,

"policy":<String_value>,

"groupextraction":<Boolean_value>,

"name":<String_value>,

"secondary":<Boolean_value>,

"bindpoint":<String_value>,

"acttype":<Double_value>

}]}
```

get (all)

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/vpnvserver_auditsyslogpolicy_binding

Query-parameters:

bulkbindings

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/vpnvserver_auditsyslogpolicy_binding?**bulkbindings=yes**

NITRO allows you to fetch bindings in bulk.

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "vpnvserver_auditsyslogpolicy_binding": [ {

"priority":<Double_value>,

"gotopriorityexpression":<String_value>,

"policy":<String_value>,

"groupextraction":<Boolean_value>,

"name":<String_value>,

"secondary":<Boolean_value>,
```

```
"bindpoint":<String_value>,

"acttype":<Double_value>

}]}
```

count

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/vpnvserver_auditsyslogpolicy_binding/name_value<String>?**count=yes**

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{"vpnvserver_auditsyslogpolicy_binding": [ { "__count": "#no"} ] }
```

vpnvserver_authenticationoauthidppolicy_binding

Binding object showing the authenticationoauthidppolicy that can be bound to vpnvserver.

Properties

(click to see Operations)

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
priority	<Double>	Read-write	Integer specifying the policy's priority. The lower the number, the higher the priority. Policies are evaluated in the order of their priority numbers. Maximum value for default syntax policies is 2147483647 and for classic policies is 64000. Minimum value = 0 Maximum value = 2147483647
gotopriorityexpression	<String>	Read-write	Next priority expression.
policy	<String>	Read-write	The name of the policy, if any, bound to the VPN virtual server.
groupextraction	<Boolean>	Read-write	Binds the authentication policy to a tertiary chain which will be used only for group extraction. The user will not authenticate against this server, and this will only be called if primary and/or secondary authentication has succeeded.

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
name	<String>	Read-write	Name of the virtual server. Minimum length = 1
secondary	<Boolean>	Read-write	Binds the authentication policy as the secondary policy to use in a two-factor configuration. A user must then authenticate not only via a primary authentication method but also via a secondary authentication method. User groups are aggregated across both. The user name must be exactly the same for both authentication methods, but they can require different passwords.
bindpoint	<String>	Read-write	Bind point to which to bind the policy. Applies only to rewrite and cache policies. If you do not set this parameter, the policy is bound to REQ_DEFAULT or RES_DEFAULT, depending on whether the policy rule is a response-

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
			time or a request-time expression. Possible values = REQUEST, RESPONSE, ICA_REQUEST, OTHERTCP_REQUEST
__count	<Double>	Read-write	count parameter

Operations

(click to see Properties)

ADD:| DELETE:| GET| GET (ALL)| COUNT

Some options that you can use for each operations:

- **Getting warnings in response:**NITRO allows you to get warnings in an operation by specifying the "warning" query parameter as "yes". For example, to get warnings while connecting to the Citrix ADC appliance, the URL is as follows:

  http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/login?warning=yes

  If any, the warnings are displayed in the response payload with the HTTP code "209 X-NITRO-WARNING".

- **Authenticated access for individual NITRO operations:**NITRO allows you to logon to the Citrix ADC appliance to perform individual operations. You can use this option instead of creating a NITRO session (using the login object) and then using that session to perform all operations,

  To do this, you must specify the username and password in the request header of the NITRO request as follows:

  X-NITRO-USER:<username>

  X-NITRO-PASS:<password>

  Note: In such cases, make sure that the request header DOES not include the following:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue>

`*Note:` *

Mandatory parameters are marked in redand placeholder content is marked in <green>.

add

URL:http://<Citrix-ADC-IP-address(NSIP)/nitro/v1/config/vpnvserver_authenticationoauthidppolicy_binding

HTTP Method:PUT

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Content-Type:application/json

Request Payload:

```
{

"vpnvserver_authenticationoauthidppolicy_binding":{

<b>"name":<String_value>,

</b>"policy":<String_value>,

"priority":<Double_value>,

"secondary":<Boolean_value>,

"groupextraction":<Boolean_value>,

"gotopriorityexpression":<String_value>,

"bindpoint":<String_value>

}}
```

Response:

HTTP Status Code on Success: 201 Created HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error

delete

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/vpnvserver_authenticationoauthidppolicy_binding/name_value<String>

Query-parameters:

args

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/vpnvserver_authenticationoauthidppolicy_binding/name_value<String>?**args=policy:<String_value>,secondary:<Boolean_value>,groupextraction:<Boolean_value>,bindpoint:<**

HTTP Method:DELETE

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue>

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error

get

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/vpnvserver_authenticationoauthidppolicy_binding/name_value<String>

Query-parameters:

filter

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/vpnvserver_authenticationoauthidppolicy_binding/name_value<String>?**filter=property-name1:property-value1,property-name2:property-value2**

Use this query-parameter to get the filtered set of vpnvserver_authenticationoauthidppolicy_binding resources configured on Citrix ADC. Filtering can be done on any of the properties of the resource.

pagination

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/vpnvserver_authenticationoauthidppolicy_binding/name_value<String>?**pagesize=#no;pageno=#no**

Use this query-parameter to get the vpnvserver_authenticationoauthidppolicy_binding resources in chunks.

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "vpnvserver_authenticationoauthidppolicy_binding": [ {

"priority":<Double_value>,

"gotopriorityexpression":<String_value>,

"policy":<String_value>,

"groupextraction":<Boolean_value>,

"name":<String_value>,

"secondary":<Boolean_value>,

"bindpoint":<String_value>

}]}
```

get (all)

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/vpnvserver_authenticationoauthidppolicy_binding

Query-parameters:

bulkbindings

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/vpnvserver_authenticationoauthidppolicy_binding?**bulkbindings=yes**

NITRO allows you to fetch bindings in bulk.

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "vpnvserver_authenticationoauthidppolicy_binding": [ {

"priority":<Double_value>,

"gotopriorityexpression":<String_value>,

"policy":<String_value>,

"groupextraction":<Boolean_value>,

"name":<String_value>,
```

```
"secondary":<Boolean_value>,

"bindpoint":<String_value>

}]}
```

count

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/vpnvserver_authenticationoauthidppolicy_binding/name_value<String>?**count=yes**

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{"vpnvserver_authenticationoauthidppolicy_binding": [ { "__count":
"#no"} ] }
```

vpnvserver_authenticationcertpolicy_binding

Binding object showing the authenticationcertpolicy that can be bound to vpnvserver.

Properties

(click to see Operations)

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
priority	<Double>	Read-write	Integer specifying the policy's priority. The lower the number, the higher the priority. Policies are evaluated in the order of their priority numbers. Maximum value for default syntax policies is 2147483647 and for classic policies is 64000. Minimum value = 0 Maximum value = 2147483647
gotopriorityexpression	<String>	Read-write	Applicable only to advance vpn session policy. Expression or other value specifying the next policy to evaluate if the current policy evaluates to TRUE. Specify one of the following values: * NEXT - Evaluate the policy with the next higher priority number. * END - End policy evaluation. * An expression that evaluates to a number. If you specify an expression, the number to which it evaluates determines the next

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
			policy to evaluate, as follows: * If the expression evaluates to a higher numbered priority, the policy with that priority is evaluated next. * If the expression evaluates to the priority of the current policy, the policy with the next higher numbered priority is evaluated next. * If the expression evaluates to a number that is larger than the largest numbered priority, policy evaluation ends. An UNDEF event is triggered if: * The expression is invalid. * The expression evaluates to a priority number that is numerically lower than the current policy's priority. * The expression evaluates to a priority number that is between the current policy's priority number (say, 30) and the highest priority number (say, 100), but does not match any configured priority number (for

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
			example, the expression evaluates to the number 85). This example assumes that the priority number increments by 10 for every successive policy, and therefore a priority number of 85 does not exist in the policy label.
policy	<String>	Read-write	The name of the policy, if any, bound to the VPN virtual server.
groupextraction	<Boolean>	Read-write	Binds the authentication policy to a tertiary chain which will be used only for group extraction. The user will not authenticate against this server, and this will only be called if primary and/or secondary authentication has succeeded.
name	<String>	Read-write	Name of the virtual server. Minimum length = 1
secondary	<Boolean>	Read-write	Binds the authentication policy as the secondary policy to use in a two-factor

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
			configuration. A user must then authenticate not only via a primary authentication method but also via a secondary authentication method. User groups are aggregated across both. The user name must be exactly the same for both authentication methods, but they can require different passwords.
bindpoint	<String>	Read-write	Bind point to which to bind the policy. Applies only to rewrite and cache policies. If you do not set this parameter, the policy is bound to REQ_DEFAULT or RES_DEFAULT, depending on whether the policy rule is a response-time or a request-time expression. Possible values = REQUEST, RESPONSE, ICA_REQUEST, OTHERTCP_REQUEST
acttype	<Double>	Read-only	.

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
__count	<Double>	Read-only	count parameter

Operations

(click to see Properties)

ADD:| DELETE:| GET| GET (ALL)| COUNT

Some options that you can use for each operations:

- **Getting warnings in response:**NITRO allows you to get warnings in an operation by specifying the "warning" query parameter as "yes". For example, to get warnings while connecting to the Citrix ADC appliance, the URL is as follows:

  http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/login?warning=yes

  If any, the warnings are displayed in the response payload with the HTTP code "209 X-NITRO-WARNING".
- **Authenticated access for individual NITRO operations:**NITRO allows you to logon to the Citrix ADC appliance to perform individual operations. You can use this option instead of creating a NITRO session (using the login object) and then using that session to perform all operations,

  To do this, you must specify the username and password in the request header of the NITRO request as follows:

  X-NITRO-USER:<username>

  X-NITRO-PASS:<password>

  Note: In such cases, make sure that the request header DOES not include the following:

  Cookie:NITRO_AUTH_TOKEN=<tokenvalue>

`*Note:`*

Mandatory parameters are marked in redand placeholder content is marked in <green>.

add

URL:http://<Citrix-ADC-IP-address(NSIP)/nitro/v1/config/vpnvserver_authenticationcertpolicy_binding

HTTP Method:PUT

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Content-Type:application/json

Request Payload:

```
{

"vpnvserver_authenticationcertpolicy_binding":{

<b>"name":<String_value>,

</b>"policy":<String_value>,

"priority":<Double_value>,

"secondary":<Boolean_value>,

"groupextraction":<Boolean_value>,

"gotopriorityexpression":<String_value>,

"bindpoint":<String_value>

}}
```

Response:

HTTP Status Code on Success: 201 Created HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error

delete

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/vpnvserver_authenticationcertpolicy_binding/name_value<String>

Query-parameters:

args

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/vpnvserver_authenticationcertpolicy_binding/name_value<String>?**args=policy:<String_value>,secondary:<Boolean_value>,groupextraction:<Boolean_value>,bindpoint:<**

HTTP Method:DELETE

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue>

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error

get

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/vpnvserver_authenticationcertpolicy_binding/name_value<String>

Query-parameters:

filter

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/vpnvserver_authenticationcertpolicy_binding/name_value<String>?**filter=property-name1:property-value1,property-name2:property-value2**

Use this query-parameter to get the filtered set of vpnvserver_authenticationcertpolicy_binding resources configured on Citrix ADC. Filtering can be done on any of the properties of the resource.

pagination

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/vpnvserver_authenticationcertpolicy_binding/name_value<String>?**pagesize=#no;pageno=#no**

Use this query-parameter to get the vpnvserver_authenticationcertpolicy_binding resources in chunks.

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "vpnvserver_authenticationcertpolicy_binding": [ {

"priority":<Double_value>,

"gotopriorityexpression":<String_value>,

"policy":<String_value>,

"groupextraction":<Boolean_value>,

"name":<String_value>,

"secondary":<Boolean_value>,

"bindpoint":<String_value>,

"acttype":<Double_value>

}]}
```

get (all)

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/vpnvserver_authenticationcertpolicy_binding

Query-parameters:

bulkbindings

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/vpnvserver_authenticationcertpolicy_binding?**bulkbindings=yes**

NITRO allows you to fetch bindings in bulk.

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "vpnvserver_authenticationcertpolicy_binding": [ {

"priority":<Double_value>,

"gotopriorityexpression":<String_value>,

"policy":<String_value>,

"groupextraction":<Boolean_value>,

"name":<String_value>,

"secondary":<Boolean_value>,

"bindpoint":<String_value>,

"acttype":<Double_value>

}]}
```

count

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/vpnvserver_authenticationcertpolicy_binding/name_value<String>?**count=yes**

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{"vpnvserver_authenticationcertpolicy_binding": [ { "__count": "#no"} ] }
```

vpnvserver_authenticationdfapolicy_binding

Binding object showing the authenticationdfapolicy that can be bound to vpnvserver.

Properties

(click to see Operations)

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
priority	<Double>	Read-write	Integer specifying the policy's priority. The lower the number, the higher the priority. Policies are evaluated in the order of their priority numbers. Maximum

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
			value for default syntax policies is 2147483647 and for classic policies is 64000. Minimum value = 0 Maximum value = 2147483647
gotopriorityexpression	<String>	Read-write	Applicable only to advance vpn session policy. Expression or other value specifying the next policy to evaluate if the current policy evaluates to TRUE. Specify one of the following values: * NEXT - Evaluate the policy with the next higher priority number. * END - End policy evaluation. * An expression that evaluates to a number. If you specify an expression, the number to which it evaluates determines the next policy to evaluate, as follows: * If the expression evaluates to a higher numbered priority, the policy with that priority is evaluated next. * If the

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
			expression evaluates to the priority of the current policy, the policy with the next higher numbered priority is evaluated next. * If the expression evaluates to a number that is larger than the largest numbered priority, policy evaluation ends. An UNDEF event is triggered if: * The expression is invalid. * The expression evaluates to a priority number that is numerically lower than the current policy's priority. * The expression evaluates to a priority number that is between the current policy's priority number (say, 30) and the highest priority number (say, 100), but does not match any configured priority number (for example, the expression evaluates to the number 85). This example assumes that the priority number increments by 10 for every successive

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
			policy, and therefore a priority number of 85 does not exist in the policy label.
policy	<String>	Read-write	The name of the policy, if any, bound to the VPN virtual server.
groupextraction	<Boolean>	Read-write	Binds the authentication policy to a tertiary chain which will be used only for group extraction. The user will not authenticate against this server, and this will only be called if primary and/or secondary authentication has succeeded.
name	<String>	Read-write	Name of the virtual server. Minimum length = 1
secondary	<Boolean>	Read-write	Binds the authentication policy as the secondary policy to use in a two-factor configuration. A user must then authenticate not only via a primary authentication method but also via a secondary authentication

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
			method. User groups are aggregated across both. The user name must be exactly the same for both authentication methods, but they can require different passwords.
bindpoint	<String>	Read-write	Bind point to which to bind the policy. Applies only to rewrite and cache policies. If you do not set this parameter, the policy is bound to REQ_DEFAULT or RES_DEFAULT, depending on whether the policy rule is a response-time or a request-time expression. Possible values = REQUEST, RESPONSE, ICA_REQUEST, OTHERTCP_REQUEST
acttype	<Double>	Read-only	.
__count	<Double>	Read-only	count parameter

Operations

(click to see Properties)

ADD:| DELETE:| GET| GET (ALL)| COUNT

Some options that you can use for each operations:

- **Getting warnings in response:**NITRO allows you to get warnings in an operation by specifying the "warning" query parameter as "yes". For example, to get warnings while connecting to the Citrix ADC appliance, the URL is as follows:

  http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/login?warning=yes

  If any, the warnings are displayed in the response payload with the HTTP code "209 X-NITRO-WARNING".

- **Authenticated access for individual NITRO operations:**NITRO allows you to logon to the Citrix ADC appliance to perform individual operations. You can use this option instead of creating a NITRO session (using the login object) and then using that session to perform all operations,

  To do this, you must specify the username and password in the request header of the NITRO request as follows:

  X-NITRO-USER:<username>

  X-NITRO-PASS:<password>

  Note: In such cases, make sure that the request header DOES not include the following:

  Cookie:NITRO_AUTH_TOKEN=<tokenvalue>

`*Note:`*

Mandatory parameters are marked in redand placeholder content is marked in <green>.

add

URL:http://<Citrix-ADC-IP-address(NSIP)/nitro/v1/config/vpnvserver_authenticationdfapolicy_binding

HTTP Method:PUT

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Content-Type:application/json

Request Payload:

```
{

"vpnvserver_authenticationdfapolicy_binding":{

<b>"name":<String_value>,

</b>"policy":<String_value>,

"priority":<Double_value>,

"secondary":<Boolean_value>,

"groupextraction":<Boolean_value>,

"gotopriorityexpression":<String_value>,

"bindpoint":<String_value>

}}
```

Response:

HTTP Status Code on Success: 201 Created HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error

delete

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/vpnvserver_authenticationdfapolicy_binding/name_value<String>

Query-parameters:

args

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/vpnvserver_authenticationdfapolicy_binding/name_value<String>?**args=policy:<String_value>,secondary:<Boolean_value>,groupextraction:<Boolean_value>,bindpoint:<**

HTTP Method:DELETE

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue>

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error

get

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/vpnvserver_authenticationdfapolicy_binding/name_value<String>

Query-parameters:

filter

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/vpnvserver_authenticationdfapolicy_binding/name_value<String>?**filter=property-name1:property-value1,property-name2:property-value2**

Use this query-parameter to get the filtered set of vpnvserver_authenticationdfapolicy_binding resources configured on Citrix ADC. Filtering can be done on any of the properties of the resource.

pagination

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/vpnvserver_authenticationdfapolicy_binding/name_value<String>?**pagesize=#no;pageno=#no**

Use this query-parameter to get the vpnvserver_authenticationdfapolicy_binding resources in chunks.

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "vpnvserver_authenticationdfapolicy_binding": [ {

"priority":<Double_value>,

"gotopriorityexpression":<String_value>,

"policy":<String_value>,

"groupextraction":<Boolean_value>,

"name":<String_value>,

"secondary":<Boolean_value>,

"bindpoint":<String_value>,

"acttype":<Double_value>

}]}
```

get (all)

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/vpnvserver_authenticationdfapolicy_binding

Query-parameters:

bulkbindings

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/vpnvserver_authenticationdfapolicy_binding?**bulkbindings=yes**

NITRO allows you to fetch bindings in bulk.

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "vpnvserver_authenticationdfapolicy_binding": [ {

"priority":<Double_value>,

"gotopriorityexpression":<String_value>,

"policy":<String_value>,

"groupextraction":<Boolean_value>,

"name":<String_value>,

"secondary":<Boolean_value>,

"bindpoint":<String_value>,

"acttype":<Double_value>

}]}
```

count

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/vpnvserver_authenticationdfapolicy_binding/name_value<String>?**count=yes**

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{"vpnvserver_authenticationdfapolicy_binding": [ { "__count": "#no"} ] }
```

vpnvserver_authenticationldappolicy_binding

Binding object showing the authenticationldappolicy that can be bound to vpnvserver.

Properties

(click to see Operations)

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
priority	<Double>	Read-write	Integer specifying the policy's priority. The lower the number, the higher the priority. Policies are evaluated in the order of their priority numbers. Maximum value for default syntax policies is 2147483647 and for classic policies is 64000. Minimum value = 0

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
			Maximum value = 2147483647
gotopriorityexpression	<String>	Read-write	Applicable only to advance vpn session policy. Expression or other value specifying the next policy to evaluate if the current policy evaluates to TRUE. Specify one of the following values: * NEXT - Evaluate the policy with the next higher priority number. * END - End policy evaluation. * An expression that evaluates to a number. If you specify an expression, the number to which it evaluates determines the next policy to evaluate, as follows: * If the expression evaluates to a higher numbered priority, the policy with that priority is evaluated next. * If the expression evaluates to the priority of the current policy, the policy with the next higher numbered priority is evaluated next. * If the

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
			expression evaluates to a number that is larger than the largest numbered priority, policy evaluation ends. An UNDEF event is triggered if: * The expression is invalid. * The expression evaluates to a priority number that is numerically lower than the current policy's priority. * The expression evaluates to a priority number that is between the current policy's priority number (say, 30) and the highest priority number (say, 100), but does not match any configured priority number (for example, the expression evaluates to the number 85). This example assumes that the priority number increments by 10 for every successive policy, and therefore a priority number of 85 does not exist in the policy label.
policy	<String>	Read-write	The name of the policy, if any, bound

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
			to the VPN virtual server.
groupextraction	<Boolean>	Read-write	Binds the authentication policy to a tertiary chain which will be used only for group extraction. The user will not authenticate against this server, and this will only be called if primary and/or secondary authentication has succeeded.
name	<String>	Read-write	Name of the virtual server. Minimum length = 1
secondary	<Boolean>	Read-write	Binds the authentication policy as the secondary policy to use in a two-factor configuration. A user must then authenticate not only via a primary authentication method but also via a secondary authentication method. User groups are aggregated across both. The user name must be exactly the same for both authentication methods, but they

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
			can require different passwords.
bindpoint	<String>	Read-write	Bind point to which to bind the policy. Applies only to rewrite and cache policies. If you do not set this parameter, the policy is bound to REQ_DEFAULT or RES_DEFAULT, depending on whether the policy rule is a response-time or a request-time expression. Possible values = REQUEST, RESPONSE, ICA_REQUEST, OTHERTCP_REQUEST
acttype	<Double>	Read-only	.
__count	<Double>	Read-only	count parameter

Operations

(click to see Properties)

ADD:| DELETE:| GET| GET (ALL)| COUNT

Some options that you can use for each operations:

- **Getting warnings in response:**NITRO allows you to get warnings in an operation by specifying the "warning" query parameter as "yes". For example, to get warnings while connecting to the Citrix ADC appliance, the URL is as follows:

  http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/login?warning=yes

If any, the warnings are displayed in the response payload with the HTTP code "209 X-NITRO-WARNING".

- **Authenticated access for individual NITRO operations:**NITRO allows you to logon to the Citrix ADC appliance to perform individual operations. You can use this option instead of creating a NITRO session (using the login object) and then using that session to perform all operations,

  To do this, you must specify the username and password in the request header of the NITRO request as follows:

  X-NITRO-USER:<username>

  X-NITRO-PASS:<password>

  Note: In such cases, make sure that the request header DOES not include the following:

  Cookie:NITRO_AUTH_TOKEN=<tokenvalue>

`*Note:`*

Mandatory parameters are marked in redand placeholder content is marked in <green>.

add

URL:http://<Citrix-ADC-IP-address(NSIP)/nitro/v1/config/vpnvserver_authenticationldappolicy_binding

HTTP Method:PUT

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Content-Type:application/json

Request Payload:

```
{

"vpnvserver_authenticationldappolicy_binding":{

<b>"name":<String_value>,

</b>"policy":<String_value>,
```

```
"priority":<Double_value>,

"secondary":<Boolean_value>,

"groupextraction":<Boolean_value>,

"gotopriorityexpression":<String_value>,

"bindpoint":<String_value>

}}
```

Response:

HTTP Status Code on Success: 201 Created HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error

delete

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/vpnvserver_authenticationldappolicy_binding/name_value<String>

Query-parameters:

args

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/vpnvserver_authenticationldappolicy_binding/name_value<String>?
args=policy:<String_value>,secondary:<Boolean_value>,groupextraction:<Boolean_value>,bindpoint:<

HTTP Method:DELETE

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue>

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error

get

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/vpnvserver_authenticationldappolicy_binding/name_value<String>

Query-parameters:

filter

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/vpnvserver_authenticationldappolicy_binding/name_value<String>?**filter=property-name1:property-value1,property-name2:property-value2**

Use this query-parameter to get the filtered set of vpnvserver_authenticationldappolicy_binding resources configured on Citrix ADC. Filtering can be done on any of the properties of the resource.

pagination

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/vpnvserver_authenticationldappolicy_binding/name_value<String>?**pagesize=#no;pageno=#no**

Use this query-parameter to get the vpnvserver_authenticationldappolicy_binding resources in chunks.

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "vpnvserver_authenticationldappolicy_binding": [ {

"priority":<Double_value>,
```

```
"gotopriorityexpression":<String_value>,

"policy":<String_value>,

"groupextraction":<Boolean_value>,

"name":<String_value>,

"secondary":<Boolean_value>,

"bindpoint":<String_value>,

"acttype":<Double_value>

}]}
```

get (all)

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/vpnvserver_authenticationldappolicy_binding

Query-parameters:

bulkbindings

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/vpnvserver_authenticationldappolicy_binding?**bulkbindings=yes**

NITRO allows you to fetch bindings in bulk.

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "vpnvserver_authenticationldappolicy_binding": [ {

"priority":<Double_value>,

"gotopriorityexpression":<String_value>,

"policy":<String_value>,

"groupextraction":<Boolean_value>,

"name":<String_value>,

"secondary":<Boolean_value>,

"bindpoint":<String_value>,

"acttype":<Double_value>

}]}
```

count

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/vpnvserver_authenticationldappolicy_binding/name_value<String>?**count=yes**

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{"vpnvserver_authenticationldappolicy_binding": [ { "__count": "#no"} ] }
```

vpnvserver_authenticationlocalpolicy_binding

Binding object showing the authenticationlocalpolicy that can be bound to vpnvserver.

Properties

(click to see Operations)

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
priority	<Double>	Read-write	Integer specifying the policy's priority. The lower the number, the higher the priority. Policies are evaluated in the order of their priority numbers. Maximum value for default syntax policies is 2147483647 and for classic policies is 64000. Minimum value = 0 Maximum value = 2147483647
gotopriorityexpression	<String>	Read-write	Applicable only to advance vpn session policy. Expression or other value specifying the next policy to evaluate if the current policy evaluates to TRUE.

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
			Specify one of the following values: * NEXT - Evaluate the policy with the next higher priority number. * END - End policy evaluation. * An expression that evaluates to a number. If you specify an expression, the number to which it evaluates determines the next policy to evaluate, as follows: * If the expression evaluates to a higher numbered priority, the policy with that priority is evaluated next. * If the expression evaluates to the priority of the current policy, the policy with the next higher numbered priority is evaluated next. * If the expression evaluates to a number that is larger than the largest numbered priority, policy evaluation ends. An UNDEF event is triggered if: * The expression is invalid. * The expression evaluates to a

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
			priority number that is numerically lower than the current policy's priority. * The expression evaluates to a priority number that is between the current policy's priority number (say, 30) and the highest priority number (say, 100), but does not match any configured priority number (for example, the expression evaluates to the number 85). This example assumes that the priority number increments by 10 for every successive policy, and therefore a priority number of 85 does not exist in the policy label.
policy	<String>	Read-write	The name of the policy, if any, bound to the VPN virtual server.
groupextraction	<Boolean>	Read-write	Binds the authentication policy to a tertiary chain which will be used only for group extraction. The user will not authenticate against this server,

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
			and this will only be called if primary and/or secondary authentication has succeeded.
name	<String>	Read-write	Name of the virtual server. Minimum length = 1
secondary	<Boolean>	Read-write	Binds the authentication policy as the secondary policy to use in a two-factor configuration. A user must then authenticate not only via a primary authentication method but also via a secondary authentication method. User groups are aggregated across both. The user name must be exactly the same for both authentication methods, but they can require different passwords.
bindpoint	<String>	Read-write	Bind point to which to bind the policy. Applies only to rewrite and cache policies. If you do not set this parameter, the policy is bound to REQ_DEFAULT or

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
			RES_DEFAULT, depending on whether the policy rule is a response-time or a request-time expression. Possible values = REQUEST, RESPONSE, ICA_REQUEST, OTHERTCP_REQUEST
acttype	<Double>	Read-only	.
__count	<Double>	Read-only	count parameter

Operations

(click to see Properties)

ADD:| DELETE:| GET| GET (ALL)| COUNT

Some options that you can use for each operations:

- **Getting warnings in response:**NITRO allows you to get warnings in an operation by specifying the "warning" query parameter as "yes". For example, to get warnings while connecting to the Citrix ADC appliance, the URL is as follows:

  http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/login?warning=yes

  If any, the warnings are displayed in the response payload with the HTTP code "209 X-NITRO-WARNING".

- **Authenticated access for individual NITRO operations:**NITRO allows you to logon to the Citrix ADC appliance to perform individual operations. You can use this option instead of creating a NITRO session (using the login object) and then using that session to perform all operations,

  To do this, you must specify the username and password in the request header of the NITRO request as follows:

X-NITRO-USER:<username>

X-NITRO-PASS:<password>

Note: In such cases, make sure that the request header DOES not include the following:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue>

`*Note:`*

Mandatory parameters are marked in redand placeholder content is marked in <green>.

add

URL:http://<Citrix-ADC-IP-address(NSIP)/nitro/v1/config/vpnvserver_authenticationlocalpolicy_binding

HTTP Method:PUT

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Content-Type:application/json

Request Payload:

```
{

"vpnvserver_authenticationlocalpolicy_binding":{

<b>"name":<String_value>,

</b>"policy":<String_value>,

"priority":<Double_value>,

"secondary":<Boolean_value>,

"groupextraction":<Boolean_value>,

"gotopriorityexpression":<String_value>,

"bindpoint":<String_value>
```

```
}}
```

Response:

HTTP Status Code on Success: 201 Created HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error

delete

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/vpnvserver_authenticationlocalpolicy_binding/name_value<String>

Query-parameters:

args

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/vpnvserver_authenticationlocalpolicy_binding/name_value<String>?
args=policy:<String_value>,secondary:<Boolean_value>,groupextraction:<Boolean_value>,bindpoint:<

HTTP Method:DELETE

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue>

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error

get

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/vpnvserver_authenticationlocalpolicy_binding/name_value<String>

Query-parameters:

filter

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/vpnvserver_authenticationlocalpolicy_binding/name_value<String>?**filter=property-name1:property-value1,property-name2:property-value2**

Use this query-parameter to get the filtered set of vpnvserver_authenticationlocalpolicy_binding resources configured on Citrix ADC. Filtering can be done on any of the properties of the resource.

pagination

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/vpnvserver_authenticationlocalpolicy_binding/name_value<String>?**pagesize=#no;pageno=#no**

Use this query-parameter to get the vpnvserver_authenticationlocalpolicy_binding resources in chunks.

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "vpnvserver_authenticationlocalpolicy_binding": [ {

"priority":<Double_value>,

"gotopriorityexpression":<String_value>,

"policy":<String_value>,

"groupextraction":<Boolean_value>,

"name":<String_value>,
```

```
"secondary":<Boolean_value>,

"bindpoint":<String_value>,

"acttype":<Double_value>

}]}
```

get (all)

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/vpnvserver_authenticationlocalpolicy_binding

Query-parameters:

bulkbindings

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/vpnvserver_authenticationlocalpolicy_binding?**bulkbindings=yes**

NITRO allows you to fetch bindings in bulk.

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "vpnvserver_authenticationlocalpolicy_binding": [ {

"priority":<Double_value>,
```

```
"gotopriorityexpression":<String_value>,

"policy":<String_value>,

"groupextraction":<Boolean_value>,

"name":<String_value>,

"secondary":<Boolean_value>,

"bindpoint":<String_value>,

"acttype":<Double_value>

}]}
```

count

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/vpnvserver_authenticationlocalpolicy_binding/name_value<String>?**count=yes**

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{"vpnvserver_authenticationlocalpolicy_binding": [ { "__count": "#no"} ] }
```

vpnvserver_authenticationloginschemapolicy_binding

Binding object showing the authenticationloginschemapolicy that can be bound to vpnvserver.

Properties

(click to see Operations)

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
priority	<Double>	Read-write	Integer specifying the policy's priority. The lower the number, the higher the priority. Policies are evaluated in the order of their priority numbers. Maximum value for default syntax policies is 2147483647 and for classic policies is 64000. Minimum value = 0 Maximum value = 2147483647
gotopriorityexpression	<String>	Read-write	Next priority expression.
policy	<String>	Read-write	The name of the policy, if any, bound to the VPN virtual server.
groupextraction	<Boolean>	Read-write	Binds the authentication policy to a tertiary chain which will be used only for group extraction. The user will not authenticate against this server,

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
			and this will only be called if primary and/or secondary authentication has succeeded.
name	<String>	Read-write	Name of the virtual server. Minimum length = 1
secondary	<Boolean>	Read-write	Binds the authentication policy as the secondary policy to use in a two-factor configuration. A user must then authenticate not only via a primary authentication method but also via a secondary authentication method. User groups are aggregated across both. The user name must be exactly the same for both authentication methods, but they can require different passwords.
bindpoint	<String>	Read-write	Bind point to which to bind the policy. Applies only to rewrite and cache policies. If you do not set this parameter, the policy is bound to REQ_DEFAULT or

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
			RES_DEFAULT, depending on whether the policy rule is a response-time or a request-time expression. Possible values = REQUEST, RESPONSE, ICA_REQUEST, OTHERTCP_REQUEST
__count	<Double>	Read-write	count parameter

Operations

(click to see Properties)

ADD:| DELETE:| GET| GET (ALL)| COUNT

Some options that you can use for each operations:

- **Getting warnings in response:**NITRO allows you to get warnings in an operation by specifying the "warning" query parameter as "yes". For example, to get warnings while connecting to the Citrix ADC appliance, the URL is as follows:

  http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/login?warning=yes

  If any, the warnings are displayed in the response payload with the HTTP code "209 X-NITRO-WARNING".

- **Authenticated access for individual NITRO operations:**NITRO allows you to logon to the Citrix ADC appliance to perform individual operations. You can use this option instead of creating a NITRO session (using the login object) and then using that session to perform all operations,

  To do this, you must specify the username and password in the request header of the NITRO request as follows:

  X-NITRO-USER:<username>

X-NITRO-PASS:<password>

Note: In such cases, make sure that the request header DOES not include the following:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue>

`*Note:`*

Mandatory parameters are marked in redand placeholder content is marked in <green>.

add

URL:http://<Citrix-ADC-IP-address(NSIP)/nitro/v1/config/vpnvserver_authenticationloginschemapolicy_binding

HTTP Method:PUT

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Content-Type:application/json

Request Payload:

```
{
"vpnvserver_authenticationloginschemapolicy_binding":{
<b>"name":<String_value>,
</b>"policy":<String_value>,
"priority":<Double_value>,
"secondary":<Boolean_value>,
"groupextraction":<Boolean_value>,
"gotopriorityexpression":<String_value>,
"bindpoint":<String_value>
}}
```

Response:

HTTP Status Code on Success: 201 Created HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error

delete

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/vpnvserver_authenticationloginschemapolicy_binding/name_value<String>

Query-parameters:

args

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/vpnvserver_authenticationloginschemapolicy_binding/name_value<String>?**args=policy:<String_value>,secondary:<Boolean_value>,groupextraction:<Boolean_value>,bindpoint:<**

HTTP Method:DELETE

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue>

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error

get

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/vpnvserver_authenticationloginschemapolicy_binding/name_value<String>

Query-parameters:

filter

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/vpnvserver_authenticationloginschemapolicy_binding/name_value<String>?**filter=property-name1:property-value1,property-name2:property-value2**

Use this query-parameter to get the filtered set of vpnvserver_authenticationloginschemapolicy_binding resources configured on Citrix ADC. Filtering can be done on any of the properties of the resource.

pagination

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/vpnvserver_authenticationloginschemapolicy_binding/name_value<String>?**pagesize=#no;pageno=#no**

Use this query-parameter to get the vpnvserver_authenticationloginschemapolicy_binding resources in chunks.

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "vpnvserver_authenticationloginschemapolicy_binding": [ {

"priority":<Double_value>,

"gotopriorityexpression":<String_value>,

"policy":<String_value>,

"groupextraction":<Boolean_value>,

"name":<String_value>,

"secondary":<Boolean_value>,
```

```
"bindpoint":<String_value>

}]}
```

get (all)

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/vpnvserver_authenticationloginschemapolicy_binding

Query-parameters:

bulkbindings

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/vpnvserver_authenticationloginschemapolicy_binding?**bulkbindings=yes**

NITRO allows you to fetch bindings in bulk.

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "vpnvserver_authenticationloginschemapolicy_binding": [ {

"priority":<Double_value>,

"gotopriorityexpression":<String_value>,

"policy":<String_value>,
```

```
"groupextraction":<Boolean_value>,

"name":<String_value>,

"secondary":<Boolean_value>,

"bindpoint":<String_value>

}]}
```

count

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/vpnvserver_authenticationloginschemapolicy_binding/name_value<String>?**count=yes**

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{"vpnvserver_authenticationloginschemapolicy_binding": [ { "__count":
"#no"} ] }
```

vpnvserver_authenticationnegotiatepolicy_binding

Binding object showing the authenticationnegotiatepolicy that can be bound to vpnvserver.

Properties

(click to see Operations)

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
priority	<Double>	Read-write	Integer specifying the policy's priority. The lower the number, the higher the priority. Policies are evaluated in the order of their priority numbers. Maximum value for default syntax policies is 2147483647 and for classic policies is 64000. Minimum value = 0 Maximum value = 2147483647
gotopriorityexpression	<String>	Read-write	Applicable only to advance vpn session policy. Expression or other value specifying the next policy to evaluate if the current policy evaluates to TRUE. Specify one of the following values: * NEXT - Evaluate the policy with the next higher priority number. * END - End policy evaluation. * An expression that evaluates to a number. If you specify an

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
			expression, the number to which it evaluates determines the next policy to evaluate, as follows: * If the expression evaluates to a higher numbered priority, the policy with that priority is evaluated next. * If the expression evaluates to the priority of the current policy, the policy with the next higher numbered priority is evaluated next. * If the expression evaluates to a number that is larger than the largest numbered priority, policy evaluation ends. An UNDEF event is triggered if: * The expression is invalid. * The expression evaluates to a priority number that is numerically lower than the current policy's priority. * The expression evaluates to a priority number that is between the current policy's priority number (say, 30) and the highest priority number (say,

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
			100), but does not match any configured priority number (for example, the expression evaluates to the number 85). This example assumes that the priority number increments by 10 for every successive policy, and therefore a priority number of 85 does not exist in the policy label.
policy	<String>	Read-write	The name of the policy, if any, bound to the VPN virtual server.
groupextraction	<Boolean>	Read-write	Binds the authentication policy to a tertiary chain which will be used only for group extraction. The user will not authenticate against this server, and this will only be called if primary and/or secondary authentication has succeeded.
name	<String>	Read-write	Name of the virtual server. Minimum length = 1

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
secondary	<Boolean>	Read-write	Binds the authentication policy as the secondary policy to use in a two-factor configuration. A user must then authenticate not only via a primary authentication method but also via a secondary authentication method. User groups are aggregated across both. The user name must be exactly the same for both authentication methods, but they can require different passwords.
bindpoint	<String>	Read-write	Bind point to which to bind the policy. Applies only to rewrite and cache policies. If you do not set this parameter, the policy is bound to REQ_DEFAULT or RES_DEFAULT, depending on whether the policy rule is a response-time or a request-time expression. Possible values = REQUEST, RESPONSE,

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
			ICA_REQUEST, OTHERTCP_REQUEST
acttype	<Double>	Read-only	.
__count	<Double>	Read-only	count parameter

Operations

(click to see Properties)

ADD:| DELETE:| GET| GET (ALL)| COUNT

Some options that you can use for each operations:

- **Getting warnings in response:**NITRO allows you to get warnings in an operation by specifying the "warning" query parameter as "yes". For example, to get warnings while connecting to the Citrix ADC appliance, the URL is as follows:

  http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/login?warning=yes

  If any, the warnings are displayed in the response payload with the HTTP code "209 X-NITRO-WARNING".

- **Authenticated access for individual NITRO operations:**NITRO allows you to logon to the Citrix ADC appliance to perform individual operations. You can use this option instead of creating a NITRO session (using the login object) and then using that session to perform all operations,

  To do this, you must specify the username and password in the request header of the NITRO request as follows:

  X-NITRO-USER:<username>

  X-NITRO-PASS:<password>

  Note: In such cases, make sure that the request header DOES not include the following:

  Cookie:NITRO_AUTH_TOKEN=<tokenvalue>

`*Note:`*

Mandatory parameters are marked in redand placeholder content is marked in <green>.

add

URL:http://<Citrix-ADC-IP-address(NSIP)/nitro/v1/config/vpnvserver_authenticationnegotiatepolicy_binding

HTTP Method:PUT

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Content-Type:application/json

Request Payload:

```
{

"vpnvserver_authenticationnegotiatepolicy_binding":{

<b>"name":<String_value>,

</b>"policy":<String_value>,

"priority":<Double_value>,

"secondary":<Boolean_value>,

"groupextraction":<Boolean_value>,

"gotopriorityexpression":<String_value>,

"bindpoint":<String_value>

}}
```

Response:

HTTP Status Code on Success: 201 Created HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error

delete

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/vpnvserver_authenticationnegotiatepolicy_binding/name_value<String>

Query-parameters:

args

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/vpnvserver_authenticationnegotiatepolicy_binding/name_value<String>?**args=policy:<String_value>,secondary:<Boolean_value>,groupextraction:<Boolean_value>,bindpoint:<**

HTTP Method:DELETE

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue>

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error

get

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/vpnvserver_authenticationnegotiatepolicy_binding/name_value<String>

Query-parameters:

filter

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/vpnvserver_authenticationnegotiatepolicy_binding/name_value<String>?**filter=property-name1:property-value1,property-name2:property-value2**

Use this query-parameter to get the filtered set of vpnvserver_authenticationnegotiatepolicy_binding resources configured on Citrix ADC. Filtering can be done on any of the properties of the resource.

pagination

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/vpnvserver_authenticationnegotiatepolicy_binding/name_value<String>?**pagesize=#no;pageno=#no**

Use this query-parameter to get the vpnvserver_authenticationnegotiatepolicy_binding resources in chunks.

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "vpnvserver_authenticationnegotiatepolicy_binding": [ {

"priority":<Double_value>,

"gotopriorityexpression":<String_value>,

"policy":<String_value>,

"groupextraction":<Boolean_value>,

"name":<String_value>,

"secondary":<Boolean_value>,

"bindpoint":<String_value>,

"acttype":<Double_value>

}]}
```

get (all)

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/vpnvserver_authenticationnegotiatepolicy_binding

Query-parameters:

bulkbindings

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/vpnvserver_authenticationnegotiatepolicy_binding?**bulkbindings=yes**

NITRO allows you to fetch bindings in bulk.

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "vpnvserver_authenticationnegotiatepolicy_binding": [ {

"priority":<Double_value>,

"gotopriorityexpression":<String_value>,

"policy":<String_value>,

"groupextraction":<Boolean_value>,

"name":<String_value>,
```

```
"secondary":<Boolean_value>,

"bindpoint":<String_value>,

"acttype":<Double_value>

}]}
```

count

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/vpnvserver_authenticationnegotiatepolicy_binding/name_value<String>?**count=yes**

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{"vpnvserver_authenticationnegotiatepolicy_binding": [ { "__count":
"#no"} ] }
```

vpnvserver_authenticationpolicy_binding

Binding object showing the authenticationpolicy that can be bound to vpnvserver.

Properties

(click to see Operations)

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
priority	<Double>	Read-write	Integer specifying the policy's priority. The lower the number, the higher the priority. Policies are evaluated in the order of their priority numbers. Maximum value for default syntax policies is 2147483647 and for classic policies is 64000. Minimum value = 0 Maximum value = 2147483647
gotopriorityexpression	<String>	Read-write	Applicable only to advance vpn session policy. Expression or other value specifying the next policy to evaluate if the current policy evaluates to TRUE. Specify one of the following values: * NEXT - Evaluate the policy with the next higher priority number. * END - End policy evaluation. * An expression that evaluates to a number. If you specify an expression, the number to which it evaluates determines the next

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
			policy to evaluate, as follows: * If the expression evaluates to a higher numbered priority, the policy with that priority is evaluated next. * If the expression evaluates to the priority of the current policy, the policy with the next higher numbered priority is evaluated next. * If the expression evaluates to a number that is larger than the largest numbered priority, policy evaluation ends. An UNDEF event is triggered if: * The expression is invalid. * The expression evaluates to a priority number that is numerically lower than the current policy's priority. * The expression evaluates to a priority number that is between the current policy's priority number (say, 30) and the highest priority number (say, 100), but does not match any configured priority number (for

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
			example, the expression evaluates to the number 85). This example assumes that the priority number increments by 10 for every successive policy, and therefore a priority number of 85 does not exist in the policy label.
policy	<String>	Read-write	The name of the policy, if any, bound to the VPN virtual server.
groupextraction	<Boolean>	Read-write	Binds the authentication policy to a tertiary chain which will be used only for group extraction. The user will not authenticate against this server, and this will only be called if primary and/or secondary authentication has succeeded.
name	<String>	Read-write	Name of the virtual server. Minimum length = 1
secondary	<Boolean>	Read-write	Binds the authentication policy as the secondary policy to use in a two-factor

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
			configuration. A user must then authenticate not only via a primary authentication method but also via a secondary authentication method. User groups are aggregated across both. The user name must be exactly the same for both authentication methods, but they can require different passwords.
bindpoint	<String>	Read-write	Bind point to which to bind the policy. Applies only to rewrite and cache policies. If you do not set this parameter, the policy is bound to REQ_DEFAULT or RES_DEFAULT, depending on whether the policy rule is a response-time or a request-time expression. Possible values = REQUEST, RESPONSE, ICA_REQUEST, OTHERTCP_REQUEST
acttype	<Double>	Read-only	.

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
__count	<Double>	Read-only	count parameter

Operations

(click to see Properties)

ADD:| DELETE:| GET| GET (ALL)| COUNT

Some options that you can use for each operations:

- **Getting warnings in response:**NITRO allows you to get warnings in an operation by specifying the "warning" query parameter as "yes". For example, to get warnings while connecting to the Citrix ADC appliance, the URL is as follows:

  http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/login?warning=yes

  If any, the warnings are displayed in the response payload with the HTTP code "209 X-NITRO-WARNING".
- **Authenticated access for individual NITRO operations:**NITRO allows you to logon to the Citrix ADC appliance to perform individual operations. You can use this option instead of creating a NITRO session (using the login object) and then using that session to perform all operations,

  To do this, you must specify the username and password in the request header of the NITRO request as follows:

  X-NITRO-USER:<username>

  X-NITRO-PASS:<password>

  Note: In such cases, make sure that the request header DOES not include the following:

  Cookie:NITRO_AUTH_TOKEN=<tokenvalue>

`*Note:`*

Mandatory parameters are marked in redand placeholder content is marked in <green>.

add

URL:http://<Citrix-ADC-IP-address(NSIP)/nitro/v1/config/vpnvserver_authenticationpolicy_binding

HTTP Method:PUT

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Content-Type:application/json

Request Payload:

```
{

"vpnvserver_authenticationpolicy_binding":{

<b>"name":<String_value>,

</b>"policy":<String_value>,

"priority":<Double_value>,

"secondary":<Boolean_value>,

"groupextraction":<Boolean_value>,

"gotopriorityexpression":<String_value>,

"bindpoint":<String_value>

}}
```

Response:

HTTP Status Code on Success: 201 Created HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error

delete

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/vpnvserver_authenticationpolicy_binding/name_value<String>

Query-parameters:

args

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/vpnvserver_authenticationpolicy_binding/name_value<String>?**args=policy:<String_value>,secondary:<Boolean_value>,groupextraction:<Boolean_value>,bindpoint:<**

HTTP Method:DELETE

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue>

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error

get

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/vpnvserver_authenticationpolicy_binding/name_value<String>

Query-parameters:

filter

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/vpnvserver_authenticationpolicy_binding/name_value<String>?**filter=property-name1:property-value1,property-name2:property-value2**

Use this query-parameter to get the filtered set of vpnvserver_authenticationpolicy_binding resources configured on Citrix ADC. Filtering can be done on any of the properties of the resource.

pagination

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/vpnvserver_authenticationpolicy_binding/name_value<String>?**pagesize=#no;pageno=#no**

Use this query-parameter to get the vpnvserver_authenticationpolicy_binding resources in chunks.

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "vpnvserver_authenticationpolicy_binding": [ {

"priority":<Double_value>,

"gotopriorityexpression":<String_value>,

"policy":<String_value>,

"groupextraction":<Boolean_value>,

"name":<String_value>,

"secondary":<Boolean_value>,

"bindpoint":<String_value>,

"acttype":<Double_value>

}]}
```

get (all)

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/vpnvserver_authenticationpolicy_binding

Query-parameters:

bulkbindings

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/vpnvserver_authenticationpolicy_binding?**bulkbindings=yes**

NITRO allows you to fetch bindings in bulk.

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "vpnvserver_authenticationpolicy_binding": [ {

"priority":<Double_value>,

"gotopriorityexpression":<String_value>,

"policy":<String_value>,

"groupextraction":<Boolean_value>,

"name":<String_value>,

"secondary":<Boolean_value>,

"bindpoint":<String_value>,

"acttype":<Double_value>

}]}
```

count

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/vpnvserver_authenticationpolicy_binding/name_value<String>?**count=yes**

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{"vpnvserver_authenticationpolicy_binding": [ { "__count": "#no"} ] }
```

vpnvserver_authenticationradiuspolicy_binding

Binding object showing the authenticationradiuspolicy that can be bound to vpnvserver.

Properties

(click to see Operations)

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
priority	<Double>	Read-write	Integer specifying the policy's priority. The lower the number, the higher the priority. Policies are evaluated in the order of their priority numbers. Maximum value for default syntax policies is 2147483647 and for

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
			classic policies is 64000. Minimum value = 0 Maximum value = 2147483647
gotopriorityexpression	<String>	Read-write	Applicable only to advance vpn session policy. Expression or other value specifying the next policy to evaluate if the current policy evaluates to TRUE. Specify one of the following values: * NEXT - Evaluate the policy with the next higher priority number. * END - End policy evaluation. * An expression that evaluates to a number. If you specify an expression, the number to which it evaluates determines the next policy to evaluate, as follows: * If the expression evaluates to a higher numbered priority, the policy with that priority is evaluated next. * If the expression evaluates to the priority of the current policy, the

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
			policy with the next higher numbered priority is evaluated next. * If the expression evaluates to a number that is larger than the largest numbered priority, policy evaluation ends. An UNDEF event is triggered if: * The expression is invalid. * The expression evaluates to a priority number that is numerically lower than the current policy's priority. * The expression evaluates to a priority number that is between the current policy's priority number (say, 30) and the highest priority number (say, 100), but does not match any configured priority number (for example, the expression evaluates to the number 85). This example assumes that the priority number increments by 10 for every successive policy, and therefore a priority number of

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
			85 does not exist in the policy label.
policy	<String>	Read-write	The name of the policy, if any, bound to the VPN virtual server.
groupextraction	<Boolean>	Read-write	Binds the authentication policy to a tertiary chain which will be used only for group extraction. The user will not authenticate against this server, and this will only be called if primary and/or secondary authentication has succeeded.
name	<String>	Read-write	Name of the virtual server. Minimum length = 1
secondary	<Boolean>	Read-write	Binds the authentication policy as the secondary policy to use in a two-factor configuration. A user must then authenticate not only via a primary authentication method but also via a secondary authentication method. User groups are aggregated

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
			across both. The user name must be exactly the same for both authentication methods, but they can require different passwords.
bindpoint	<String>	Read-write	Bind point to which to bind the policy. Applies only to rewrite and cache policies. If you do not set this parameter, the policy is bound to REQ_DEFAULT or RES_DEFAULT, depending on whether the policy rule is a response-time or a request-time expression. Possible values = REQUEST, RESPONSE, ICA_REQUEST, OTHERTCP_REQUEST
acttype	<Double>	Read-only	.
__count	<Double>	Read-only	count parameter

Operations

(click to see Properties)

ADD:| DELETE:| GET| GET (ALL)| COUNT

Some options that you can use for each operations:

- **Getting warnings in response:**NITRO allows you to get warnings in an operation by specifying the "warning" query parameter as "yes". For example, to get warnings while connecting to the Citrix ADC appliance, the URL is as follows:

  http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/login?warning=yes

  If any, the warnings are displayed in the response payload with the HTTP code "209 X-NITRO-WARNING".

- **Authenticated access for individual NITRO operations:**NITRO allows you to logon to the Citrix ADC appliance to perform individual operations. You can use this option instead of creating a NITRO session (using the login object) and then using that session to perform all operations,

  To do this, you must specify the username and password in the request header of the NITRO request as follows:

  X-NITRO-USER:<username>

  X-NITRO-PASS:<password>

  Note: In such cases, make sure that the request header DOES not include the following:

  Cookie:NITRO_AUTH_TOKEN=<tokenvalue>

`*Note:`*

Mandatory parameters are marked in redand placeholder content is marked in <green>.

add

URL:http://<Citrix-ADC-IP-address(NSIP)/nitro/v1/config/vpnvserver_authenticationradiuspolicy_binding

HTTP Method:PUT

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Content-Type:application/json

Request Payload:

```
{
```

```
"vpnvserver_authenticationradiuspolicy_binding":{

<b>"name":<String_value>,

</b>"policy":<String_value>,

"priority":<Double_value>,

"secondary":<Boolean_value>,

"groupextraction":<Boolean_value>,

"gotopriorityexpression":<String_value>,

"bindpoint":<String_value>

}}
```

Response:

HTTP Status Code on Success: 201 Created HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error

delete

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/vpnvserver_authenticationradiuspolicy_binding/name_value<String>

Query-parameters:

args

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/vpnvserver_authenticationradiuspolicy_binding/name_value<String>?**args=policy:<String_value>,secondary:<Boolean_value>,groupextraction:<Boolean_value>,bindpoint:<**

HTTP Method:DELETE

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue>

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error

get

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/vpnvserver_authenticationradiuspolicy_binding/name_value<String>

Query-parameters:

filter

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/vpnvserver_authenticationradiuspolicy_binding/name_value<String>?**filter=property-name1:property-value1,property-name2:property-value2**

Use this query-parameter to get the filtered set of vpnvserver_authenticationradiuspolicy_binding resources configured on Citrix ADC. Filtering can be done on any of the properties of the resource.

pagination

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/vpnvserver_authenticationradiuspolicy_binding/name_value<String>?**pagesize=#no;pageno=#no**

Use this query-parameter to get the vpnvserver_authenticationradiuspolicy_binding resources in chunks.

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "vpnvserver_authenticationradiuspolicy_binding": [ {

"priority":<Double_value>,

"gotopriorityexpression":<String_value>,

"policy":<String_value>,

"groupextraction":<Boolean_value>,

"name":<String_value>,

"secondary":<Boolean_value>,

"bindpoint":<String_value>,

"acttype":<Double_value>

}]}
```

get (all)

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/vpnvserver_authenticationradiuspolicy_binding

Query-parameters:

bulkbindings

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/vpnvserver_authenticationradiuspolicy_binding?**bulkbindings=yes**

NITRO allows you to fetch bindings in bulk.

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "vpnvserver_authenticationradiuspolicy_binding": [ {

"priority":<Double_value>,

"gotopriorityexpression":<String_value>,

"policy":<String_value>,

"groupextraction":<Boolean_value>,

"name":<String_value>,

"secondary":<Boolean_value>,

"bindpoint":<String_value>,

"acttype":<Double_value>

}]}
```

count

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/vpnvserver_authenticationradiuspolicy_binding/name_value<String>?**count=yes**

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{"vpnvserver_authenticationradiuspolicy_binding": [ { "__count":
"#no"} ] }
```

vpnvserver_authenticationsamlidppolicy_binding

Binding object showing the authenticationsamlidppolicy that can be bound to vpnvserver.

Properties

(click to see Operations)

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
priority	<Double>	Read-write	Integer specifying the policy's priority. The lower the number, the higher the priority. Policies are evaluated in the order of their priority numbers. Maximum value for default syntax policies is 2147483647 and for classic policies is 64000. Minimum value = 0 Maximum value = 2147483647

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
gotopriorityexpression	<String>	Read-write	Next priority expression.
policy	<String>	Read-write	The name of the policy, if any, bound to the VPN virtual server.
groupextraction	<Boolean>	Read-write	Binds the authentication policy to a tertiary chain which will be used only for group extraction. The user will not authenticate against this server, and this will only be called if primary and/or secondary authentication has succeeded.
name	<String>	Read-write	Name of the virtual server. Minimum length = 1
secondary	<Boolean>	Read-write	Binds the authentication policy as the secondary policy to use in a two-factor configuration. A user must then authenticate not only via a primary authentication method but also via a secondary authentication method. User groups are aggregated

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
			across both. The user name must be exactly the same for both authentication methods, but they can require different passwords.
bindpoint	<String>	Read-write	Bind point to which to bind the policy. Applies only to rewrite and cache policies. If you do not set this parameter, the policy is bound to REQ_DEFAULT or RES_DEFAULT, depending on whether the policy rule is a response-time or a request-time expression. Possible values = REQUEST, RESPONSE, ICA_REQUEST, OTHERTCP_REQUEST
__count	<Double>	Read-write	count parameter

Operations

(click to see Properties)

ADD:| DELETE:| GET| GET (ALL)| COUNT

Some options that you can use for each operations:

- **Getting warnings in response:**NITRO allows you to get warnings in an operation by specifying the "warning" query parameter as "yes". For example, to get warnings while connecting to the Citrix ADC appliance, the URL is as follows:

  http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/login?warning=yes

  If any, the warnings are displayed in the response payload with the HTTP code "209 X-NITRO-WARNING".

- **Authenticated access for individual NITRO operations:**NITRO allows you to logon to the Citrix ADC appliance to perform individual operations. You can use this option instead of creating a NITRO session (using the login object) and then using that session to perform all operations,

  To do this, you must specify the username and password in the request header of the NITRO request as follows:

  X-NITRO-USER:<username>

  X-NITRO-PASS:<password>

  Note: In such cases, make sure that the request header DOES not include the following:

  Cookie:NITRO_AUTH_TOKEN=<tokenvalue>

`*Note:`*

Mandatory parameters are marked in redand placeholder content is marked in <green>.

add

URL:http://<Citrix-ADC-IP-address(NSIP)/nitro/v1/config/vpnvserver_authenticationsamlidppolicy_binding

HTTP Method:PUT

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Content-Type:application/json

Request Payload:

```
{
```

```
"vpnvserver_authenticationsamlidppolicy_binding":{

<b>"name":<String_value>,

</b>"policy":<String_value>,

"priority":<Double_value>,

"secondary":<Boolean_value>,

"groupextraction":<Boolean_value>,

"gotopriorityexpression":<String_value>,

"bindpoint":<String_value>

}}
```

Response:

HTTP Status Code on Success: 201 Created HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error

delete

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/vpnvserver_authenticationsamlidppolicy_binding/name_value<String>

Query-parameters:

args

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/vpnvserver_authenticationsamlidppolicy_binding/name_value<String>?**args=policy:<String_value>,secondary:<Boolean_value>,groupextraction:<Boolean_value>,bindpoint:<**

HTTP Method:DELETE

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue>

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error

get

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/vpnvserver_authenticationsamlidppolicy_binding/name_value<String>

Query-parameters:

filter

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/vpnvserver_authenticationsamlidppolicy_binding/name_value<String>?**filter=property-name1:property-value1,property-name2:property-value2**

Use this query-parameter to get the filtered set of vpnvserver_authenticationsamlidppolicy_binding resources configured on Citrix ADC. Filtering can be done on any of the properties of the resource.

pagination

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/vpnvserver_authenticationsamlidppolicy_binding/name_value<String>?**pagesize=#no;pageno=#no**

Use this query-parameter to get the vpnvserver_authenticationsamlidppolicy_binding resources in chunks.

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "vpnvserver_authenticationsamlidppolicy_binding": [ {

"priority":<Double_value>,

"gotopriorityexpression":<String_value>,

"policy":<String_value>,

"groupextraction":<Boolean_value>,

"name":<String_value>,

"secondary":<Boolean_value>,

"bindpoint":<String_value>

}]}
```

get (all)

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/vpnvserver_authenticationsamlidppolicy_binding

Query-parameters:

bulkbindings

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/vpnvserver_authenticationsamlidppolicy_binding?**bulkbindings=yes**

NITRO allows you to fetch bindings in bulk.

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "vpnvserver_authenticationsamlidppolicy_binding": [ {

"priority":<Double_value>,

"gotopriorityexpression":<String_value>,

"policy":<String_value>,

"groupextraction":<Boolean_value>,

"name":<String_value>,

"secondary":<Boolean_value>,

"bindpoint":<String_value>

}]}
```

count

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/vpnvserver_authenticationsamlidppolicy_binding/name_value<String>?**count=yes**

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{"vpnvserver_authenticationsamlidppolicy_binding": [ { "__count":
"#no"} ] }
```

vpnvserver_authenticationsamlpolicy_binding

Binding object showing the authenticationsamlpolicy that can be bound to vpnvserver.

Properties

(click to see Operations)

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
priority	<Double>	Read-write	Integer specifying the policy's priority. The lower the number, the higher the priority. Policies are evaluated in the order of their priority numbers. Maximum value for default syntax policies is 2147483647 and for classic policies is 64000. Minimum value = 0 Maximum value = 2147483647

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
gotopriorityexpression	<String>	Read-write	Applicable only to advance vpn session policy. Expression or other value specifying the next policy to evaluate if the current policy evaluates to TRUE. Specify one of the following values: * NEXT - Evaluate the policy with the next higher priority number. * END - End policy evaluation. * An expression that evaluates to a number. If you specify an expression, the number to which it evaluates determines the next policy to evaluate, as follows: * If the expression evaluates to a higher numbered priority, the policy with that priority is evaluated next. * If the expression evaluates to the priority of the current policy, the policy with the next higher numbered priority is evaluated next. * If the expression evaluates to a number that is larger than the

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
			largest numbered priority, policy evaluation ends. An UNDEF event is triggered if: * The expression is invalid. * The expression evaluates to a priority number that is numerically lower than the current policy's priority. * The expression evaluates to a priority number that is between the current policy's priority number (say, 30) and the highest priority number (say, 100), but does not match any configured priority number (for example, the expression evaluates to the number 85). This example assumes that the priority number increments by 10 for every successive policy, and therefore a priority number of 85 does not exist in the policy label.
policy	<String>	Read-write	The name of the policy, if any, bound to the VPN virtual server.

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
groupextraction	<Boolean>	Read-write	Binds the authentication policy to a tertiary chain which will be used only for group extraction. The user will not authenticate against this server, and this will only be called if primary and/or secondary authentication has succeeded.
name	<String>	Read-write	Name of the virtual server. Minimum length = 1
secondary	<Boolean>	Read-write	Binds the authentication policy as the secondary policy to use in a two-factor configuration. A user must then authenticate not only via a primary authentication method but also via a secondary authentication method. User groups are aggregated across both. The user name must be exactly the same for both authentication methods, but they can require different passwords.

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
bindpoint	<String>	Read-write	Bind point to which to bind the policy. Applies only to rewrite and cache policies. If you do not set this parameter, the policy is bound to REQ_DEFAULT or RES_DEFAULT, depending on whether the policy rule is a response-time or a request-time expression. Possible values = REQUEST, RESPONSE, ICA_REQUEST, OTHERTCP_REQUEST
acttype	<Double>	Read-only	.
__count	<Double>	Read-only	count parameter

Operations

(click to see Properties)

ADD:| DELETE:| GET| GET (ALL)| COUNT

Some options that you can use for each operations:

- **Getting warnings in response:**NITRO allows you to get warnings in an operation by specifying the "warning" query parameter as "yes". For example, to get warnings while connecting to the Citrix ADC appliance, the URL is as follows:

  http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/login?warning=yes

  If any, the warnings are displayed in the response payload with the HTTP code "209 X-NITRO-WARNING".

- **Authenticated access for individual NITRO operations:**NITRO allows you to logon to the Citrix ADC appliance to perform individual operations. You can use this option instead of creating a NITRO session (using the login object) and then using that session to perform all operations,

  To do this, you must specify the username and password in the request header of the NITRO request as follows:

  X-NITRO-USER:<username>

  X-NITRO-PASS:<password>

  Note: In such cases, make sure that the request header DOES not include the following:

  Cookie:NITRO_AUTH_TOKEN=<tokenvalue>

`*Note:`*

Mandatory parameters are marked in redand placeholder content is marked in <green>.

add

URL:http://<Citrix-ADC-IP-address(NSIP)/nitro/v1/config/vpnvserver_authenticationsamlpolicy_binding

HTTP Method:PUT

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Content-Type:application/json

Request Payload:

```
{

"vpnvserver_authenticationsamlpolicy_binding":{

<b>"name":<String_value>,

</b>"policy":<String_value>,

"priority":<Double_value>,
```

```
"secondary":<Boolean_value>,

"groupextraction":<Boolean_value>,

"gotopriorityexpression":<String_value>,

"bindpoint":<String_value>

}}
```

Response:

HTTP Status Code on Success: 201 Created HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error

delete

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/vpnvserver_authenticationsamlpolicy_binding/name_value<String>

Query-parameters:

args

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/vpnvserver_authenticationsamlpolicy_binding/name_value<String>?**args=policy:<String_value>,secondary:<Boolean_value>,groupextraction:<Boolean_value>,bindpoint:<**

HTTP Method:DELETE

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue>

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error

get

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/vpnvserver_authenticationsamlpolicy_binding/name_value<String>

Query-parameters:

filter

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/vpnvserver_authenticationsamlpolicy_binding/name_value<String>?**filter=property-name1:property-value1,property-name2:property-value2**

Use this query-parameter to get the filtered set of vpnvserver_authenticationsamlpolicy_binding resources configured on Citrix ADC. Filtering can be done on any of the properties of the resource.

pagination

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/vpnvserver_authenticationsamlpolicy_binding/name_value<String>?**pagesize=#no;pageno=#no**

Use this query-parameter to get the vpnvserver_authenticationsamlpolicy_binding resources in chunks.

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "vpnvserver_authenticationsamlpolicy_binding": [ {

"priority":<Double_value>,
```

```
"gotopriorityexpression":<String_value>,

"policy":<String_value>,

"groupextraction":<Boolean_value>,

"name":<String_value>,

"secondary":<Boolean_value>,

"bindpoint":<String_value>,

"acttype":<Double_value>

}]}
```

get (all)

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/vpnvserver_authenticationsamlpolicy_binding

Query-parameters:

bulkbindings

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/vpnvserver_authenticationsamlpolicy_binding?**bulkbindings=yes**

NITRO allows you to fetch bindings in bulk.

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "vpnvserver_authenticationsamlpolicy_binding": [ {

"priority":<Double_value>,

"gotopriorityexpression":<String_value>,

"policy":<String_value>,

"groupextraction":<Boolean_value>,

"name":<String_value>,

"secondary":<Boolean_value>,

"bindpoint":<String_value>,

"acttype":<Double_value>

}]}
```

count

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/vpnvserver_authenticationsamlpolicy_binding/name_value<String>?**count=yes**

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{"vpnvserver_authenticationsamlpolicy_binding": [ { "__count": "#no"} ] }
```

vpnvserver_authenticationtacacspolicy_binding

Binding object showing the authenticationtacacspolicy that can be bound to vpnvserver.

Properties

(click to see Operations)

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
priority	<Double>	Read-write	Integer specifying the policy's priority. The lower the number, the higher the priority. Policies are evaluated in the order of their priority numbers. Maximum value for default syntax policies is 2147483647 and for classic policies is 64000. Minimum value = 0 Maximum value = 2147483647
gotopriorityexpression	<String>	Read-write	Applicable only to advance vpn session policy. Expression or other value specifying the next policy to evaluate if the current policy evaluates to TRUE.

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
			Specify one of the following values: * NEXT - Evaluate the policy with the next higher priority number. * END - End policy evaluation. * An expression that evaluates to a number. If you specify an expression, the number to which it evaluates determines the next policy to evaluate, as follows: * If the expression evaluates to a higher numbered priority, the policy with that priority is evaluated next. * If the expression evaluates to the priority of the current policy, the policy with the next higher numbered priority is evaluated next. * If the expression evaluates to a number that is larger than the largest numbered priority, policy evaluation ends. An UNDEF event is triggered if: * The expression is invalid. * The expression evaluates to a

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
			priority number that is numerically lower than the current policy's priority. * The expression evaluates to a priority number that is between the current policy's priority number (say, 30) and the highest priority number (say, 100), but does not match any configured priority number (for example, the expression evaluates to the number 85). This example assumes that the priority number increments by 10 for every successive policy, and therefore a priority number of 85 does not exist in the policy label.
policy	<String>	Read-write	The name of the policy, if any, bound to the VPN virtual server.
groupextraction	<Boolean>	Read-write	Binds the authentication policy to a tertiary chain which will be used only for group extraction. The user will not authenticate against this server,

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
			and this will only be called if primary and/or secondary authentication has succeeded.
name	<String>	Read-write	Name of the virtual server. Minimum length = 1
secondary	<Boolean>	Read-write	Binds the authentication policy as the secondary policy to use in a two-factor configuration. A user must then authenticate not only via a primary authentication method but also via a secondary authentication method. User groups are aggregated across both. The user name must be exactly the same for both authentication methods, but they can require different passwords.
bindpoint	<String>	Read-write	Bind point to which to bind the policy. Applies only to rewrite and cache policies. If you do not set this parameter, the policy is bound to REQ_DEFAULT or

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
			RES_DEFAULT, depending on whether the policy rule is a response-time or a request-time expression. Possible values = REQUEST, RESPONSE, ICA_REQUEST, OTHERTCP_REQUEST
acttype	<Double>	Read-only	.
__count	<Double>	Read-only	count parameter

Operations

(click to see Properties)

ADD:| DELETE:| GET| GET (ALL)| COUNT

Some options that you can use for each operations:

- **Getting warnings in response:**NITRO allows you to get warnings in an operation by specifying the "warning" query parameter as "yes". For example, to get warnings while connecting to the Citrix ADC appliance, the URL is as follows:

  http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/login?warning=yes

  If any, the warnings are displayed in the response payload with the HTTP code "209 X-NITRO-WARNING".

- **Authenticated access for individual NITRO operations:**NITRO allows you to logon to the Citrix ADC appliance to perform individual operations. You can use this option instead of creating a NITRO session (using the login object) and then using that session to perform all operations,

  To do this, you must specify the username and password in the request header of the NITRO request as follows:

X-NITRO-USER:<username>

X-NITRO-PASS:<password>

Note: In such cases, make sure that the request header DOES not include the following:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue>

`*Note:`*

Mandatory parameters are marked in redand placeholder content is marked in <green>.

add

URL:http://<Citrix-ADC-IP-address(NSIP)/nitro/v1/config/vpnvserver_authenticationtacacspolicy_binding

HTTP Method:PUT

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Content-Type:application/json

Request Payload:

```
{

"vpnvserver_authenticationtacacspolicy_binding":{

<b>"name":<String_value>,

</b>"policy":<String_value>,

"priority":<Double_value>,

"secondary":<Boolean_value>,

"groupextraction":<Boolean_value>,

"gotopriorityexpression":<String_value>,

"bindpoint":<String_value>
```

```
}}
```

Response:

HTTP Status Code on Success: 201 Created HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error

delete

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/vpnvserver_authenticationtacacspolicy_binding/name_value<String>

Query-parameters:

args

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/vpnvserver_authenticationtacacspolicy_binding/name_value<String>?**args=policy:<String_value>,secondary:<Boolean_value>,groupextraction:<Boolean_value>,bindpoint:<**

HTTP Method:DELETE

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue>

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error

get

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/vpnvserver_authenticationtacacspolicy_binding/name_value<String>

Query-parameters:

filter

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/vpnvserver_authenticationtacacspolicy_binding/name_value<String>?**filter=property-name1:property-value1,property-name2:property-value2**

Use this query-parameter to get the filtered set of vpnvserver_authenticationtacacspolicy_binding resources configured on Citrix ADC. Filtering can be done on any of the properties of the resource.

pagination

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/vpnvserver_authenticationtacacspolicy_binding/name_value<String>?**pagesize=#no;pageno=#no**

Use this query-parameter to get the vpnvserver_authenticationtacacspolicy_binding resources in chunks.

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "vpnvserver_authenticationtacacspolicy_binding": [ {

"priority":<Double_value>,

"gotopriorityexpression":<String_value>,

"policy":<String_value>,

"groupextraction":<Boolean_value>,

"name":<String_value>,
```

```
"secondary":<Boolean_value>,

"bindpoint":<String_value>,

"acttype":<Double_value>

}]}
```

get (all)

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/vpnvserver_authenticationtacacspolicy_binding

Query-parameters:

bulkbindings

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/vpnvserver_authenticationtacacspolicy_binding?**bulkbindings=yes**

NITRO allows you to fetch bindings in bulk.

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "vpnvserver_authenticationtacacspolicy_binding": [ {

"priority":<Double_value>,
```

```
"gotopriorityexpression":<String_value>,

"policy":<String_value>,

"groupextraction":<Boolean_value>,

"name":<String_value>,

"secondary":<Boolean_value>,

"bindpoint":<String_value>,

"acttype":<Double_value>

}]}
```

count

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/vpnvserver_authenticationtacacspolicy_binding/name_value<String>?**count=yes**

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{"vpnvserver_authenticationtacacspolicy_binding": [ { "__count":
"#no"} ] }
```

vpnvserver_authenticationwebauthpolicy_binding

Binding object showing the authenticationwebauthpolicy that can be bound to vpnvserver.

Properties

(click to see Operations)

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
priority	<Double>	Read-write	Integer specifying the policy's priority. The lower the number, the higher the priority. Policies are evaluated in the order of their priority numbers. Maximum value for default syntax policies is 2147483647 and for classic policies is 64000. Minimum value = 0 Maximum value = 2147483647
gotopriorityexpression	<String>	Read-write	Applicable only to advance vpn session policy. Expression or other value specifying the next policy to evaluate if the current policy evaluates to TRUE. Specify one of the following values: * NEXT - Evaluate the policy with the next higher priority

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
			number. * END - End policy evaluation. * An expression that evaluates to a number. If you specify an expression, the number to which it evaluates determines the next policy to evaluate, as follows: * If the expression evaluates to a higher numbered priority, the policy with that priority is evaluated next. * If the expression evaluates to the priority of the current policy, the policy with the next higher numbered priority is evaluated next. * If the expression evaluates to a number that is larger than the largest numbered priority, policy evaluation ends. An UNDEF event is triggered if: * The expression is invalid. * The expression evaluates to a priority number that is numerically lower than the current policy's priority. * The expression evaluates

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
			to a priority number that is between the current policy's priority number (say, 30) and the highest priority number (say, 100), but does not match any configured priority number (for example, the expression evaluates to the number 85). This example assumes that the priority number increments by 10 for every successive policy, and therefore a priority number of 85 does not exist in the policy label.
policy	<String>	Read-write	The name of the policy, if any, bound to the VPN virtual server.
groupextraction	<Boolean>	Read-write	Binds the authentication policy to a tertiary chain which will be used only for group extraction. The user will not authenticate against this server, and this will only be called if primary and/or secondary authentication has succeeded.

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
name	<String>	Read-write	Name of the virtual server. Minimum length = 1
secondary	<Boolean>	Read-write	Binds the authentication policy as the secondary policy to use in a two-factor configuration. A user must then authenticate not only via a primary authentication method but also via a secondary authentication method. User groups are aggregated across both. The user name must be exactly the same for both authentication methods, but they can require different passwords.
bindpoint	<String>	Read-write	Bind point to which to bind the policy. Applies only to rewrite and cache policies. If you do not set this parameter, the policy is bound to REQ_DEFAULT or RES_DEFAULT, depending on whether the policy rule is a response-

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
			time or a request-time expression. Possible values = REQUEST, RESPONSE, ICA_REQUEST, OTHERTCP_REQUEST
acttype	<Double>	Read-only	.
__count	<Double>	Read-only	count parameter

Operations

(click to see Properties)

ADD:| DELETE:| GET| GET (ALL)| COUNT

Some options that you can use for each operations:

- **Getting warnings in response:**NITRO allows you to get warnings in an operation by specifying the "warning" query parameter as "yes". For example, to get warnings while connecting to the Citrix ADC appliance, the URL is as follows:

  http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/login?warning=yes

  If any, the warnings are displayed in the response payload with the HTTP code "209 X-NITRO-WARNING".
- **Authenticated access for individual NITRO operations:**NITRO allows you to logon to the Citrix ADC appliance to perform individual operations. You can use this option instead of creating a NITRO session (using the login object) and then using that session to perform all operations,

  To do this, you must specify the username and password in the request header of the NITRO request as follows:

  X-NITRO-USER:<username>

  X-NITRO-PASS:<password>

Note: In such cases, make sure that the request header DOES not include the following:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue>

`*Note:`*

Mandatory parameters are marked in redand placeholder content is marked in <green>.

add

URL:http://<Citrix-ADC-IP-address(NSIP)/nitro/v1/config/vpnvserver_authenticationwebauthpolicy_binding

HTTP Method:PUT

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Content-Type:application/json

Request Payload:

```
{

"vpnvserver_authenticationwebauthpolicy_binding":{

<b>"name":<String_value>,

</b>"policy":<String_value>,

"priority":<Double_value>,

"secondary":<Boolean_value>,

"groupextraction":<Boolean_value>,

"gotopriorityexpression":<String_value>,

"bindpoint":<String_value>

}}
```

Response:

HTTP Status Code on Success: 201 Created HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error

delete

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/vpnvserver_authenticationwebauthpolicy_binding/name_value<String>

Query-parameters:

args

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/vpnvserver_authenticationwebauthpolicy_binding/name_value<String>?**args=policy:<String_value>,secondary:<Boolean_value>,groupextraction:<Boolean_value>,bindpoint:<**

HTTP Method:DELETE

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue>

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error

get

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/vpnvserver_authenticationwebauthpolicy_binding/name_value<String>

Query-parameters:

filter

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/vpnvserver_authenticationwebauthpolicy_binding/name_value<String>?**filter=property-name1:property-value1,property-name2:property-value2**

Use this query-parameter to get the filtered set of vpnvserver_authenticationwebauthpolicy_binding resources configured on Citrix ADC. Filtering can be done on any of the properties of the resource.

pagination

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/vpnvserver_authenticationwebauthpolicy_binding/name_value<String>?**pagesize=#no;pageno=#no**

Use this query-parameter to get the vpnvserver_authenticationwebauthpolicy_binding resources in chunks.

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "vpnvserver_authenticationwebauthpolicy_binding": [ {

"priority":<Double_value>,

"gotopriorityexpression":<String_value>,

"policy":<String_value>,

"groupextraction":<Boolean_value>,

"name":<String_value>,

"secondary":<Boolean_value>,

"bindpoint":<String_value>,

"acttype":<Double_value>
```

```
}]}
```

get (all)

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/vpnvserver_authenticationwebauthpolicy_binding

Query-parameters:

bulkbindings

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/vpnvserver_authenticationwebauthpolicy_binding?**bulkbindings=yes**

NITRO allows you to fetch bindings in bulk.

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "vpnvserver_authenticationwebauthpolicy_binding": [ {

"priority":<Double_value>,

"gotopriorityexpression":<String_value>,

"policy":<String_value>,

"groupextraction":<Boolean_value>,
```

```
"name":<String_value>,

"secondary":<Boolean_value>,

"bindpoint":<String_value>,

"acttype":<Double_value>

}]}
```

count

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/vpnvserver_authenticationwebauthpolicy_binding/name_value<String>?**count=yes**

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{"vpnvserver_authenticationwebauthpolicy_binding": [ { "__count":
"#no"} ] }
```

vpnvserver_binding

Binding object which returns the resources bound to vpnvserver.

Properties

(click to see Operations)

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
name	<String>	Read-write	Name of the Citrix Gateway virtual server for which to show detailed information. Minimum length = 1
vpnvserver_intranetip6_binding	<vpnvserver_intranetip6_binding[]>	Read-only	intranetip6 that can be bound to vpnvserver.
vpnvserver_authenticationradiuspolicy_binding	<vpnvserver_authenticationradiuspolicy_binding[]>	Read-only	authenticationradiuspolicy that can be bound to vpnvserver.
vpnvserver_vpnsessionpolicy_binding	<vpnvserver_vpnsessionpolicy_binding[]>	Read-only	vpnsessionpolicy that can be bound to vpnvserver.
vpnvserver_vpnurlpolicy_binding	<vpnvserver_vpnurlpolicy_binding[]>	Read-only	vpnurlpolicy that can be bound to vpnvserver.
vpnvserver_vpnnexthopserver_binding	<vpnvserver_vpnnexthopserver_binding[]>	Read-only	vpnnexthopserver that can be bound to vpnvserver.
vpnvserver_authenticationldappolicy_binding	<vpnvserver_authenticationldappolicy_binding[]>	Read-only	authenticationldappolicy that can be bound to vpnvserver.
vpnvserver_authenticationsamlidppolicy_binding	<vpnvserver_authenticationsamlidppolicy_binding[]>	Read-only	authenticationsamlidppolicy that can be bound to vpnvserver.
vpnvserver_authenticationwebauthpolicy_binding	<vpnvserver_authenticationwebauthpolicy_binding[]>	Read-only	authenticationwebauthpolicy that can be bound to vpnvserver.

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
vpnvserver_sharefileserver_binding	<vpnvserver_sharefileserver_binding[]>	Read-only	sharefileserver that can be bound to vpnvserver.
vpnvserver_authenticationlocalpolicy_binding	<vpnvserver_authenticationlocalpolicy_binding[]>	Read-only	authenticationlocalpolicy that can be bound to vpnvserver.
vpnvserver_vpnintranetapplication_binding	<vpnvserver_vpnintranetapplication_binding[]>	Read-only	vpnintranetapplication that can be bound to vpnvserver.
vpnvserver_appcontroller_binding	<vpnvserver_appcontroller_binding[]>	Read-only	appcontroller that can be bound to vpnvserver.
vpnvserver_rewritepolicy_binding	<vpnvserver_rewritepolicy_binding[]>	Read-only	rewritepolicy that can be bound to vpnvserver.
vpnvserver_vpnportaltheme_binding	<vpnvserver_vpnportaltheme_binding[]>	Read-only	vpnportaltheme that can be bound to vpnvserver.
vpnvserver_cspolicy_binding	<vpnvserver_cspolicy_binding[]>	Read-only	cspolicy that can be bound to vpnvserver.
vpnvserver_auditsyslogpolicy_binding	<vpnvserver_auditsyslogpolicy_binding[]>	Read-only	auditsyslogpolicy that can be bound to vpnvserver.
vpnvserver_authenticationcertpolicy_binding	<vpnvserver_authenticationcertpolicy_binding[]>	Read-only	authenticationcertpolicy that can be bound to vpnvserver.
vpnvserver_vpnepaprofile_binding	<vpnvserver_vpnepaprofile_binding[]>	Read-only	vpnepaprofile that can be bound to vpnvserver.
vpnvserver_vpnurl_binding	<vpnvserver_vpnurl_binding[]>	Read-only	vpnurl that can be bound to vpnvserver.

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
vpnvserver_authenticationoauthidppolicy_binding	<vpnvserver_authenticationoauthidppolicy_binding[]>	Read-only	authenticationoauthidppolicy that can be bound to vpnvserver.
vpnvserver_auditnslogpolicy_binding	<vpnvserver_auditnslogpolicy_binding[]>	Read-only	auditnslogpolicy that can be bound to vpnvserver.
vpnvserver_authenticationpolicy_binding	<vpnvserver_authenticationpolicy_binding[]>	Read-only	authenticationpolicy that can be bound to vpnvserver.
vpnvserver_vpneula_binding	<vpnvserver_vpneula_binding[]>	Read-only	vpneula that can be bound to vpnvserver.
vpnvserver_icapolicy_binding	<vpnvserver_icapolicy_binding[]>	Read-only	icapolicy that can be bound to vpnvserver.
vpnvserver_vpnclientlessaccesspolicy_binding	<vpnvserver_vpnclientlessaccesspolicy_binding[]>	Read-only	vpnclientlessaccesspolicy that can be bound to vpnvserver.
vpnvserver_intranetip_binding	<vpnvserver_intranetip_binding[]>	Read-only	intranetip that can be bound to vpnvserver.
vpnvserver_appflowpolicy_binding	<vpnvserver_appflowpolicy_binding[]>	Read-only	appflowpolicy that can be bound to vpnvserver.
vpnvserver_responderpolicy_binding	<vpnvserver_responderpolicy_binding[]>	Read-only	responderpolicy that can be bound to vpnvserver.
vpnvserver_vpntrafficpolicy_binding	<vpnvserver_vpntrafficpolicy_binding[]>	Read-only	vpntrafficpolicy that can be bound to vpnvserver.
vpnvserver_authenticationnegotiatepolicy_binding	<vpnvserver_authenticationnegotiatepolicy_binding[]>	Read-only	authenticationnegotiatepolicy that can be bound to vpnvserver.

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
vpnvserver_feopolicy_binding	<vpnvserver_feopolicy_binding[]>	Read-only	feopolicy that can be bound to vpnvserver.
vpnvserver_aaapreauthenticationpolicy_binding	<vpnvserver_aaapreauthenticationpolicy_binding[]>	Read-only	aaapreauthentication policy that can be bound to vpnvserver.
vpnvserver_cachepolicy_binding	<vpnvserver_cachepolicy_binding[]>	Read-only	cachepolicy that can be bound to vpnvserver.
vpnvserver_authenticationtacacspolicy_binding	<vpnvserver_authenticationtacacspolicy_binding[]>	Read-only	authenticationtacacs policy that can be bound to vpnvserver.
vpnvserver_authenticationsamlpolicy_binding	<vpnvserver_authenticationsamlpolicy_binding[]>	Read-only	authenticationsamlpolicy that can be bound to vpnvserver.
vpnvserver_authenticationdfapolicy_binding	<vpnvserver_authenticationdfapolicy_binding[]>	Read-only	authenticationdfapolicy that can be bound to vpnvserver.
vpnvserver_staserver_binding	<vpnvserver_staserver_binding[]>	Read-only	staserver that can be bound to vpnvserver.
vpnvserver_authenticationloginschemapolicy_binding	<vpnvserver_authenticationloginschemapolicy_binding[]>	Read-only	authenticationloginschemapolicy that can be bound to vpnvserver.

Operations

(click to see Properties)

GET| GET (ALL)

Some options that you can use for each operations:

- **Getting warnings in response:**NITRO allows you to get warnings in an operation by specifying the "warning" query parameter as "yes". For example, to get warnings while connecting to the Citrix ADC appliance, the URL is as follows:

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/login?warning=yes

If any, the warnings are displayed in the response payload with the HTTP code "209 X-NITRO-WARNING".

- **Authenticated access for individual NITRO operations:**NITRO allows you to logon to the Citrix ADC appliance to perform individual operations. You can use this option instead of creating a NITRO session (using the login object) and then using that session to perform all operations,

  To do this, you must specify the username and password in the request header of the NITRO request as follows:

  X-NITRO-USER:<username>

  X-NITRO-PASS:<password>

  Note: In such cases, make sure that the request header DOES not include the following:

  Cookie:NITRO_AUTH_TOKEN=<tokenvalue>

`*Note:`*

Mandatory parameters are marked in redand placeholder content is marked in <green>.

get

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/vpnvserver_binding/name_value<String>

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "vpnvserver_binding": [ {

"name":<String_value>,

"vpnvserver_vpnurlpolicy_binding":<vpnvserver_vpnurlpolicy_binding[]_value
>,

"vpnvserver_authenticationoauthidppolicy_binding":<vpnvserver_authenticati
onoauthidppolicy_binding[]_value>,

"vpnvserver_authenticationloginschemapolicy_binding":<vpnvserver_authentic
ationloginschemapolicy_binding[]_value>,

"vpnvserver_authenticationsamlpolicy_binding":<vpnvserver_authenticationsa
mlpolicy_binding[]_value>,

"vpnvserver_cspolicy_binding":<vpnvserver_cspolicy_binding[]_value>,

"vpnvserver_icapolicy_binding":<vpnvserver_icapolicy_binding[]_value>,

"vpnvserver_vpnportaltheme_binding":<vpnvserver_vpnportaltheme_binding[]_v
alue>,

"vpnvserver_appflowpolicy_binding":<vpnvserver_appflowpolicy_binding[]_val
ue>,

"vpnvserver_intranetip_binding":<vpnvserver_intranetip_binding[]_value>,

"vpnvserver_vpntrafficpolicy_binding":<vpnvserver_vpntrafficpolicy_binding
[]_value>,

"vpnvserver_vpnepaprofile_binding":<vpnvserver_vpnepaprofile_binding[]_val
ue>,

"vpnvserver_authenticationtacacspolicy_binding":<vpnvserver_authentication
tacacspolicy_binding[]_value>,

"vpnvserver_vpnintranetapplication_binding":<vpnvserver_vpnintranetapplica
tion_binding[]_value>,

"vpnvserver_authenticationradiuspolicy_binding":<vpnvserver_authentication
```

```
radiuspolicy_binding[]_value>,

"vpnvserver_authenticationpolicy_binding":<vpnvserver_authenticationpolicy
_binding[]_value>,

"vpnvserver_aaapreauthenticationpolicy_binding":<vpnvserver_aaapreauthenti
cationpolicy_binding[]_value>,

"vpnvserver_feopolicy_binding":<vpnvserver_feopolicy_binding[]_value>,

"vpnvserver_authenticationlocalpolicy_binding":<vpnvserver_authenticationl
ocalpolicy_binding[]_value>,

"vpnvserver_intranetip6_binding":<vpnvserver_intranetip6_binding[]_value>,

"vpnvserver_vpnsessionpolicy_binding":<vpnvserver_vpnsessionpolicy_binding
[]_value>,

"vpnvserver_staserver_binding":<vpnvserver_staserver_binding[]_value>,

"vpnvserver_authenticationcertpolicy_binding":<vpnvserver_authenticationce
rtpolicy_binding[]_value>,

"vpnvserver_authenticationsamlidppolicy_binding":<vpnvserver_authenticatio
nsamlidppolicy_binding[]_value>,

"vpnvserver_vpnnexthopserver_binding":<vpnvserver_vpnnexthopserver_binding
[]_value>,

"vpnvserver_auditnslogpolicy_binding":<vpnvserver_auditnslogpolicy_binding
[]_value>,

"vpnvserver_auditsyslogpolicy_binding":<vpnvserver_auditsyslogpolicy_bindi
ng[]_value>,

"vpnvserver_sharefileserver_binding":<vpnvserver_sharefileserver_binding[]
_value>,

"vpnvserver_responderpolicy_binding":<vpnvserver_responderpolicy_binding[]
_value>,

"vpnvserver_vpnurl_binding":<vpnvserver_vpnurl_binding[]_value>,

"vpnvserver_cachepolicy_binding":<vpnvserver_cachepolicy_binding[]_value>,
```

```
"vpnvserver_authenticationldappolicy_binding":<vpnvserver_authenticationld
appolicy_binding[]_value>,

"vpnvserver_authenticationwebauthpolicy_binding":<vpnvserver_authenticatio
nwebauthpolicy_binding[]_value>,

"vpnvserver_rewritepolicy_binding":<vpnvserver_rewritepolicy_binding[]_val
ue>,

"vpnvserver_authenticationnegotiatepolicy_binding":<vpnvserver_authenticat
ionnegotiatepolicy_binding[]_value>,

"vpnvserver_appcontroller_binding":<vpnvserver_appcontroller_binding[]_val
ue>,

"vpnvserver_vpnclientlessaccesspolicy_binding":<vpnvserver_vpnclientlessac
cesspolicy_binding[]_value>,

"vpnvserver_authenticationdfapolicy_binding":<vpnvserver_authenticationdfa
policy_binding[]_value>,

"vpnvserver_vpneula_binding":<vpnvserver_vpneula_binding[]_value>

}]}
```

get (all)

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/vpnvserver_binding

Query-parameters:

bulkbindings

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/vpnvserver_binding?**bulkbindings=yes**

NITRO allows you to fetch bindings in bulk.

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "vpnvserver_binding": [ {

"name":<String_value>,

"vpnvserver_vpnurlpolicy_binding":<vpnvserver_vpnurlpolicy_binding[]_value
>,

"vpnvserver_authenticationoauthidppolicy_binding":<vpnvserver_authenticati
onoauthidppolicy_binding[]_value>,

"vpnvserver_authenticationloginschemapolicy_binding":<vpnvserver_authentic
ationloginschemapolicy_binding[]_value>,

"vpnvserver_authenticationsamlpolicy_binding":<vpnvserver_authenticationsa
mlpolicy_binding[]_value>,

"vpnvserver_cspolicy_binding":<vpnvserver_cspolicy_binding[]_value>,

"vpnvserver_icapolicy_binding":<vpnvserver_icapolicy_binding[]_value>,

"vpnvserver_vpnportaltheme_binding":<vpnvserver_vpnportaltheme_binding[]_v
alue>,

"vpnvserver_appflowpolicy_binding":<vpnvserver_appflowpolicy_binding[]_val
ue>,

"vpnvserver_intranetip_binding":<vpnvserver_intranetip_binding[]_value>,

"vpnvserver_vpntrafficpolicy_binding":<vpnvserver_vpntrafficpolicy_binding
[]_value>,
```

```
"vpnvserver_vpnepaprofile_binding":<vpnvserver_vpnepaprofile_binding[]_val
ue>,

"vpnvserver_authenticationtacacspolicy_binding":<vpnvserver_authentication
tacacspolicy_binding[]_value>,

"vpnvserver_vpnintranetapplication_binding":<vpnvserver_vpnintranetapplica
tion_binding[]_value>,

"vpnvserver_authenticationradiuspolicy_binding":<vpnvserver_authentication
radiuspolicy_binding[]_value>,

"vpnvserver_authenticationpolicy_binding":<vpnvserver_authenticationpolicy
_binding[]_value>,

"vpnvserver_aaapreauthenticationpolicy_binding":<vpnvserver_aaapreauthenti
cationpolicy_binding[]_value>,

"vpnvserver_feopolicy_binding":<vpnvserver_feopolicy_binding[]_value>,

"vpnvserver_authenticationlocalpolicy_binding":<vpnvserver_authenticationl
ocalpolicy_binding[]_value>,

"vpnvserver_intranetip6_binding":<vpnvserver_intranetip6_binding[]_value>,

"vpnvserver_vpnsessionpolicy_binding":<vpnvserver_vpnsessionpolicy_binding
[]_value>,

"vpnvserver_staserver_binding":<vpnvserver_staserver_binding[]_value>,

"vpnvserver_authenticationcertpolicy_binding":<vpnvserver_authenticationce
rtpolicy_binding[]_value>,

"vpnvserver_authenticationsamlidppolicy_binding":<vpnvserver_authenticatio
nsamlidppolicy_binding[]_value>,

"vpnvserver_vpnnexthopserver_binding":<vpnvserver_vpnnexthopserver_binding
[]_value>,

"vpnvserver_auditnslogpolicy_binding":<vpnvserver_auditnslogpolicy_binding
[]_value>,

"vpnvserver_auditsyslogpolicy_binding":<vpnvserver_auditsyslogpolicy_bindi
```

```
ng[]_value>,

"vpnvserver_sharefileserver_binding":<vpnvserver_sharefileserver_binding[]
_value>,

"vpnvserver_responderpolicy_binding":<vpnvserver_responderpolicy_binding[]
_value>,

"vpnvserver_vpnurl_binding":<vpnvserver_vpnurl_binding[]_value>,

"vpnvserver_cachepolicy_binding":<vpnvserver_cachepolicy_binding[]_value>,

"vpnvserver_authenticationldappolicy_binding":<vpnvserver_authenticationld
appolicy_binding[]_value>,

"vpnvserver_authenticationwebauthpolicy_binding":<vpnvserver_authenticatio
nwebauthpolicy_binding[]_value>,

"vpnvserver_rewritepolicy_binding":<vpnvserver_rewritepolicy_binding[]_val
ue>,

"vpnvserver_authenticationnegotiatepolicy_binding":<vpnvserver_authenticat
ionnegotiatepolicy_binding[]_value>,

"vpnvserver_appcontroller_binding":<vpnvserver_appcontroller_binding[]_val
ue>,

"vpnvserver_vpnclientlessaccesspolicy_binding":<vpnvserver_vpnclientlessac
cesspolicy_binding[]_value>,

"vpnvserver_authenticationdfapolicy_binding":<vpnvserver_authenticationdfa
policy_binding[]_value>,

"vpnvserver_vpneula_binding":<vpnvserver_vpneula_binding[]_value>

}]}
```

vpnvserver_cachepolicy_binding

Binding object showing the cachepolicy that can be bound to vpnvserver.

Properties

(click to see Operations)

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
priority	<Double>	Read-write	Integer specifying the policy's priority. The lower the number, the higher the priority. Policies are evaluated in the order of their priority numbers. Maximum value for default syntax policies is 2147483647 and for classic policies is 64000. Minimum value = 0 Maximum value = 2147483647
gotopriorityexpression	<String>	Read-write	Next priority expression.
policy	<String>	Read-write	The name of the policy, if any, bound to the VPN virtual server.
groupextraction	<Boolean>	Read-write	Binds the authentication policy to a tertiary chain which will be used only for group extraction. The user will not authenticate against this server, and this will only be called if primary and/or secondary

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
			authentication has succeeded.
name	<String>	Read-write	Name of the virtual server. Minimum length = 1
secondary	<Boolean>	Read-write	Binds the authentication policy as the secondary policy to use in a two-factor configuration. A user must then authenticate not only via a primary authentication method but also via a secondary authentication method. User groups are aggregated across both. The user name must be exactly the same for both authentication methods, but they can require different passwords.
bindpoint	<String>	Read-write	Bind point to which to bind the policy. Applies only to rewrite and cache policies. If you do not set this parameter, the policy is bound to REQ_DEFAULT or RES_DEFAULT, depending on whether the policy

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
			rule is a response-time or a request-time expression. Possible values = REQUEST, RESPONSE, ICA_REQUEST, OTHERTCP_REQUEST
__count	<Double>	Read-write	count parameter

Operations

(click to see Properties)

ADD:| DELETE:| GET| GET (ALL)| COUNT

Some options that you can use for each operations:

- **Getting warnings in response:**NITRO allows you to get warnings in an operation by specifying the "warning" query parameter as "yes". For example, to get warnings while connecting to the Citrix ADC appliance, the URL is as follows:

  http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/login?warning=yes

  If any, the warnings are displayed in the response payload with the HTTP code "209 X-NITRO-WARNING".
- **Authenticated access for individual NITRO operations:**NITRO allows you to logon to the Citrix ADC appliance to perform individual operations. You can use this option instead of creating a NITRO session (using the login object) and then using that session to perform all operations,

  To do this, you must specify the username and password in the request header of the NITRO request as follows:

  X-NITRO-USER:<username>

  X-NITRO-PASS:<password>

  Note: In such cases, make sure that the request header DOES not include the following:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue>

`*Note:`*

Mandatory parameters are marked in redand placeholder content is marked in <green>.

add

URL:http://<Citrix-ADC-IP-address(NSIP)/nitro/v1/config/vpnvserver_cachepolicy_binding

HTTP Method:PUT

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Content-Type:application/json

Request Payload:

```
{

"vpnvserver_cachepolicy_binding":{

<b>"name":<String_value>,

</b>"policy":<String_value>,

"priority":<Double_value>,

"secondary":<Boolean_value>,

"groupextraction":<Boolean_value>,

"gotopriorityexpression":<String_value>,

"bindpoint":<String_value>

}}
```

Response:

HTTP Status Code on Success: 201 Created HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error

delete

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/vpnvserver_cachepolicy_binding/name_value<String>

Query-parameters:

args

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/vpnvserver_cachepolicy_binding/name_value<String>?**args=policy:<String_value>,secondary:<Boolean_value>,groupextraction:<Boolean_value>,bindpoint:<**

HTTP Method:DELETE

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue>

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error

get

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/vpnvserver_cachepolicy_binding/name_value<String>

Query-parameters:

filter

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/vpnvserver_cachepolicy_binding/name_value<String>?**filter=property-name1:property-value1,property-name2:property-value2**

Use this query-parameter to get the filtered set of vpnvserver_cachepolicy_binding resources configured on Citrix ADC. Filtering can be done on any of the properties of the resource.

pagination

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/vpnvserver_cachepolicy_binding/name_value<String>?**pagesize=#no;pageno=#no**

Use this query-parameter to get the vpnvserver_cachepolicy_binding resources in chunks.

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "vpnvserver_cachepolicy_binding": [ {

"priority":<Double_value>,

"gotopriorityexpression":<String_value>,

"policy":<String_value>,

"groupextraction":<Boolean_value>,

"name":<String_value>,

"secondary":<Boolean_value>,

"bindpoint":<String_value>

}]}
```

get (all)

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/vpnvserver_cachepolicy_binding

Query-parameters:

bulkbindings

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/vpnvserver_cachepolicy_binding?**bulkbindings=yes**

NITRO allows you to fetch bindings in bulk.

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "vpnvserver_cachepolicy_binding": [ {

"priority":<Double_value>,

"gotopriorityexpression":<String_value>,

"policy":<String_value>,

"groupextraction":<Boolean_value>,

"name":<String_value>,

"secondary":<Boolean_value>,
```

```
"bindpoint":<String_value>

}]}
```

count

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/vpnvserver_cachepolicy_binding/name_value<String>?**count=yes**

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{"vpnvserver_cachepolicy_binding": [ { "__count": "#no"} ] }
```

vpnvserver_cspolicy_binding

Binding object showing the cspolicy that can be bound to vpnvserver.

Properties

(click to see Operations)

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
priority	<Double>	Read-write	Integer specifying the policy's priority.

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
			The lower the number, the higher the priority. Policies are evaluated in the order of their priority numbers. Maximum value for default syntax policies is 2147483647 and for classic policies is 64000. Minimum value = 0 Maximum value = 2147483647
gotopriorityexpression	<String>	Read-write	Next priority expression.
policy	<String>	Read-write	The name of the policy, if any, bound to the VPN virtual server.
groupextraction	<Boolean>	Read-write	Binds the authentication policy to a tertiary chain which will be used only for group extraction. The user will not authenticate against this server, and this will only be called if primary and/or secondary authentication has succeeded.
name	<String>	Read-write	Name of the virtual server. Minimum length = 1

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
secondary	<Boolean>	Read-write	Binds the authentication policy as the secondary policy to use in a two-factor configuration. A user must then authenticate not only via a primary authentication method but also via a secondary authentication method. User groups are aggregated across both. The user name must be exactly the same for both authentication methods, but they can require different passwords.
bindpoint	<String>	Read-write	Bind point to which to bind the policy. Applies only to rewrite and cache policies. If you do not set this parameter, the policy is bound to REQ_DEFAULT or RES_DEFAULT, depending on whether the policy rule is a response-time or a request-time expression. Possible values = REQUEST, RESPONSE,

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
			ICA_REQUEST, OTHERTCP_REQUEST
__count	<Double>	Read-write	count parameter

Operations

(click to see Properties)

ADD:| DELETE:| GET| GET (ALL)| COUNT

Some options that you can use for each operations:

- **Getting warnings in response:**NITRO allows you to get warnings in an operation by specifying the "warning" query parameter as "yes". For example, to get warnings while connecting to the Citrix ADC appliance, the URL is as follows:

  http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/login?warning=yes

  If any, the warnings are displayed in the response payload with the HTTP code "209 X-NITRO-WARNING".

- **Authenticated access for individual NITRO operations:**NITRO allows you to logon to the Citrix ADC appliance to perform individual operations. You can use this option instead of creating a NITRO session (using the login object) and then using that session to perform all operations,

  To do this, you must specify the username and password in the request header of the NITRO request as follows:

  X-NITRO-USER:<username>

  X-NITRO-PASS:<password>

  Note: In such cases, make sure that the request header DOES not include the following:

  Cookie:NITRO_AUTH_TOKEN=<tokenvalue>

`*Note:`*

Mandatory parameters are marked in redand placeholder content is marked in <green>.

add

URL:http://<Citrix-ADC-IP-address(NSIP)/nitro/v1/config/vpnvserver_cspolicy_binding

HTTP Method:PUT

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Content-Type:application/json

Request Payload:

```
{

"vpnvserver_cspolicy_binding":{

<b>"name":<String_value>,

</b>"policy":<String_value>,

"priority":<Double_value>,

"secondary":<Boolean_value>,

"groupextraction":<Boolean_value>,

"gotopriorityexpression":<String_value>,

"bindpoint":<String_value>

}}
```

Response:

HTTP Status Code on Success: 201 Created HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error

delete

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/vpnvserver_cspolicy_binding/name_value<String>

Query-parameters:

args

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/vpnvserver_cspolicy_binding/name_value<String>?**args=policy:<String_value>,secondary:<Boolean_value>,groupextraction:<Boolean_value>,bindpoint:<**

HTTP Method:DELETE

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue>

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error

get

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/vpnvserver_cspolicy_binding/name_value<String>

Query-parameters:

filter

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/vpnvserver_cspolicy_binding/name_value<String>?**filter=property-name1:property-value1,property-name2:property-value2**

Use this query-parameter to get the filtered set of vpnvserver_cspolicy_binding resources configured on Citrix ADC. Filtering can be done on any of the properties of the resource.

pagination

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/vpnvserver_cspolicy_binding/name_value<String>?**pagesize=#no;pageno=#no**

Use this query-parameter to get the vpnvserver_cspolicy_binding resources in chunks.

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "vpnvserver_cspolicy_binding": [ {

"priority":<Double_value>,

"gotopriorityexpression":<String_value>,

"policy":<String_value>,

"groupextraction":<Boolean_value>,

"name":<String_value>,

"secondary":<Boolean_value>,

"bindpoint":<String_value>

}]}
```

get (all)

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/vpnvserver_cspolicy_binding

Query-parameters:

bulkbindings

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/vpnvserver_cspolicy_binding?**bulkbindings=yes**

NITRO allows you to fetch bindings in bulk.

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "vpnvserver_cspolicy_binding": [ {

"priority":<Double_value>,

"gotopriorityexpression":<String_value>,

"policy":<String_value>,

"groupextraction":<Boolean_value>,

"name":<String_value>,

"secondary":<Boolean_value>,

"bindpoint":<String_value>

}]}
```

count

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/vpnvserver_cspolicy_binding/name_value<String>?**count=yes**

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{"vpnvserver_cspolicy_binding": [ { "__count": "#no"} ] }
```

vpnvserver_feopolicy_binding

Binding object showing the feopolicy that can be bound to vpnvserver.

Properties

(click to see Operations)

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
priority	<Double>	Read-write	Integer specifying the policy's priority. The lower the number, the higher the priority. Policies are evaluated in the order of their priority numbers. Maximum

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
			value for default syntax policies is 2147483647 and for classic policies is 64000. Minimum value = 0 Maximum value = 2147483647
gotopriorityexpression	<String>	Read-write	Next priority expression.
policy	<String>	Read-write	Name of a policy to bind to the virtual server (for example, the name of an authentication, session, or endpoint analysis policy). Minimum length = 1
groupextraction	<Boolean>	Read-write	Binds the authentication policy to a tertiary chain which will be used only for group extraction. The user will not authenticate against this server, and this will only be called if primary and/or secondary authentication has succeeded.
name	<String>	Read-write	Name of the virtual server. Minimum length = 1

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
secondary	<Boolean>	Read-write	Binds the authentication policy as the secondary policy to use in a two-factor configuration. A user must then authenticate not only via a primary authentication method but also via a secondary authentication method. User groups are aggregated across both. The user name must be exactly the same for both authentication methods, but they can require different passwords.
bindpoint	<String>	Read-write	Bindpoint to which the policy is bound. Possible values = REQUEST, RESPONSE, ICA_REQUEST, OTHERTCP_REQUEST
__count	<Double>	Read-write	count parameter

Operations

(click to see Properties)

ADD:| DELETE:| GET| GET (ALL)| COUNT

Some options that you can use for each operations:

- **Getting warnings in response:**NITRO allows you to get warnings in an operation by specifying the "warning" query parameter as "yes". For example, to get warnings while connecting to the Citrix ADC appliance, the URL is as follows:

  http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/login?warning=yes

  If any, the warnings are displayed in the response payload with the HTTP code "209 X-NITRO-WARNING".

- **Authenticated access for individual NITRO operations:**NITRO allows you to logon to the Citrix ADC appliance to perform individual operations. You can use this option instead of creating a NITRO session (using the login object) and then using that session to perform all operations,

  To do this, you must specify the username and password in the request header of the NITRO request as follows:

  X-NITRO-USER:<username>

  X-NITRO-PASS:<password>

  Note: In such cases, make sure that the request header DOES not include the following:

  Cookie:NITRO_AUTH_TOKEN=<tokenvalue>

`*Note:`*

Mandatory parameters are marked in redand placeholder content is marked in <green>.

add

URL:http://<Citrix-ADC-IP-address(NSIP)/nitro/v1/config/vpnvserver_feopolicy_binding

HTTP Method:PUT

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Content-Type:application/json

Request Payload:

```
{

"vpnvserver_feopolicy_binding":{

<b>"name":<String_value>,

</b>"policy":<String_value>,

"priority":<Double_value>,

"secondary":<Boolean_value>,

"groupextraction":<Boolean_value>,

"gotopriorityexpression":<String_value>,

"bindpoint":<String_value>

}}
```

Response:

HTTP Status Code on Success: 201 Created HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error

delete

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/vpnvserver_feopolicy_binding/name_value<String>

Query-parameters:

args

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/vpnvserver_feopolicy_binding/name_value<String>?**args=policy:<String_value>,secondary:<Boolean_value>,groupextraction:<Boolean_value>,bindpoint:<**

HTTP Method:DELETE

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue>

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error

get

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/vpnvserver_feopolicy_binding/name_value<String>

Query-parameters:

filter

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/vpnvserver_feopolicy_binding/name_value<String>?**filter=property-name1:property-value1,property-name2:property-value2**

Use this query-parameter to get the filtered set of vpnvserver_feopolicy_binding resources configured on Citrix ADC. Filtering can be done on any of the properties of the resource.

pagination

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/vpnvserver_feopolicy_binding/name_value<String>?**pagesize=#no;pageno=#no**

Use this query-parameter to get the vpnvserver_feopolicy_binding resources in chunks.

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "vpnvserver_feopolicy_binding": [ {

"priority":<Double_value>,

"gotopriorityexpression":<String_value>,

"policy":<String_value>,

"groupextraction":<Boolean_value>,

"name":<String_value>,

"secondary":<Boolean_value>,

"bindpoint":<String_value>

}]}
```

get (all)

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/vpnvserver_feopolicy_binding

Query-parameters:

bulkbindings

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/vpnvserver_feopolicy_binding?**bulkbindings=yes**

NITRO allows you to fetch bindings in bulk.

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "vpnvserver_feopolicy_binding": [ {

"priority":<Double_value>,

"gotopriorityexpression":<String_value>,

"policy":<String_value>,

"groupextraction":<Boolean_value>,

"name":<String_value>,

"secondary":<Boolean_value>,

"bindpoint":<String_value>

}]}
```

count

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/vpnvserver_feopolicy_binding/name_value<String>?**count=yes**

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{"vpnvserver_feopolicy_binding": [ { "__count": "#no"} ] }
```

vpnvserver_icapolicy_binding

Binding object showing the icapolicy that can be bound to vpnvserver.

Properties

(click to see Operations)

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
priority	<Double>	Read-write	Integer specifying the policy's priority. The lower the number, the higher the priority. Policies are evaluated in the order of their priority numbers. Maximum value for default syntax policies is 2147483647 and for classic policies is 64000. Minimum value = 0 Maximum value = 2147483647

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
gotopriorityexpression	<String>	Read-write	Next priority expression.
policy	<String>	Read-write	The name of the policy, if any, bound to the VPN virtual server.
groupextraction	<Boolean>	Read-write	Binds the authentication policy to a tertiary chain which will be used only for group extraction. The user will not authenticate against this server, and this will only be called if primary and/or secondary authentication has succeeded.
name	<String>	Read-write	Name of the virtual server. Minimum length = 1
secondary	<Boolean>	Read-write	Binds the authentication policy as the secondary policy to use in a two-factor configuration. A user must then authenticate not only via a primary authentication method but also via a secondary authentication method. User groups are aggregated

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
			across both. The user name must be exactly the same for both authentication methods, but they can require different passwords.
bindpoint	<String>	Read-write	Bind point to which to bind the policy. Applies only to rewrite and cache policies. If you do not set this parameter, the policy is bound to REQ_DEFAULT or RES_DEFAULT, depending on whether the policy rule is a response-time or a request-time expression. Possible values = REQUEST, RESPONSE, ICA_REQUEST, OTHERTCP_REQUEST
__count	<Double>	Read-write	count parameter

Operations

(click to see Properties)

ADD:| DELETE:| GET| GET (ALL)| COUNT

Some options that you can use for each operations:

- **Getting warnings in response:**NITRO allows you to get warnings in an operation by specifying the "warning" query parameter as "yes". For example, to get warnings while connecting to the Citrix ADC appliance, the URL is as follows:

  http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/login?warning=yes

  If any, the warnings are displayed in the response payload with the HTTP code "209 X-NITRO-WARNING".

- **Authenticated access for individual NITRO operations:**NITRO allows you to logon to the Citrix ADC appliance to perform individual operations. You can use this option instead of creating a NITRO session (using the login object) and then using that session to perform all operations,

  To do this, you must specify the username and password in the request header of the NITRO request as follows:

  X-NITRO-USER:<username>

  X-NITRO-PASS:<password>

  Note: In such cases, make sure that the request header DOES not include the following:

  Cookie:NITRO_AUTH_TOKEN=<tokenvalue>

`*Note:`*

Mandatory parameters are marked in redand placeholder content is marked in <green>.

add

URL:http://<Citrix-ADC-IP-address(NSIP)/nitro/v1/config/vpnvserver_icapolicy_binding

HTTP Method:PUT

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Content-Type:application/json

Request Payload:

```
{
```

```
"vpnvserver_icapolicy_binding":{

<b>"name":<String_value>,

</b>"policy":<String_value>,

"priority":<Double_value>,

"secondary":<Boolean_value>,

"groupextraction":<Boolean_value>,

"gotopriorityexpression":<String_value>,

"bindpoint":<String_value>

}}
```

Response:

HTTP Status Code on Success: 201 Created HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error

delete

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/vpnvserver_icapolicy_binding/name_value<String>

Query-parameters:

args

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/vpnvserver_icapolicy_binding/name_value<String>?
args=policy:<String_value>,secondary:<Boolean_value>,groupextraction:<Boolean_value>,bindpoint:<

HTTP Method:DELETE

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue>

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error

get

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/vpnvserver_icapolicy_binding/name_value<String>

Query-parameters:

filter

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/vpnvserver_icapolicy_binding/name_value<String>?**filter=property-name1:property-value1,property-name2:property-value2**

Use this query-parameter to get the filtered set of vpnvserver_icapolicy_binding resources configured on Citrix ADC. Filtering can be done on any of the properties of the resource.

pagination

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/vpnvserver_icapolicy_binding/name_value<String>?**pagesize=#no;pageno=#no**

Use this query-parameter to get the vpnvserver_icapolicy_binding resources in chunks.

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "vpnvserver_icapolicy_binding": [ {

"priority":<Double_value>,

"gotopriorityexpression":<String_value>,

"policy":<String_value>,

"groupextraction":<Boolean_value>,

"name":<String_value>,

"secondary":<Boolean_value>,

"bindpoint":<String_value>

}]}
```

get (all)

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/vpnvserver_icapolicy_binding

Query-parameters:

bulkbindings

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/vpnvserver_icapolicy_binding?**bulkbindings=yes**

NITRO allows you to fetch bindings in bulk.

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "vpnvserver_icapolicy_binding": [ {

"priority":<Double_value>,

"gotopriorityexpression":<String_value>,

"policy":<String_value>,

"groupextraction":<Boolean_value>,

"name":<String_value>,

"secondary":<Boolean_value>,

"bindpoint":<String_value>

}]}
```

count

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/vpnvserver_icapolicy_binding/name_value<String>?**count=yes**

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{"vpnvserver_icapolicy_binding": [ { "__count": "#no"} ] }
```

vpnvserver_intranetip6_binding

Binding object showing the intranetip6 that can be bound to vpnvserver.

Properties

(click to see Operations)

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
intranetip6	<String>	Read-write	The network id for the range of intranet IP6 addresses or individual intranet ip to be bound to the vserver.
numaddr	<Double>	Read-write	The number of ipv6 addresses.
name	<String>	Read-write	Name of the virtual server. Minimum length = 1
acttype	<Double>	Read-only	.
__count	<Double>	Read-only	count parameter

Operations

(click to see Properties)

ADD:| DELETE:| GET| GET (ALL)| COUNT

Some options that you can use for each operations:

- **Getting warnings in response:**NITRO allows you to get warnings in an operation by specifying the "warning" query parameter as "yes". For example, to get warnings while connecting to the Citrix ADC appliance, the URL is as follows:

  http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/login?warning=yes

  If any, the warnings are displayed in the response payload with the HTTP code "209 X-NITRO-WARNING".

- **Authenticated access for individual NITRO operations:**NITRO allows you to logon to the Citrix ADC appliance to perform individual operations. You can use this option instead of creating a NITRO session (using the login object) and then using that session to perform all operations,

  To do this, you must specify the username and password in the request header of the NITRO request as follows:

  X-NITRO-USER:<username>

  X-NITRO-PASS:<password>

  Note: In such cases, make sure that the request header DOES not include the following:

  Cookie:NITRO_AUTH_TOKEN=<tokenvalue>

`*Note:`*

Mandatory parameters are marked in redand placeholder content is marked in <green>.

add

URL:http://<Citrix-ADC-IP-address(NSIP)/nitro/v1/config/vpnvserver_intranetip6_binding

HTTP Method:PUT

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Content-Type:application/json

Request Payload:

```
{

"vpnvserver_intranetip6_binding":{

<b>"name":<String_value>,

</b>"intranetip6":<String_value>,

"numaddr":<Double_value>

}}
```

Response:

HTTP Status Code on Success: 201 Created HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error

delete

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/vpnvserver_intranetip6_binding/name_value<String>

Query-parameters:

args

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/vpnvserver_intranetip6_binding/name_value<String>?**args=intranetip6:<String_value>,numaddr:<Double_value>**

HTTP Method:DELETE

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue>

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error

get

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/vpnvserver_intranetip6_binding/name_value<String>

Query-parameters:

filter

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/vpnvserver_intranetip6_binding/name_value<String>?**filter=property-name1:property-value1,property-name2:property-value2**

Use this query-parameter to get the filtered set of vpnvserver_intranetip6_binding resources configured on Citrix ADC. Filtering can be done on any of the properties of the resource.

pagination

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/vpnvserver_intranetip6_binding/name_value<String>?**pagesize=#no;pageno=#no**

Use this query-parameter to get the vpnvserver_intranetip6_binding resources in chunks.

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "vpnvserver_intranetip6_binding": [ {
```

```
"intranetip6":<String_value>,

"numaddr":<Double_value>,

"name":<String_value>,

"acttype":<Double_value>

}]}
```

get (all)

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/vpnvserver_intranetip6_binding

Query-parameters:

bulkbindings

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/vpnvserver_intranetip6_binding?**bulkbindings=yes**

NITRO allows you to fetch bindings in bulk.

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "vpnvserver_intranetip6_binding": [ {
```

```
"intranetip6":<String_value>,

"numaddr":<Double_value>,

"name":<String_value>,

"acttype":<Double_value>

}]}
```

count

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/vpnvserver_intranetip6_binding/name_value<String>?**count=yes**

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{"vpnvserver_intranetip6_binding": [ { "__count": "#no"} ] }
```

vpnvserver_intranetip_binding

Binding object showing the intranetip that can be bound to vpnvserver.

Properties

(click to see Operations)

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
name	<String>	Read-write	Name of the virtual server. Minimum length = 1
intranetip	<String>	Read-write	The network ID for the range of intranet IP addresses or individual intranet IP addresses to be bound to the virtual server.
netmask	<String>	Read-write	The netmask of the intranet IP address or range.
map	<String>	Read-only	Whether or not mapped IP addresses are ON or OFF. Mapped IP addresses are source IP addresses for the virtual servers running on the Citrix ADC. Mapped IP addresses are used by the system to connect to the backend servers. Possible values = ON, OFF
acttype	<Double>	Read-only	.
__count	<Double>	Read-only	count parameter

Operations

(click to see Properties)

ADD:| DELETE:| GET| GET (ALL)| COUNT

Some options that you can use for each operations:

- **Getting warnings in response:**NITRO allows you to get warnings in an operation by specifying the "warning" query parameter as "yes". For example, to get warnings while connecting to the Citrix ADC appliance, the URL is as follows:

  http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/login?warning=yes

  If any, the warnings are displayed in the response payload with the HTTP code "209 X-NITRO-WARNING".

- **Authenticated access for individual NITRO operations:**NITRO allows you to logon to the Citrix ADC appliance to perform individual operations. You can use this option instead of creating a NITRO session (using the login object) and then using that session to perform all operations,

  To do this, you must specify the username and password in the request header of the NITRO request as follows:

  X-NITRO-USER:<username>

  X-NITRO-PASS:<password>

  Note: In such cases, make sure that the request header DOES not include the following:

  Cookie:NITRO_AUTH_TOKEN=<tokenvalue>

`*Note:`*

Mandatory parameters are marked in redand placeholder content is marked in <green>.

add

URL:http://<Citrix-ADC-IP-address(NSIP)/nitro/v1/config/vpnvserver_intranetip_binding

HTTP Method:PUT

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Content-Type:application/json

Request Payload:

```
{

"vpnvserver_intranetip_binding":{

<b>"name":<String_value>,

</b>"intranetip":<String_value>,

"netmask":<String_value>

}}
```

Response:

HTTP Status Code on Success: 201 Created HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error

delete

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/vpnvserver_intranetip_binding/name_value<String>

Query-parameters:

args

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/vpnvserver_intranetip_binding/name_value<String>?**args=intranetip:<String_value>,netmask:<String_value>**

HTTP Method:DELETE

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue>

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error

get

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/vpnvserver_intranetip_binding/name_value<String>

Query-parameters:

filter

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/vpnvserver_intranetip_binding/name_value<String>?**filter=property-name1:property-value1,property-name2:property-value2**

Use this query-parameter to get the filtered set of vpnvserver_intranetip_binding resources configured on Citrix ADC. Filtering can be done on any of the properties of the resource.

pagination

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/vpnvserver_intranetip_binding/name_value<String>?**pagesize=#no;pageno=#no**

Use this query-parameter to get the vpnvserver_intranetip_binding resources in chunks.

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "vpnvserver_intranetip_binding": [ {
```

```
"name":<String_value>,

"intranetip":<String_value>,

"netmask":<String_value>,

"map":<String_value>,

"acttype":<Double_value>

}]}
```

get (all)

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/vpnvserver_intranetip_binding

Query-parameters:

bulkbindings

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/vpnvserver_intranetip_binding?**bulkbindings=yes**

NITRO allows you to fetch bindings in bulk.

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "vpnvserver_intranetip_binding": [ {

"name":<String_value>,

"intranetip":<String_value>,

"netmask":<String_value>,

"map":<String_value>,

"acttype":<Double_value>

}]}
```

count

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/vpnvserver_intranetip_binding/name_value<String>?**count=yes**

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{"vpnvserver_intranetip_binding": [ { "__count": "#no"} ] }
```

vpnvserver_responderpolicy_binding

Binding object showing the responderpolicy that can be bound to vpnvserver.

Properties

(click to see Operations)

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
priority	<Double>	Read-write	Integer specifying the policy's priority. The lower the number, the higher the priority. Policies are evaluated in the order of their priority numbers. Maximum value for default syntax policies is 2147483647 and for classic policies is 64000. Minimum value = 0 Maximum value = 2147483647
gotopriorityexpression	<String>	Read-write	Next priority expression.
policy	<String>	Read-write	The name of the policy, if any, bound to the VPN virtual server.
groupextraction	<Boolean>	Read-write	Binds the authentication policy to a tertiary chain which will be used only for group extraction. The user will not authenticate against this server,

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
			and this will only be called if primary and/or secondary authentication has succeeded.
name	<String>	Read-write	Name of the virtual server. Minimum length = 1
secondary	<Boolean>	Read-write	Binds the authentication policy as the secondary policy to use in a two-factor configuration. A user must then authenticate not only via a primary authentication method but also via a secondary authentication method. User groups are aggregated across both. The user name must be exactly the same for both authentication methods, but they can require different passwords.
bindpoint	<String>	Read-write	Bind point to which to bind the policy. Applies only to rewrite and cache policies. If you do not set this parameter, the policy is bound to REQ_DEFAULT or

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
			RES_DEFAULT, depending on whether the policy rule is a response-time or a request-time expression. Possible values = REQUEST, RESPONSE, ICA_REQUEST, OTHERTCP_REQUEST
__count	<Double>	Read-write	count parameter

Operations

(click to see Properties)

ADD:| DELETE:| GET| GET (ALL)| COUNT

Some options that you can use for each operations:

- **Getting warnings in response:**NITRO allows you to get warnings in an operation by specifying the "warning" query parameter as "yes". For example, to get warnings while connecting to the Citrix ADC appliance, the URL is as follows:

  http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/login?warning=yes

  If any, the warnings are displayed in the response payload with the HTTP code "209 X-NITRO-WARNING".

- **Authenticated access for individual NITRO operations:**NITRO allows you to logon to the Citrix ADC appliance to perform individual operations. You can use this option instead of creating a NITRO session (using the login object) and then using that session to perform all operations,

  To do this, you must specify the username and password in the request header of the NITRO request as follows:

  X-NITRO-USER:<username>

X-NITRO-PASS:<password>

Note: In such cases, make sure that the request header DOES not include the following:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue>

`*Note:`*

Mandatory parameters are marked in redand placeholder content is marked in <green>.

add

URL:http://<Citrix-ADC-IP-address(NSIP)/nitro/v1/config/vpnvserver_responderpolicy_binding

HTTP Method:PUT

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Content-Type:application/json

Request Payload:

```
{
"vpnvserver_responderpolicy_binding":{
<b>"name":<String_value>,
</b>"policy":<String_value>,
"priority":<Double_value>,
"secondary":<Boolean_value>,
"groupextraction":<Boolean_value>,
"gotopriorityexpression":<String_value>,
"bindpoint":<String_value>
}}
```

Response:

HTTP Status Code on Success: 201 Created HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error

delete

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/vpnvserver_responderpolicy_binding/name_value<String>

Query-parameters:

args

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/vpnvserver_responderpolicy_binding/name_value<String>?**args=policy:<String_value>,secondary:<Boolean_value>,groupextraction:<Boolean_value>,bindpoint:<**

HTTP Method:DELETE

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue>

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error

get

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/vpnvserver_responderpolicy_binding/name_value<String>

Query-parameters:

filter

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/vpnvserver_responderpolicy_binding/name_value<String>?**filter=property-name1:property-value1,property-name2:property-value2**

Use this query-parameter to get the filtered set of vpnvserver_responderpolicy_binding resources configured on Citrix ADC. Filtering can be done on any of the properties of the resource.

pagination

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/vpnvserver_responderpolicy_binding/name_value<String>?**pagesize=#no;pageno=#no**

Use this query-parameter to get the vpnvserver_responderpolicy_binding resources in chunks.

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "vpnvserver_responderpolicy_binding": [ {

"priority":<Double_value>,

"gotopriorityexpression":<String_value>,

"policy":<String_value>,

"groupextraction":<Boolean_value>,

"name":<String_value>,

"secondary":<Boolean_value>,

"bindpoint":<String_value>

}]}
```

get (all)

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/vpnvserver_responderpolicy_binding

Query-parameters:

bulkbindings

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/vpnvserver_responderpolicy_binding?**bulkbindings=yes**

NITRO allows you to fetch bindings in bulk.

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "vpnvserver_responderpolicy_binding": [ {

"priority":<Double_value>,

"gotopriorityexpression":<String_value>,

"policy":<String_value>,

"groupextraction":<Boolean_value>,

"name":<String_value>,

"secondary":<Boolean_value>,
```

```
"bindpoint":<String_value>

}]}
```

count

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/vpnvserver_responderpolicy_binding/name_value<String>?**count=yes**

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{"vpnvserver_responderpolicy_binding": [ { "__count": "#no"} ] }
```

vpnvserver_rewritepolicy_binding

Binding object showing the rewritepolicy that can be bound to vpnvserver.

Properties

(click to see Operations)

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
priority	<Double>	Read-write	Integer specifying the policy's priority.

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
			The lower the number, the higher the priority. Policies are evaluated in the order of their priority numbers. Maximum value for default syntax policies is 2147483647 and for classic policies is 64000. Minimum value = 0 Maximum value = 2147483647
gotopriorityexpression	<String>	Read-write	Next priority expression.
policy	<String>	Read-write	The name of the policy, if any, bound to the VPN virtual server.
groupextraction	<Boolean>	Read-write	Binds the authentication policy to a tertiary chain which will be used only for group extraction. The user will not authenticate against this server, and this will only be called if primary and/or secondary authentication has succeeded.
name	<String>	Read-write	Name of the virtual server. Minimum length = 1

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
secondary	<Boolean>	Read-write	Binds the authentication policy as the secondary policy to use in a two-factor configuration. A user must then authenticate not only via a primary authentication method but also via a secondary authentication method. User groups are aggregated across both. The user name must be exactly the same for both authentication methods, but they can require different passwords.
bindpoint	<String>	Read-write	Bindpoint to which the policy is bound. Possible values = REQUEST, RESPONSE, ICA_REQUEST, OTHERTCP_REQUEST
__count	<Double>	Read-write	count parameter

Operations

(click to see Properties)

ADD:| DELETE:| GET| GET (ALL)| COUNT

Some options that you can use for each operations:

- **Getting warnings in response:**NITRO allows you to get warnings in an operation by specifying the "warning" query parameter as "yes". For example, to get warnings while connecting to the Citrix ADC appliance, the URL is as follows:

  http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/login?warning=yes

  If any, the warnings are displayed in the response payload with the HTTP code "209 X-NITRO-WARNING".

- **Authenticated access for individual NITRO operations:**NITRO allows you to logon to the Citrix ADC appliance to perform individual operations. You can use this option instead of creating a NITRO session (using the login object) and then using that session to perform all operations,

  To do this, you must specify the username and password in the request header of the NITRO request as follows:

  X-NITRO-USER:<username>

  X-NITRO-PASS:<password>

  Note: In such cases, make sure that the request header DOES not include the following:

  Cookie:NITRO_AUTH_TOKEN=<tokenvalue>

`*Note:`*

Mandatory parameters are marked in redand placeholder content is marked in <green>.

add

URL:http://<Citrix-ADC-IP-address(NSIP)/nitro/v1/config/vpnvserver_rewritepolicy_binding

HTTP Method:PUT

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Content-Type:application/json

Request Payload:

```
{

"vpnvserver_rewritepolicy_binding":{

<b>"name":<String_value>,

</b>"policy":<String_value>,

"priority":<Double_value>,

"secondary":<Boolean_value>,

"groupextraction":<Boolean_value>,

"gotopriorityexpression":<String_value>,

"bindpoint":<String_value>

}}
```

Response:

HTTP Status Code on Success: 201 Created HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error

delete

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/vpnvserver_rewritepolicy_binding/name_value<String>

Query-parameters:

args

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/vpnvserver_rewritepolicy_binding/name_value<String>?**args=policy:<String_value>,secondary:<Boolean_value>,groupextraction:<Boolean_value>,bindpoint:<**

HTTP Method:DELETE

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue>

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error

get

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/vpnvserver_rewritepolicy_binding/name_value<String>

Query-parameters:

filter

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/vpnvserver_rewritepolicy_binding/name_value<String>?**filter=property-name1:property-value1,property-name2:property-value2**

Use this query-parameter to get the filtered set of vpnvserver_rewritepolicy_binding resources configured on Citrix ADC. Filtering can be done on any of the properties of the resource.

pagination

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/vpnvserver_rewritepolicy_binding/name_value<String>?**pagesize=#no;pageno=#no**

Use this query-parameter to get the vpnvserver_rewritepolicy_binding resources in chunks.

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "vpnvserver_rewritepolicy_binding": [ {

"priority":<Double_value>,

"gotopriorityexpression":<String_value>,

"policy":<String_value>,

"groupextraction":<Boolean_value>,

"name":<String_value>,

"secondary":<Boolean_value>,

"bindpoint":<String_value>

}]}
```

get (all)

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/vpnvserver_rewritepolicy_binding

Query-parameters:

bulkbindings

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/vpnvserver_rewritepolicy_binding?**bulkbindings=yes**

NITRO allows you to fetch bindings in bulk.

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "vpnvserver_rewritepolicy_binding": [ {

"priority":<Double_value>,

"gotopriorityexpression":<String_value>,

"policy":<String_value>,

"groupextraction":<Boolean_value>,

"name":<String_value>,

"secondary":<Boolean_value>,

"bindpoint":<String_value>

}]}
```

count

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/vpnvserver_rewritepolicy_binding/name_value<String>?**count=yes**

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{"vpnvserver_rewritepolicy_binding": [ { "__count": "#no"} ] }
```

vpnvserver_sharefileserver_binding

Binding object showing the sharefileserver that can be bound to vpnvserver.

Properties

(click to see Operations)

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
name	<String>	Read-write	Name of the virtual server. Minimum length = 1
sharefile	<String>	Read-write	Configured ShareFile server in XenMobile deployment. Format IP:PORT / FQDN:PORT.
acttype	<Double>	Read-only	.
__count	<Double>	Read-only	count parameter

Operations

(click to see Properties)

ADD:| DELETE:| GET| GET (ALL)| COUNT

Some options that you can use for each operations:

- **Getting warnings in response:**NITRO allows you to get warnings in an operation by specifying the "warning" query parameter as "yes". For example, to get warnings while connecting to the Citrix ADC appliance, the URL is as follows:

  http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/login?warning=yes

  If any, the warnings are displayed in the response payload with the HTTP code "209 X-NITRO-WARNING".

- **Authenticated access for individual NITRO operations:**NITRO allows you to logon to the Citrix ADC appliance to perform individual operations. You can use this option instead of creating a NITRO session (using the login object) and then using that session to perform all operations,

  To do this, you must specify the username and password in the request header of the NITRO request as follows:

  X-NITRO-USER:<username>

  X-NITRO-PASS:<password>

  Note: In such cases, make sure that the request header DOES not include the following:

  Cookie:NITRO_AUTH_TOKEN=<tokenvalue>

`*Note:`*

Mandatory parameters are marked in redand placeholder content is marked in <green>.

add

URL:http://<Citrix-ADC-IP-address(NSIP)/nitro/v1/config/vpnvserver_sharefileserver_binding

HTTP Method:PUT

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Content-Type:application/json

Request Payload:

```
{

"vpnvserver_sharefileserver_binding":{

<b>"name":<String_value>,

</b>"sharefile":<String_value>

}}
```

Response:

HTTP Status Code on Success: 201 Created HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error

delete

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/vpnvserver_sharefileserver_binding/name_value<String>

Query-parameters:

args

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/vpnvserver_sharefileserver_binding/name_value<String>?**args=sharefile:<String_value>**

HTTP Method:DELETE

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue>

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error

get

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/vpnvserver_sharefileserver_binding/name_value<String>

Query-parameters:

filter

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/vpnvserver_sharefileserver_binding/name_value<String>?**filter=property-name1:property-value1,property-name2:property-value2**

Use this query-parameter to get the filtered set of vpnvserver_sharefileserver_binding resources configured on Citrix ADC. Filtering can be done on any of the properties of the resource.

pagination

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/vpnvserver_sharefileserver_binding/name_value<String>?**pagesize=#no;pageno=#no**

Use this query-parameter to get the vpnvserver_sharefileserver_binding resources in chunks.

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "vpnvserver_sharefileserver_binding": [ {

"name":<String_value>,

"sharefile":<String_value>,
```

```
"acttype":<Double_value>

}]}
```

get (all)

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/vpnvserver_sharefileserver_binding

Query-parameters:

bulkbindings

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/vpnvserver_sharefileserver_binding?**bulkbindings=yes**

NITRO allows you to fetch bindings in bulk.

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "vpnvserver_sharefileserver_binding": [ {

"name":<String_value>,

"sharefile":<String_value>,

"acttype":<Double_value>
```

```
}]}
```

count

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/vpnvserver_sharefileserver_binding/name_value<String>?**count=yes**

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{"vpnvserver_sharefileserver_binding": [ { "__count": "#no"} ] }
```

vpnvserver_staserver_binding

Binding object showing the staserver that can be bound to vpnvserver.

Properties

(click to see Operations)

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
name	<String>	Read-write	Name of the virtual server. Minimum length = 1

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
staaddresstype	<String>	Read-write	Type of the STA server address(ipv4/v6). Possible values = IPV4, IPV6
staserver	<String>	Read-write	Configured Secure Ticketing Authority (STA) server.
stastate	<String>	Read-only	State of the STA Server. If Authority ID is set then STA Server is UP else DOWN. Possible values = UP, DOWN
acttype	<Double>	Read-only	.
staauthid	<String>	Read-only	Authority ID of the STA Server. Authority ID is used to match incoming STA tickets in the SOCKS/CGP protocol with the right STA server.
__count	<Double>	Read-only	count parameter

Operations

(click to see Properties)

ADD:| DELETE:| GET| GET (ALL)| COUNT

Some options that you can use for each operations:

- **Getting warnings in response:**NITRO allows you to get warnings in an operation by specifying the "warning" query parameter as "yes". For example, to get warnings while connecting to the Citrix ADC appliance, the URL is as follows:

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/login?warning=yes

If any, the warnings are displayed in the response payload with the HTTP code "209 X-NITRO-WARNING".

- **Authenticated access for individual NITRO operations:**NITRO allows you to logon to the Citrix ADC appliance to perform individual operations. You can use this option instead of creating a NITRO session (using the login object) and then using that session to perform all operations,

  To do this, you must specify the username and password in the request header of the NITRO request as follows:

  X-NITRO-USER:<username>

  X-NITRO-PASS:<password>

  Note: In such cases, make sure that the request header DOES not include the following:

  Cookie:NITRO_AUTH_TOKEN=<tokenvalue>

`*Note:`*

Mandatory parameters are marked in redand placeholder content is marked in <green>.

add

URL:http://<Citrix-ADC-IP-address(NSIP)/nitro/v1/config/vpnvserver_staserver_binding

HTTP Method:PUT

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Content-Type:application/json

Request Payload:

```
{

"vpnvserver_staserver_binding":{

<b>"name":<String_value>,

</b>"staserver":<String_value>,
```

```
"staaddresstype":<String_value>

}}
```

Response:

HTTP Status Code on Success: 201 Created HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error

delete

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/vpnvserver_staserver_binding/name_value<String>

Query-parameters:

args

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/vpnvserver_staserver_binding/name_value<String>?**args=staserver:<String_value>**

HTTP Method:DELETE

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue>

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error

get

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/vpnvserver_staserver_binding/name_value<String>

Query-parameters:

filter

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/vpnvserver_staserver_binding/name_value<String>?**filter=property-name1:property-value1,property-name2:property-value2**

Use this query-parameter to get the filtered set of vpnvserver_staserver_binding resources configured on Citrix ADC. Filtering can be done on any of the properties of the resource.

pagination

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/vpnvserver_staserver_binding/name_value<String>?**pagesize=#no;pageno=#no**

Use this query-parameter to get the vpnvserver_staserver_binding resources in chunks.

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "vpnvserver_staserver_binding": [ {

"name":<String_value>,

"staaddresstype":<String_value>,

"staserver":<String_value>,

"stastate":<String_value>,

"acttype":<Double_value>,

"staauthid":<String_value>
```

```
}]}
```

get (all)

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/vpnvserver_staserver_binding

Query-parameters:

bulkbindings

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/vpnvserver_staserver_binding?**bulkbindings=yes**

NITRO allows you to fetch bindings in bulk.

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "vpnvserver_staserver_binding": [ {

"name":<String_value>,

"staaddresstype":<String_value>,

"staserver":<String_value>,

"stastate":<String_value>,
```

```
"acttype":<Double_value>,

"staauthid":<String_value>

}]}
```

count

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/vpnvserver_staserver_binding/name_value<String>?**count=yes**

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{"vpnvserver_staserver_binding": [ { "__count": "#no"} ] }
```

vpnvserver_vpnclientlessaccesspolicy_binding

Binding object showing the vpnclientlessaccesspolicy that can be bound to vpnvserver.

Properties

(click to see Operations)

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
priority	<Double>	Read-write	Integer specifying the policy's priority. The lower the number, the higher the priority. Policies are evaluated in the order of their priority numbers. Maximum value for default syntax policies is 2147483647 and for classic policies is 64000. Minimum value = 0 Maximum value = 2147483647
gotopriorityexpression	<String>	Read-write	Next priority expression.
policy	<String>	Read-write	The name of the policy, if any, bound to the VPN virtual server.
groupextraction	<Boolean>	Read-write	Binds the authentication policy to a tertiary chain which will be used only for group extraction. The user will not authenticate against this server, and this will only be called if primary and/or secondary authentication has succeeded.

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
name	<String>	Read-write	Name of the virtual server. Minimum length = 1
secondary	<Boolean>	Read-write	Binds the authentication policy as the secondary policy to use in a two-factor configuration. A user must then authenticate not only via a primary authentication method but also via a secondary authentication method. User groups are aggregated across both. The user name must be exactly the same for both authentication methods, but they can require different passwords.
bindpoint	<String>	Read-write	Bindpoint to which the policy is bound. Possible values = REQUEST, RESPONSE, ICA_REQUEST, OTHERTCP_REQUEST
acttype	<Double>	Read-only	.
__count	<Double>	Read-only	count parameter

Operations

(click to see Properties)

ADD:| DELETE:| GET| GET (ALL)| COUNT

Some options that you can use for each operations:

- **Getting warnings in response:**NITRO allows you to get warnings in an operation by specifying the "warning" query parameter as "yes". For example, to get warnings while connecting to the Citrix ADC appliance, the URL is as follows:

  http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/login?warning=yes

  If any, the warnings are displayed in the response payload with the HTTP code "209 X-NITRO-WARNING".

- **Authenticated access for individual NITRO operations:**NITRO allows you to logon to the Citrix ADC appliance to perform individual operations. You can use this option instead of creating a NITRO session (using the login object) and then using that session to perform all operations,

  To do this, you must specify the username and password in the request header of the NITRO request as follows:

  X-NITRO-USER:<username>

  X-NITRO-PASS:<password>

  Note: In such cases, make sure that the request header DOES not include the following:

  Cookie:NITRO_AUTH_TOKEN=<tokenvalue>

`*Note:`*

Mandatory parameters are marked in redand placeholder content is marked in <green>.

add

URL:http://<Citrix-ADC-IP-address(NSIP)/nitro/v1/config/vpnvserver_vpnclientlessaccesspolicy_binding

HTTP Method:PUT

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Content-Type:application/json

Request Payload:

```
{

"vpnvserver_vpnclientlessaccesspolicy_binding":{

<b>"name":<String_value>,

</b>"policy":<String_value>,

"priority":<Double_value>,

"secondary":<Boolean_value>,

"groupextraction":<Boolean_value>,

"gotopriorityexpression":<String_value>,

"bindpoint":<String_value>

}}
```

Response:

HTTP Status Code on Success: 201 Created HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error

delete

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/vpnvserver_vpnclientlessaccesspolicy_binding/name_value<String>

Query-parameters:

args

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/vpnvserver_vpnclientlessaccesspolicy_binding/name_value<String>?**args=policy:<String_value>,secondary:<Boolean_value>,groupextraction:<Boolean_value>,bindpoint:<**

HTTP Method:DELETE

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue>

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error

get

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/vpnvserver_vpnclientlessaccesspolicy_binding/name_value<String>

Query-parameters:

filter

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/vpnvserver_vpnclientlessaccesspolicy_binding/name_value<String>?**filter=property-name1:property-value1,property-name2:property-value2**

Use this query-parameter to get the filtered set of vpnvserver_vpnclientlessaccesspolicy_binding resources configured on Citrix ADC. Filtering can be done on any of the properties of the resource.

pagination

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/vpnvserver_vpnclientlessaccesspolicy_binding/name_value<String>?**pagesize=#no;pageno=#no**

Use this query-parameter to get the vpnvserver_vpnclientlessaccesspolicy_binding resources in chunks.

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "vpnvserver_vpnclientlessaccesspolicy_binding": [ {

"priority":<Double_value>,

"gotopriorityexpression":<String_value>,

"policy":<String_value>,

"groupextraction":<Boolean_value>,

"name":<String_value>,

"secondary":<Boolean_value>,

"bindpoint":<String_value>,

"acttype":<Double_value>

}]}
```

get (all)

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/vpnvserver_vpnclientlessaccesspolicy_binding

Query-parameters:

bulkbindings

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/vpnvserver_vpnclientlessaccesspolicy_binding?**bulkbindings=yes**

NITRO allows you to fetch bindings in bulk.

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "vpnvserver_vpnclientlessaccesspolicy_binding": [ {

"priority":<Double_value>,

"gotopriorityexpression":<String_value>,

"policy":<String_value>,

"groupextraction":<Boolean_value>,

"name":<String_value>,

"secondary":<Boolean_value>,

"bindpoint":<String_value>,

"acttype":<Double_value>

}]}
```

count

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/vpnvserver_vpnclientlessaccesspolicy_binding/name_value<String>?**count=yes**

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{"vpnvserver_vpnclientlessaccesspolicy_binding": [ { "__count": "#no"} ] }
```

vpnvserver_vpnepaprofile_binding

Binding object showing the vpnepaprofile that can be bound to vpnvserver.

Properties

(click to see Operations)

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
name	<String>	Read-write	Name of the virtual server. Minimum length = 1
epaprofile	<String>	Read-write	Advanced EPA profile to bind.

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
epaprofileoptional	<Boolean>	Read-write	Mark the EPA profile optional for preauthentication EPA profile. User would be shown a logon page even if the EPA profile fails to evaluate.
acttype	<Double>	Read-only	.
__count	<Double>	Read-only	count parameter

Operations

(click to see Properties)

ADD:| DELETE:| GET| GET (ALL)| COUNT

Some options that you can use for each operations:

- **Getting warnings in response:**NITRO allows you to get warnings in an operation by specifying the "warning" query parameter as "yes". For example, to get warnings while connecting to the Citrix ADC appliance, the URL is as follows:

  http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/login?warning=yes

  If any, the warnings are displayed in the response payload with the HTTP code "209 X-NITRO-WARNING".

- **Authenticated access for individual NITRO operations:**NITRO allows you to logon to the Citrix ADC appliance to perform individual operations. You can use this option instead of creating a NITRO session (using the login object) and then using that session to perform all operations,

  To do this, you must specify the username and password in the request header of the NITRO request as follows:

  X-NITRO-USER:<username>

  X-NITRO-PASS:<password>

Note: In such cases, make sure that the request header DOES not include the following:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue>

`*Note:` *

Mandatory parameters are marked in redand placeholder content is marked in <green>.

add

URL:http://<Citrix-ADC-IP-address(NSIP)/nitro/v1/config/vpnvserver_vpnepaprofile_binding

HTTP Method:PUT

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Content-Type:application/json

Request Payload:

```
{

"vpnvserver_vpnepaprofile_binding":{

<b>"name":<String_value>,

</b>"epaprofile":<String_value>,

"epaprofileoptional":<Boolean_value>

}}
```

Response:

HTTP Status Code on Success: 201 Created HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error

delete

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/vpnvserver_vpnepaprofile_binding/name_value<String>

Query-parameters:

args

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/vpnvserver_vpnepaprofile_binding/name_value<String>?**args=epaprofile:<String_value>**

HTTP Method:DELETE

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue>

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error

get

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/vpnvserver_vpnepaprofile_binding/name_value<String>

Query-parameters:

filter

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/vpnvserver_vpnepaprofile_binding/name_value<String>?**filter=property-name1:property-value1,property-name2:property-value2**

Use this query-parameter to get the filtered set of vpnvserver_vpnepaprofile_binding resources configured on Citrix ADC. Filtering can be done on any of the properties of the resource.

pagination

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/vpnvserver_vpnepaprofile_binding/name_value<String>?**pagesize=#no;pageno=#no**

Use this query-parameter to get the vpnvserver_vpnepaprofile_binding resources in chunks.

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "vpnvserver_vpnepaprofile_binding": [ {

"name":<String_value>,

"epaprofile":<String_value>,

"epaprofileoptional":<Boolean_value>,

"acttype":<Double_value>

}]}
```

get (all)

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/vpnvserver_vpnepaprofile_binding

Query-parameters:

bulkbindings

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/vpnvserver_vpnepaprofile_binding?**bulkbindings=yes**

NITRO allows you to fetch bindings in bulk.

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "vpnvserver_vpnepaprofile_binding": [ {

"name":<String_value>,

"epaprofile":<String_value>,

"epaprofileoptional":<Boolean_value>,

"acttype":<Double_value>

}]}
```

count

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/vpnvserver_vpnepaprofile_binding/name_value<String>?**count=yes**

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{"vpnvserver_vpnepaprofile_binding": [ { "__count": "#no"} ] }
```

vpnvserver_vpneula_binding

Binding object showing the vpneula that can be bound to vpnvserver.

Properties

(click to see Operations)

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
eula	<String>	Read-write	Name of the EULA bound to VPN vserver.
name	<String>	Read-write	Name of the virtual server. Minimum length = 1
acttype	<Double>	Read-only	.
__count	<Double>	Read-only	count parameter

Operations

(click to see Properties)

ADD:| DELETE:| GET| GET (ALL)| COUNT

Some options that you can use for each operations:

- **Getting warnings in response:**NITRO allows you to get warnings in an operation by specifying the "warning" query parameter as "yes". For example, to get warnings while connecting to the Citrix ADC appliance, the URL is as follows:

  http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/login?warning=yes

  If any, the warnings are displayed in the response payload with the HTTP code "209 X-NITRO-WARNING".

- **Authenticated access for individual NITRO operations:**NITRO allows you to logon to the Citrix ADC appliance to perform individual operations. You can use this option instead of creating a NITRO session (using the login object) and then using that session to perform all operations,

  To do this, you must specify the username and password in the request header of the NITRO request as follows:

  X-NITRO-USER:<username>

  X-NITRO-PASS:<password>

  Note: In such cases, make sure that the request header DOES not include the following:

  Cookie:NITRO_AUTH_TOKEN=<tokenvalue>

`*Note:`*

Mandatory parameters are marked in redand placeholder content is marked in <green>.

add

URL:http://<Citrix-ADC-IP-address(NSIP)/nitro/v1/config/vpnvserver_vpneula_binding

HTTP Method:PUT

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Content-Type:application/json

Request Payload:

```
{

"vpnvserver_vpneula_binding":{

<b>"name":<String_value>,

</b>"eula":<String_value>

}}
```

Response:

HTTP Status Code on Success: 201 Created HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error

delete

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/vpnvserver_vpneula_binding/name_value<String>

Query-parameters:

args

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/vpnvserver_vpneula_binding/name_value<String>?**args=eula:<String_value>**

HTTP Method:DELETE

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue>

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error

get

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/vpnvserver_vpneula_binding/name_value<String>

Query-parameters:

filter

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/vpnvserver_vpneula_binding/name_value<String>?**filter=property-name1:property-value1,property-name2:property-value2**

Use this query-parameter to get the filtered set of vpnvserver_vpneula_binding resources configured on Citrix ADC. Filtering can be done on any of the properties of the resource.

pagination

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/vpnvserver_vpneula_binding/name_value<String>?**pagesize=#no;pageno=#no**

Use this query-parameter to get the vpnvserver_vpneula_binding resources in chunks.

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "vpnvserver_vpneula_binding": [ {

"eula":<String_value>,

"name":<String_value>,

"acttype":<Double_value>

}]}
```

get (all)

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/vpnvserver_vpneula_binding

Query-parameters:

bulkbindings

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/vpnvserver_vpneula_binding?**bulkbindings=yes**

NITRO allows you to fetch bindings in bulk.

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "vpnvserver_vpneula_binding": [ {

"eula":<String_value>,

"name":<String_value>,

"acttype":<Double_value>

}]}
```

count

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/vpnvserver_vpneula_binding/name_value<String>?**count=yes**

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{"vpnvserver_vpneula_binding": [ { "__count": "#no"} ] }
```

vpnvserver_vpnintranetapplication_binding

Binding object showing the vpnintranetapplication that can be bound to vpnvserver.

Properties

(click to see Operations)

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
name	<String>	Read-write	Name of the virtual server. Minimum length = 1
intranetapplication	<String>	Read-write	The intranet VPN application.
acttype	<Double>	Read-only	.
__count	<Double>	Read-only	count parameter

Operations

(click to see Properties)

ADD:| DELETE:| GET| GET (ALL)| COUNT

Some options that you can use for each operations:

- **Getting warnings in response:**NITRO allows you to get warnings in an operation by specifying the "warning" query parameter as "yes". For example, to get warnings while connecting to the Citrix ADC appliance, the URL is as follows:

  http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/login?warning=yes

  If any, the warnings are displayed in the response payload with the HTTP code "209 X-NITRO-WARNING".

- **Authenticated access for individual NITRO operations:**NITRO allows you to logon to the Citrix ADC appliance to perform individual operations. You can use this option instead of creating a NITRO session (using the login object) and then using that session to perform all operations,

  To do this, you must specify the username and password in the request header of the NITRO request as follows:

  X-NITRO-USER:<username>

  X-NITRO-PASS:<password>

  Note: In such cases, make sure that the request header DOES not include the following:

  Cookie:NITRO_AUTH_TOKEN=<tokenvalue>

`*Note:`*

Mandatory parameters are marked in redand placeholder content is marked in <green>.

add

URL:http://<Citrix-ADC-IP-address(NSIP)/nitro/v1/config/vpnvserver_vpnintranetapplication_binding

HTTP Method:PUT

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Content-Type:application/json

Request Payload:

```
{
```

```
"vpnvserver_vpnintranetapplication_binding":{

<b>"name":<String_value>,

</b>"intranetapplication":<String_value>

}}
```

Response:

HTTP Status Code on Success: 201 Created HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error

delete

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/vpnvserver_vpnintranetapplication_binding/name_value<String>

Query-parameters:

args

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/vpnvserver_vpnintranetapplication_binding/name_value<String>?**args=intranetapplication:<String_value>**

HTTP Method:DELETE

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue>

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error

get

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/vpnvserver_vpnintranetapplication_binding/name_value<String>

Query-parameters:

filter

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/vpnvserver_vpnintranetapplication_binding/name_value<String>?**filter=property-name1:property-value1,property-name2:property-value2**

Use this query-parameter to get the filtered set of vpnvserver_vpnintranetapplication_binding resources configured on Citrix ADC. Filtering can be done on any of the properties of the resource.

pagination

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/vpnvserver_vpnintranetapplication_binding/name_value<String>?**pagesize=#no;pageno=#no**

Use this query-parameter to get the vpnvserver_vpnintranetapplication_binding resources in chunks.

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "vpnvserver_vpnintranetapplication_binding": [ {

"name":<String_value>,

"intranetapplication":<String_value>,

"acttype":<Double_value>
```

```
}]}
```

get (all)

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/vpnvserver_vpnintranetapplication_binding

Query-parameters:

bulkbindings

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/vpnvserver_vpnintranetapplication_binding?**bulkbindings=yes**

NITRO allows you to fetch bindings in bulk.

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "vpnvserver_vpnintranetapplication_binding": [ {

"name":<String_value>,

"intranetapplication":<String_value>,

"acttype":<Double_value>

}]}
```

count

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/vpnvserver_vpnintranetapplication_binding/name_value<String>?**count=yes**

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{"vpnvserver_vpnintranetapplication_binding": [ { "__count": "#no"} ] }
```

vpnvserver_vpnnexthopserver_binding

Binding object showing the vpnnexthopserver that can be bound to vpnvserver.

Properties

(click to see Operations)

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
name	<String>	Read-write	Name of the virtual server. Minimum length = 1

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
nexthopserver	<String>	Read-write	The name of the next hop server bound to the VPN virtual server.
acttype	<Double>	Read-only	.
__count	<Double>	Read-only	count parameter

Operations

(click to see Properties)

ADD:| DELETE:| GET| GET (ALL)| COUNT

Some options that you can use for each operations:

- **Getting warnings in response:**NITRO allows you to get warnings in an operation by specifying the "warning" query parameter as "yes". For example, to get warnings while connecting to the Citrix ADC appliance, the URL is as follows:

  http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/login?warning=yes

  If any, the warnings are displayed in the response payload with the HTTP code "209 X-NITRO-WARNING".

- **Authenticated access for individual NITRO operations:**NITRO allows you to logon to the Citrix ADC appliance to perform individual operations. You can use this option instead of creating a NITRO session (using the login object) and then using that session to perform all operations,

  To do this, you must specify the username and password in the request header of the NITRO request as follows:

  X-NITRO-USER:<username>

  X-NITRO-PASS:<password>

  Note: In such cases, make sure that the request header DOES not include the following:

  Cookie:NITRO_AUTH_TOKEN=<tokenvalue>

`*Note:`*

Mandatory parameters are marked in redand placeholder content is marked in <green>.

add

URL:http://<Citrix-ADC-IP-address(NSIP)/nitro/v1/config/vpnvserver_vpnnexthopserver_binding

HTTP Method:PUT

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Content-Type:application/json

Request Payload:

```
{

"vpnvserver_vpnnexthopserver_binding":{

<b>"name":<String_value>,

</b>"nexthopserver":<String_value>

}}
```

Response:

HTTP Status Code on Success: 201 Created HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error

delete

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/vpnvserver_vpnnexthopserver_binding/name_value<String>

Query-parameters:

args

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/vpnvserver_vpnnexthopserver_binding/name_value<String>?**args=nexthopserver:<String_value>**

HTTP Method:DELETE

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue>

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error

get

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/vpnvserver_vpnnexthopserver_binding/name_value<String>

Query-parameters:

filter

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/vpnvserver_vpnnexthopserver_binding/name_value<String>?**filter=property-name1:property-value1,property-name2:property-value2**

Use this query-parameter to get the filtered set of vpnvserver_vpnnexthopserver_binding resources configured on Citrix ADC. Filtering can be done on any of the properties of the resource.

pagination

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/vpnvserver_vpnnexthopserver_binding/name_value<String>?**pagesize=#no;pageno=#no**

Use this query-parameter to get the vpnvserver_vpnnexthopserver_binding resources in chunks.

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "vpnvserver_vpnnexthopserver_binding": [ {

"name":<String_value>,

"nexthopserver":<String_value>,

"acttype":<Double_value>

}]}
```

get (all)

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/vpnvserver_vpnnexthopserver_binding

Query-parameters:

bulkbindings

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/vpnvserver_vpnnexthopserver_binding?**bulkbindings=yes**

NITRO allows you to fetch bindings in bulk.

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "vpnvserver_vpnnexthopserver_binding": [ {

"name":<String_value>,

"nexthopserver":<String_value>,

"acttype":<Double_value>

}]}
```

count

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/vpnvserver_vpnnexthopserver_binding/name_value<String>?**count=yes**

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{"vpnvserver_vpnnexthopserver_binding": [ { "__count": "#no"} ] }
```

vpnvserver_vpnportaltheme_binding

Binding object showing the vpnportaltheme that can be bound to vpnvserver.

Properties

(click to see Operations)

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
name	<String>	Read-write	Name of the virtual server. Minimum length = 1
portaltheme	<String>	Read-write	Name of the portal theme bound to VPN vserver.
acttype	<Double>	Read-only	.
__count	<Double>	Read-only	count parameter

Operations

(click to see Properties)

ADD:| DELETE:| GET| GET (ALL)| COUNT

Some options that you can use for each operations:

- **Getting warnings in response:**NITRO allows you to get warnings in an operation by specifying the "warning" query parameter as "yes". For example, to get warnings while connecting to the Citrix ADC appliance, the URL is as follows:

  http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/login?warning=yes

  If any, the warnings are displayed in the response payload with the HTTP code "209 X-NITRO-WARNING".

- **Authenticated access for individual NITRO operations:**NITRO allows you to logon to the Citrix ADC appliance to perform individual operations. You can use this option instead of creating a NITRO session (using the login object) and then using that session to perform all operations,

  To do this, you must specify the username and password in the request header of the NITRO request as follows:

  X-NITRO-USER:<username>

X-NITRO-PASS:<password>

Note: In such cases, make sure that the request header DOES not include the following:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue>

`*Note:`*

Mandatory parameters are marked in redand placeholder content is marked in <green>.

add

URL:http://<Citrix-ADC-IP-address(NSIP)/nitro/v1/config/vpnvserver_vpnportaltheme_binding

HTTP Method:PUT

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Content-Type:application/json

Request Payload:

```
{

"vpnvserver_vpnportaltheme_binding":{

<b>"name":<String_value>,

</b>"portaltheme":<String_value>

}}
```

Response:

HTTP Status Code on Success: 201 Created HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error

delete

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/vpnvserver_vpnportaltheme_binding/name_value<String>

Query-parameters:

args

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/vpnvserver_vpnportaltheme_binding/name_value<String>?**args=portaltheme:<String_value>**

HTTP Method:DELETE

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue>

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error

get

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/vpnvserver_vpnportaltheme_binding/name_value<String>

Query-parameters:

filter

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/vpnvserver_vpnportaltheme_binding/name_value<String>?**filter=property-name1:property-value1,property-name2:property-value2**

Use this query-parameter to get the filtered set of vpnvserver_vpnportaltheme_binding resources configured on Citrix ADC. Filtering can be done on any of the properties of the resource.

pagination

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/vpnvserver_vpnportaltheme_binding/name_value<String>?**pagesize=#no;pageno=#no**

Use this query-parameter to get the vpnvserver_vpnportaltheme_binding resources in chunks.

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "vpnvserver_vpnportaltheme_binding": [ {

"name":<String_value>,

"portaltheme":<String_value>,

"acttype":<Double_value>

}]}
```

get (all)

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/vpnvserver_vpnportaltheme_binding

Query-parameters:

bulkbindings

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/vpnvserver_vpnportaltheme_binding?**bulkbindings=yes**

NITRO allows you to fetch bindings in bulk.

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "vpnvserver_vpnportaltheme_binding": [ {

"name":<String_value>,

"portaltheme":<String_value>,

"acttype":<Double_value>

}]}
```

count

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/vpnvserver_vpnportaltheme_binding/name_value<String>?**count=yes**

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{"vpnvserver_vpnportaltheme_binding": [ { "__count": "#no"} ] }
```

vpnvserver_vpnsessionpolicy_binding

Binding object showing the vpnsessionpolicy that can be bound to vpnvserver.

Properties

(click to see Operations)

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
priority	<Double>	Read-write	Integer specifying the policy's priority. The lower the number, the higher the priority. Policies are evaluated in the order of their priority numbers. Maximum value for default syntax policies is 2147483647 and for classic policies is 64000. Minimum value = 0 Maximum value = 2147483647
gotopriorityexpression	<String>	Read-write	Applicable only to advance vpn session policy. Expression or other value specifying the next policy to evaluate if the current policy evaluates to TRUE. Specify one of the following values: *

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
			NEXT - Evaluate the policy with the next higher priority number. * END - End policy evaluation. * An expression that evaluates to a number. If you specify an expression, the number to which it evaluates determines the next policy to evaluate, as follows: * If the expression evaluates to a higher numbered priority, the policy with that priority is evaluated next. * If the expression evaluates to the priority of the current policy, the policy with the next higher numbered priority is evaluated next. * If the expression evaluates to a number that is larger than the largest numbered priority, policy evaluation ends. An UNDEF event is triggered if: * The expression is invalid. * The expression evaluates to a priority number that is numerically lower

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
			than the current policy's priority. * The expression evaluates to a priority number that is between the current policy's priority number (say, 30) and the highest priority number (say, 100), but does not match any configured priority number (for example, the expression evaluates to the number 85). This example assumes that the priority number increments by 10 for every successive policy, and therefore a priority number of 85 does not exist in the policy label.
policy	<String>	Read-write	The name of the policy, if any, bound to the VPN virtual server.
groupextraction	<Boolean>	Read-write	Binds the authentication policy to a tertiary chain which will be used only for group extraction. The user will not authenticate against this server, and this will only be called if primary and/

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
			or secondary authentication has succeeded.
name	<String>	Read-write	Name of the virtual server. Minimum length = 1
secondary	<Boolean>	Read-write	Binds the authentication policy as the secondary policy to use in a two-factor configuration. A user must then authenticate not only via a primary authentication method but also via a secondary authentication method. User groups are aggregated across both. The user name must be exactly the same for both authentication methods, but they can require different passwords.
bindpoint	<String>	Read-write	Bind point to which to bind the policy. Applies only to rewrite and cache policies. If you do not set this parameter, the policy is bound to REQ_DEFAULT or RES_DEFAULT, depending on

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
			whether the policy rule is a response-time or a request-time expression. Possible values = REQUEST, RESPONSE, ICA_REQUEST, OTHERTCP_REQUEST
acttype	<Double>	Read-only	.
__count	<Double>	Read-only	count parameter

Operations

(click to see Properties)

ADD:| DELETE:| GET| GET (ALL)| COUNT

Some options that you can use for each operations:

- **Getting warnings in response:**NITRO allows you to get warnings in an operation by specifying the "warning" query parameter as "yes". For example, to get warnings while connecting to the Citrix ADC appliance, the URL is as follows:

  http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/login?warning=yes

  If any, the warnings are displayed in the response payload with the HTTP code "209 X-NITRO-WARNING".

- **Authenticated access for individual NITRO operations:**NITRO allows you to logon to the Citrix ADC appliance to perform individual operations. You can use this option instead of creating a NITRO session (using the login object) and then using that session to perform all operations,

  To do this, you must specify the username and password in the request header of the NITRO request as follows:

  X-NITRO-USER:<username>

X-NITRO-PASS:<password>

Note: In such cases, make sure that the request header DOES not include the following:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue>

`*Note:`*

Mandatory parameters are marked in redand placeholder content is marked in <green>.

add

URL:http://<Citrix-ADC-IP-address(NSIP)/nitro/v1/config/vpnvserver_vpnsessionpolicy_binding

HTTP Method:PUT

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Content-Type:application/json

Request Payload:

```
{
"vpnvserver_vpnsessionpolicy_binding":{
<b>"name":<String_value>,
</b>"policy":<String_value>,
"priority":<Double_value>,
"secondary":<Boolean_value>,
"groupextraction":<Boolean_value>,
"gotopriorityexpression":<String_value>,
"bindpoint":<String_value>
}}
```

Response:

HTTP Status Code on Success: 201 Created HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error

delete

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/vpnvserver_vpnsessionpolicy_binding/name_value<String>

Query-parameters:

args

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/vpnvserver_vpnsessionpolicy_binding/name_value<String>?**args=policy:<String_value>,secondary:<Boolean_value>,groupextraction:<Boolean_value>,bindpoint:<**

HTTP Method:DELETE

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue>

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error

get

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/vpnvserver_vpnsessionpolicy_binding/name_value<String>

Query-parameters:

filter

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/vpnvserver_vpnsessionpolicy_binding/name_value<String>?**filter=property-name1:property-value1,property-name2:property-value2**

Use this query-parameter to get the filtered set of vpnvserver_vpnsessionpolicy_binding resources configured on Citrix ADC. Filtering can be done on any of the properties of the resource.

pagination

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/vpnvserver_vpnsessionpolicy_binding/name_value<String>?**pagesize=#no;pageno=#no**

Use this query-parameter to get the vpnvserver_vpnsessionpolicy_binding resources in chunks.

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "vpnvserver_vpnsessionpolicy_binding": [ {

"priority":<Double_value>,

"gotopriorityexpression":<String_value>,

"policy":<String_value>,

"groupextraction":<Boolean_value>,

"name":<String_value>,

"secondary":<Boolean_value>,

"bindpoint":<String_value>,

"acttype":<Double_value>

}]}
```

get (all)

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/vpnvserver_vpnsessionpolicy_binding

Query-parameters:

bulkbindings

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/vpnvserver_vpnsessionpolicy_binding?**bulkbindings=yes**

NITRO allows you to fetch bindings in bulk.

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "vpnvserver_vpnsessionpolicy_binding": [ {

"priority":<Double_value>,

"gotopriorityexpression":<String_value>,

"policy":<String_value>,

"groupextraction":<Boolean_value>,

"name":<String_value>,
```

```
"secondary":<Boolean_value>,

"bindpoint":<String_value>,

"acttype":<Double_value>

}]}
```

count

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/vpnvserver_vpnsessionpolicy_binding/name_value<String>?**count=yes**

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{"vpnvserver_vpnsessionpolicy_binding": [ { "__count": "#no"} ] }
```

vpnvserver_vpntrafficpolicy_binding

Binding object showing the vpntrafficpolicy that can be bound to vpnvserver.

Properties

(click to see Operations)

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
priority	<Double>	Read-write	Integer specifying the policy's priority. The lower the number, the higher the priority. Policies are evaluated in the order of their priority numbers. Maximum value for default syntax policies is 2147483647 and for classic policies is 64000. Minimum value = 0 Maximum value = 2147483647
gotopriorityexpression	<String>	Read-write	Applicable only to advance vpn session policy. Expression or other value specifying the next policy to evaluate if the current policy evaluates to TRUE. Specify one of the following values: * NEXT - Evaluate the policy with the next higher priority number. * END - End policy evaluation. * An expression that evaluates to a number. If you specify an expression, the number to which it evaluates determines the next

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
			policy to evaluate, as follows: * If the expression evaluates to a higher numbered priority, the policy with that priority is evaluated next. * If the expression evaluates to the priority of the current policy, the policy with the next higher numbered priority is evaluated next. * If the expression evaluates to a number that is larger than the largest numbered priority, policy evaluation ends. An UNDEF event is triggered if: * The expression is invalid. * The expression evaluates to a priority number that is numerically lower than the current policy's priority. * The expression evaluates to a priority number that is between the current policy's priority number (say, 30) and the highest priority number (say, 100), but does not match any configured priority number (for

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
			example, the expression evaluates to the number 85). This example assumes that the priority number increments by 10 for every successive policy, and therefore a priority number of 85 does not exist in the policy label.
policy	<String>	Read-write	The name of the policy, if any, bound to the VPN virtual server.
groupextraction	<Boolean>	Read-write	Binds the authentication policy to a tertiary chain which will be used only for group extraction. The user will not authenticate against this server, and this will only be called if primary and/or secondary authentication has succeeded.
name	<String>	Read-write	Name of the virtual server. Minimum length = 1
secondary	<Boolean>	Read-write	Binds the authentication policy as the secondary policy to use in a two-factor

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
			configuration. A user must then authenticate not only via a primary authentication method but also via a secondary authentication method. User groups are aggregated across both. The user name must be exactly the same for both authentication methods, but they can require different passwords.
bindpoint	<String>	Read-write	Bind point to which to bind the policy. Applies only to rewrite and cache policies. If you do not set this parameter, the policy is bound to REQ_DEFAULT or RES_DEFAULT, depending on whether the policy rule is a response-time or a request-time expression. Possible values = REQUEST, RESPONSE, ICA_REQUEST, OTHERTCP_REQUEST
acttype	<Double>	Read-only	.

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
__count	<Double>	Read-only	count parameter

Operations

(click to see Properties)

ADD:| DELETE:| GET| GET (ALL)| COUNT

Some options that you can use for each operations:

- **Getting warnings in response:**NITRO allows you to get warnings in an operation by specifying the "warning" query parameter as "yes". For example, to get warnings while connecting to the Citrix ADC appliance, the URL is as follows:

  http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/login?warning=yes

  If any, the warnings are displayed in the response payload with the HTTP code "209 X-NITRO-WARNING".
- **Authenticated access for individual NITRO operations:**NITRO allows you to logon to the Citrix ADC appliance to perform individual operations. You can use this option instead of creating a NITRO session (using the login object) and then using that session to perform all operations,

  To do this, you must specify the username and password in the request header of the NITRO request as follows:

  X-NITRO-USER:<username>

  X-NITRO-PASS:<password>

  Note: In such cases, make sure that the request header DOES not include the following:

  Cookie:NITRO_AUTH_TOKEN=<tokenvalue>

`*Note:`*

Mandatory parameters are marked in redand placeholder content is marked in <green>.

add

URL:http://<Citrix-ADC-IP-address(NSIP)/nitro/v1/config/vpnvserver_vpntrafficpolicy_binding

HTTP Method:PUT

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Content-Type:application/json

Request Payload:

```
{

"vpnvserver_vpntrafficpolicy_binding":{

<b>"name":<String_value>,

</b>"policy":<String_value>,

"priority":<Double_value>,

"secondary":<Boolean_value>,

"groupextraction":<Boolean_value>,

"gotopriorityexpression":<String_value>,

"bindpoint":<String_value>

}}
```

Response:

HTTP Status Code on Success: 201 Created HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error

delete

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/vpnvserver_vpntrafficpolicy_binding/name_value<String>

Query-parameters:

args

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/vpnvserver_vpntrafficpolicy_binding/name_value<String>?**args=policy:<String_value>,secondary:<Boolean_value>,groupextraction:<Boolean_value>,bindpoint:<**

HTTP Method:DELETE

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue>

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error

get

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/vpnvserver_vpntrafficpolicy_binding/name_value<String>

Query-parameters:

filter

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/vpnvserver_vpntrafficpolicy_binding/name_value<String>?**filter=property-name1:property-value1,property-name2:property-value2**

Use this query-parameter to get the filtered set of vpnvserver_vpntrafficpolicy_binding resources configured on Citrix ADC. Filtering can be done on any of the properties of the resource.

pagination

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/vpnvserver_vpntrafficpolicy_binding/name_value<String>?**pagesize=#no;pageno=#no**

Use this query-parameter to get the vpnvserver_vpntrafficpolicy_binding resources in chunks.

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "vpnvserver_vpntrafficpolicy_binding": [ {

"priority":<Double_value>,

"gotopriorityexpression":<String_value>,

"policy":<String_value>,

"groupextraction":<Boolean_value>,

"name":<String_value>,

"secondary":<Boolean_value>,

"bindpoint":<String_value>,

"acttype":<Double_value>

}]}
```

get (all)

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/vpnvserver_vpntrafficpolicy_binding

Query-parameters:

bulkbindings

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/vpnvserver_vpntrafficpolicy_binding?**bulkbindings=yes**

NITRO allows you to fetch bindings in bulk.

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "vpnvserver_vpntrafficpolicy_binding": [ {

"priority":<Double_value>,

"gotopriorityexpression":<String_value>,

"policy":<String_value>,

"groupextraction":<Boolean_value>,

"name":<String_value>,

"secondary":<Boolean_value>,

"bindpoint":<String_value>,

"acttype":<Double_value>

}]}
```

count

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/vpnvserver_vpntrafficpolicy_binding/name_value<String>?**count=yes**

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{"vpnvserver_vpntrafficpolicy_binding": [ { "__count": "#no"} ] }
```

vpnvserver_vpnurl_binding

Binding object showing the vpnurl that can be bound to vpnvserver.

Properties

(click to see Operations)

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
urlname	<String>	Read-write	The intranet URL.
name	<String>	Read-write	Name of the virtual server. Minimum length = 1
acttype	<Double>	Read-only	.
__count	<Double>	Read-only	count parameter

Operations

(click to see Properties)

ADD:| DELETE:| GET| GET (ALL)| COUNT

Some options that you can use for each operations:

- **Getting warnings in response:**NITRO allows you to get warnings in an operation by specifying the "warning" query parameter as "yes". For example, to get warnings while connecting to the Citrix ADC appliance, the URL is as follows:

  http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/login?warning=yes

  If any, the warnings are displayed in the response payload with the HTTP code "209 X-NITRO-WARNING".

- **Authenticated access for individual NITRO operations:**NITRO allows you to logon to the Citrix ADC appliance to perform individual operations. You can use this option instead of creating a NITRO session (using the login object) and then using that session to perform all operations,

  To do this, you must specify the username and password in the request header of the NITRO request as follows:

  X-NITRO-USER:<username>

  X-NITRO-PASS:<password>

  Note: In such cases, make sure that the request header DOES not include the following:

  Cookie:NITRO_AUTH_TOKEN=<tokenvalue>

`*Note:`*

Mandatory parameters are marked in redand placeholder content is marked in <green>.

add

URL:http://<Citrix-ADC-IP-address(NSIP)/nitro/v1/config/vpnvserver_vpnurl_binding

HTTP Method:PUT

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Content-Type:application/json

Request Payload:

```
{

"vpnvserver_vpnurl_binding":{

<b>"name":<String_value>,

</b>"urlname":<String_value>

}}
```

Response:

HTTP Status Code on Success: 201 Created HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error

delete

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/vpnvserver_vpnurl_binding/name_value<String>

Query-parameters:

args

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/vpnvserver_vpnurl_binding/name_value<String>?**args=urlname:<String_value>**

HTTP Method:DELETE

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue>

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error

get

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/vpnvserver_vpnurl_binding/name_value<String>

Query-parameters:

filter

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/vpnvserver_vpnurl_binding/name_value<String>?**filter=property-name1:property-value1,property-name2:property-value2**

Use this query-parameter to get the filtered set of vpnvserver_vpnurl_binding resources configured on Citrix ADC. Filtering can be done on any of the properties of the resource.

pagination

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/vpnvserver_vpnurl_binding/name_value<String>?**pagesize=#no;pageno=#no**

Use this query-parameter to get the vpnvserver_vpnurl_binding resources in chunks.

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "vpnvserver_vpnurl_binding": [ {

"urlname":<String_value>,

"name":<String_value>,
```

```
"acttype":<Double_value>

}]}
```

get (all)

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/vpnvserver_vpnurl_binding

Query-parameters:

bulkbindings

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/vpnvserver_vpnurl_binding?**bulkbindings=yes**

NITRO allows you to fetch bindings in bulk.

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "vpnvserver_vpnurl_binding": [ {

"urlname":<String_value>,

"name":<String_value>,

"acttype":<Double_value>
```

```
}]}
```

count

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/vpnvserver_vpnurl_binding/name_value<String>?**count=yes**

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{"vpnvserver_vpnurl_binding": [ { "__count": "#no"} ] }
```

Stream

Stream

Stream configuration.

NAME	DESCRIPTION
streamidentifier	Configuration for identifier
streamidentifier_binding	Binding object showing the resources that can be bound to streamidentifier

NAME	DESCRIPTION
streamidentifier_streamsession_binding	Binding object showing the streamsession that can be bound to streamidentifier
streamselector	Configuration for selector
streamsession	Configuration for active connection

streamidentifier

Configuration for identifier resource.

Properties

(click to see Operations)

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
name	<String>	Read-write	The name of stream identifier.
selectorname	<String>	Read-write	Name of the selector to use with the stream identifier. Minimum length = 1
interval	<Double>	Read-write	Number of minutes of data to use when calculating session statistics (number of requests, bandwidth, and response times). The interval is a moving window that keeps the most recently collected data. Older data is discarded at regular intervals. Default value: 1 Minimum value = 1

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
samplecount	<Double>	Read-write	Size of the sample from which to select a request for evaluation. The smaller the sample count, the more accurate is the statistical data. To evaluate all requests, set the sample count to 1. However, such a low setting can result in excessive consumption of memory and processing resources. Default value: 1 Minimum value = 1 Maximum value = 65535
sort	<String>	Read-write	Sort stored records by the specified statistics column, in descending order. Performed during data collection, the sorting enables real-time data evaluation through Citrix ADC policies (for example, compression and caching policies) that use functions such as IS_TOP(n).

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
			Default value: REQUESTS Possible values = REQUESTS, CONNECTIONS, RESPTIME, BANDWIDTH, RESPTIME_BREACHES, NONE
snmptrap	<String>	Read-write	Enable/disable SNMP trap for stream identifier. Default value: DISABLED Possible values = ENABLED, DISABLED
appflowlog	<String>	Read-write	Enable/disable Appflow logging for stream identifier. Default value: DISABLED Possible values = ENABLED, DISABLED
trackackonlypackets	<String>	Read-write	Track ack only packets as well. This setting is applicable only when packet rate limiting is being used.

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
			Default value: DISABLED Possible values = ENABLED, DISABLED
tracktransactions	<String>	Read-write	Track transactions exceeding configured threshold. Transaction tracking can be enabled for following metric: ResponseTime. By default transaction tracking is disabled. Default value: NONE Possible values = RESPTIME, NONE
maxtransactionthreshold	<Double>	Read-write	Maximum per transcation value of metric. Metric to be tracked is specified by tracktransactions attribute. Default value: 0
mintransactionthreshold	<Double>	Read-write	Minimum per transcation value of metric. Metric to be tracked is specified by tracktransactions attribute. Default value: 0

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
acceptancethreshold	<String>	Read-write	Non-Breaching transactions to Total transactions threshold expressed in percent. Maximum of 6 decimal places is supported. Default value: 0.000000 Maximum length = 10
breachthreshold	<Double>	Read-write	Breaching transactions threshold calculated over interval. Default value: 0
rule	<String[]>	Read-only	Rule.
__count	<Double>	Read-only	count parameter

Operations

(click to see Properties)

ADD| UPDATE| UNSET| DELETE| GET (ALL)| GET| COUNT

Some options that you can use for each operations:

- **Getting warnings in response:**NITRO allows you to get warnings in an operation by specifying the "warning" query parameter as "yes". For example, to get warnings while connecting to the Citrix ADC appliance, the URL is as follows:

  http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/login?warning=yes

  If any, the warnings are displayed in the response payload with the HTTP code "209 X-NITRO-WARNING".

- **Authenticated access for individual NITRO operations:**NITRO allows you to logon to the Citrix ADC appliance to perform individual operations. You can use this option instead of creating a NITRO session (using the login object) and then using that session to perform all operations,

  To do this, you must specify the username and password in the request header of the NITRO request as follows:

  X-NITRO-USER:<username>

  X-NITRO-PASS:<password>

  Note: In such cases, make sure that the request header DOES not include the following:

  Cookie:NITRO_AUTH_TOKEN=<tokenvalue>

`*Note:`*

Mandatory parameters are marked in redand placeholder content is marked in <green>.

add

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/streamidentifier

HTTP Method:POST

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Content-Type:application/json

Request Payload:

```
{"streamidentifier":{

<b>"name":<String_value>,

</b><b>"selectorname":<String_value>,

</b>"interval":<Double_value>,

"samplecount":<Double_value>,

"sort":<String_value>,
```

```
"snmptrap":<String_value>,

"appflowlog":<String_value>,

"trackackonlypackets":<String_value>,

"tracktransactions":<String_value>,

"maxtransactionthreshold":<Double_value>,

"mintransactionthreshold":<Double_value>,

"acceptancethreshold":<String_value>,

"breachthreshold":<Double_value>

}}
```

Response:

HTTP Status Code on Success: 201 Created HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error

update

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/streamidentifier

HTTP Method:PUT

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Content-Type:application/json

Request Payload:

```
{"streamidentifier":{

<b>"name":<String_value>,
```

```
</b>"selectorname":<String_value>,

"interval":<Double_value>,

"samplecount":<Double_value>,

"sort":<String_value>,

"snmptrap":<String_value>,

"appflowlog":<String_value>,

"trackackonlypackets":<String_value>,

"tracktransactions":<String_value>,

"maxtransactionthreshold":<Double_value>,

"mintransactionthreshold":<Double_value>,

"acceptancethreshold":<String_value>,

"breachthreshold":<Double_value>

}}
```

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error

unset

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/streamidentifier?**action=unset**

HTTP Method:POST

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Content-Type:application/json

Request Payload:

```
{"streamidentifier":{

<b>"name":<String_value>,

</b>"selectorname":true,

"interval":true,

"samplecount":true,

"sort":true,

"snmptrap":true,

"appflowlog":true,

"trackackonlypackets":true,

"tracktransactions":true,

"maxtransactionthreshold":true,

"mintransactionthreshold":true,

"acceptancethreshold":true,

"breachthreshold":true

}}
```

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error

delete

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/streamidentifier/name_value<String>

HTTP Method:DELETE

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue>

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error

get (all)

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/streamidentifier

Query-parameters:

attrs

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/streamidentifier?**attrs=property-name1,property-name2**

Use this query parameter to specify the resource details that you want to retrieve.

filter

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/streamidentifier?**filter=property-name1:property-val1,property-name2:property-val2**

Use this query-parameter to get the filtered set of streamidentifier resources configured on Citrix ADC. Filtering can be done on any of the properties of the resource.

view

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/streamidentifier?**view=summary**

Note: By default, the retrieved results are displayed in detail view (?view=detail).

pagination

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/streamidentifier?**pagesize=#no;pageno=#no**

Use this query-parameter to get the streamidentifier resources in chunks.

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "streamidentifier": [ {

"name":<String_value>,

"selectorname":<String_value>,

"rule":<String[]_value>,

"interval":<Double_value>,

"samplecount":<Double_value>,

"sort":<String_value>,

"snmptrap":<String_value>,

"appflowlog":<String_value>,

"trackackonlypackets":<String_value>,

"tracktransactions":<String_value>,

"maxtransactionthreshold":<Double_value>,
```

```
"mintransactionthreshold":<Double_value>,

"acceptancethreshold":<String_value>,

"breachthreshold":<Double_value>

}]}
```

get

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/streamidentifier/name_value<String>

Query-parameters:

attrs

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/streamidentifier/name_value<String>?**attrs=property-name1,property-name2**

Use this query parameter to specify the resource details that you want to retrieve.

view

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/streamidentifier/name_value<String>?**view=summary**

Note: By default, the retrieved results are displayed in detail view (?view=detail).

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "streamidentifier": [ {

"name":<String_value>,

"selectorname":<String_value>,

"rule":<String[]_value>,

"interval":<Double_value>,

"samplecount":<Double_value>,

"sort":<String_value>,

"snmptrap":<String_value>,

"appflowlog":<String_value>,

"trackackonlypackets":<String_value>,

"tracktransactions":<String_value>,

"maxtransactionthreshold":<Double_value>,

"mintransactionthreshold":<Double_value>,

"acceptancethreshold":<String_value>,

"breachthreshold":<Double_value>

}]}
```

count

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/streamidentifier?**count=yes**

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "streamidentifier": [ { "__count": "#no"} ] }
```

streamidentifier_binding

Binding object which returns the resources bound to streamidentifier.

Properties

(click to see Operations)

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
name	<String>	Read-write	The name of stream identifier.
streamidentifier_streamsession_binding	<streamidentifier_streamsession_binding[]>	Read-only	streamsession that can be bound to streamidentifier.

Operations

(click to see Properties)

GET| GET (ALL)

Some options that you can use for each operations:

- **Getting warnings in response:**NITRO allows you to get warnings in an operation by specifying the "warning" query parameter as "yes". For example, to get warnings while connecting to the Citrix ADC appliance, the URL is as follows:

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/login?warning=yes

If any, the warnings are displayed in the response payload with the HTTP code "209 X-NITRO-WARNING".

- **Authenticated access for individual NITRO operations:**NITRO allows you to logon to the Citrix ADC appliance to perform individual operations. You can use this option instead of creating a NITRO session (using the login object) and then using that session to perform all operations,

  To do this, you must specify the username and password in the request header of the NITRO request as follows:

  X-NITRO-USER:<username>

  X-NITRO-PASS:<password>

  Note: In such cases, make sure that the request header DOES not include the following:

  Cookie:NITRO_AUTH_TOKEN=<tokenvalue>

`*Note:`*

Mandatory parameters are marked in redand placeholder content is marked in <green>.

get

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/streamidentifier_binding/name_value<String>

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "streamidentifier_binding": [ {

"name":<String_value>,

"streamidentifier_streamsession_binding":<streamidentifier_streamsession_b
inding[]_value>

}]}
```

get (all)

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/streamidentifier_binding

Query-parameters:

bulkbindings

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/streamidentifier_binding?**bulkbindings=yes**

NITRO allows you to fetch bindings in bulk.

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "streamidentifier_binding": [ {
```

```
"name":<String_value>,

"streamidentifier_streamsession_binding":<streamidentifier_streamsession_b
inding[]_value>

}]}
```

streamidentifier_streamsession_binding

Binding object showing the streamsession that can be bound to streamidentifier.

Properties

(click to see Operations)

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
name	<String>	Read-write	The name of stream identifier.
__count	<Double>	Read-write	count parameter

Operations

(click to see Properties)

GET| GET (ALL)| COUNT

Some options that you can use for each operations:

- **Getting warnings in response:**NITRO allows you to get warnings in an operation by specifying the "warning" query parameter as "yes". For example, to get warnings while connecting to the Citrix ADC appliance, the URL is as follows:

  http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/login?warning=yes

  If any, the warnings are displayed in the response payload with the HTTP code "209 X-NITRO-WARNING".

- **Authenticated access for individual NITRO operations:**NITRO allows you to logon to the Citrix ADC appliance to perform individual operations. You can use this option instead of

creating a NITRO session (using the login object) and then using that session to perform all operations,

To do this, you must specify the username and password in the request header of the NITRO request as follows:

X-NITRO-USER:<username>

X-NITRO-PASS:<password>

Note: In such cases, make sure that the request header DOES not include the following:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue>

`*Note:`*

Mandatory parameters are marked in redand placeholder content is marked in <green>.

get

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/streamidentifier_streamsession_binding

Query-parameters:

filter

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/streamidentifier_streamsession_binding?**filter=property-name1:property-value1,property-name2:property-value2**

Use this query-parameter to get the filtered set of streamidentifier_streamsession_binding resources configured on Citrix ADC. Filtering can be done on any of the properties of the resource.

pagination

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/streamidentifier_streamsession_binding?**pagesize=#no;pageno=#no**

Use this query-parameter to get the streamidentifier_streamsession_binding resources in chunks.

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "streamidentifier_streamsession_binding": [ {

"name":<String_value>

}]}
```

get (all)

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/streamidentifier_streamsession_binding

Query-parameters:

bulkbindings

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/streamidentifier_streamsession_binding?**bulkbindings=yes**

NITRO allows you to fetch bindings in bulk.

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "streamidentifier_streamsession_binding": [ {

"name":<String_value>

}]}
```

count

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/streamidentifier_streamsession_binding?**args=;count=yes**

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "streamidentifier_streamsession_binding": [ { "__count": "#no"} ] }
```

streamselector

Configuration for selector resource.

Properties

(click to see Operations)

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
name	<String>	Read-write	Name for the selector. Must begin with an ASCII alphanumeric or underscore (_) character, and must contain only ASCII alphanumeric, underscore, hash (#), period (.), space, colon (:), at (@), equals (=), and hyphen (-) characters. If the name includes one or more spaces, and you are using the Citrix ADC CLI, enclose the name in double or single quotation marks (for example, "my selector" or 'my selector').
rule	<String[]>	Read-write	Set of up to five expressions. Maximum length: 7499 characters. Each expression must identify a specific request characteristic, such as the client's IP address (with CLIENT.IP.SRC) or requested server resource (with HTTP.REQ.URL).

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
			Note: If two or more selectors contain the same expressions in different order, a separate set of records is created for each selector. Minimum length = 1
__count	<Double>	Read-only	count parameter

Operations

(click to see Properties)

ADD| UPDATE| DELETE| GET (ALL)| GET| COUNT

Some options that you can use for each operations:

- **Getting warnings in response:**NITRO allows you to get warnings in an operation by specifying the "warning" query parameter as "yes". For example, to get warnings while connecting to the Citrix ADC appliance, the URL is as follows:

  http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/login?warning=yes

  If any, the warnings are displayed in the response payload with the HTTP code "209 X-NITRO-WARNING".

- **Authenticated access for individual NITRO operations:**NITRO allows you to logon to the Citrix ADC appliance to perform individual operations. You can use this option instead of creating a NITRO session (using the login object) and then using that session to perform all operations,

  To do this, you must specify the username and password in the request header of the NITRO request as follows:

  X-NITRO-USER:<username>

  X-NITRO-PASS:<password>

Note: In such cases, make sure that the request header DOES not include the following:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue>

`*Note:` *

Mandatory parameters are marked in redand placeholder content is marked in <green>.

add

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/streamselector

HTTP Method:POST

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Content-Type:application/json

Request Payload:

```
{"streamselector":{

<b>"name":<String_value>,

</b><b>"rule":<String[]_value>

</b>}}
```

Response:

HTTP Status Code on Success: 201 Created HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error

update

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/streamselector

HTTP Method:PUT

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Content-Type:application/json

Request Payload:

```
{"streamselector":{

<b>"name":<String_value>,

</b>"rule":<String[]_value>

}}
```

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error

delete

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/streamselector/name_value<String>

HTTP Method:DELETE

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue>

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error

get (all)

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/streamselector

Query-parameters:

attrs

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/streamselector?**attrs=property-name1,property-name2**

Use this query parameter to specify the resource details that you want to retrieve.

filter

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/streamselector?**filter=property-name1:property-val1,property-name2:property-val2**

Use this query-parameter to get the filtered set of streamselector resources configured on Citrix ADC. Filtering can be done on any of the properties of the resource.

view

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/streamselector?**view=summary**

Note: By default, the retrieved results are displayed in detail view (?view=detail).

pagination

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/streamselector?**pagesize=#no;pageno=#no**

Use this query-parameter to get the streamselector resources in chunks.

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "streamselector": [ {

"name":<String_value>,

"rule":<String[]_value>
```

```
}]}
```

get

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/streamselector/name_value<String>

Query-parameters:

attrs

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/streamselector/name_value<String>?**attrs=property-name1,property-name2**

Use this query parameter to specify the resource details that you want to retrieve.

view

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/streamselector/name_value<String>?**view=summary**

Note: By default, the retrieved results are displayed in detail view (?view=detail).

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "streamselector": [ {

"name":<String_value>,
```

```
"rule":<String[]_value>

}]}
```

count

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/streamselector?**count=yes**

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "streamselector": [ { "__count": "#no"} ] }
```

streamsession

Configuration for active connection resource.

Properties

(click to see Operations)

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
name	<String>	Read-write	Name of the stream identifier. Minimum length = 1

Operations

(click to see Properties)

CLEAR

Some options that you can use for each operations:

- **Getting warnings in response:**NITRO allows you to get warnings in an operation by specifying the "warning" query parameter as "yes". For example, to get warnings while connecting to the Citrix ADC appliance, the URL is as follows:

  http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/login?warning=yes

  If any, the warnings are displayed in the response payload with the HTTP code "209 X-NITRO-WARNING".

- **Authenticated access for individual NITRO operations:**NITRO allows you to logon to the Citrix ADC appliance to perform individual operations. You can use this option instead of creating a NITRO session (using the login object) and then using that session to perform all operations,

  To do this, you must specify the username and password in the request header of the NITRO request as follows:

  X-NITRO-USER:<username>

  X-NITRO-PASS:<password>

  Note: In such cases, make sure that the request header DOES not include the following:

  Cookie:NITRO_AUTH_TOKEN=<tokenvalue>

`*Note:`*

Mandatory parameters are marked in redand placeholder content is marked in <green>.

clear

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/streamsession?**action=clear**

HTTP Method:POST

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Content-Type:application/json

Request Payload:

```
{"streamsession":{

<b>"name":<String_value>

</b>}}
```

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error

Subscriber

Subscriber

Subscriber configuration

NAME	DESCRIPTION
subscribergxinterface	Configuration for Gx interface Parameters
subscriberparam	Configuration for Subscriber Params
subscriberprofile	Configuration for Subscriber Profile
subscriberradiusinterface	Configuration for RADIUS interface Parameters
subscribersessions	Configuration for subscriber sesions

subscribergxinterface

Configuration for Gx interface Parameters resource.

Properties

(click to see Operations)

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
vserver	<String>	Read-write	Name of the load balancing, or content switching vserver to which the Gx connections are established. The service type of the virtual server must be DIAMETER/ SSL_DIAMETER. Mutually exclusive with the service parameter. Therefore, you cannot set both service and the Virtual Server in the Gx Interface. Minimum length = 1
service	<String>	Read-write	Name of DIAMETER/ SSL_DIAMETER service corresponding to PCRF to which the Gx connection is established. The service type of the service must be DIAMETER/ SSL_DIAMETER. Mutually exclusive with vserver parameter. Therefore, you cannot set both Service and the Virtual Server in the Gx Interface.

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
			Minimum length = 1
pcrfrealm	<String>	Read-write	PCRF realm is of type DiameterIdentity and contains the realm of PCRF to which the message is to be routed. This is the realm used in Destination-Realm AVP by Citrix ADC Gx client (as a Diameter node). Minimum length = 1
holdonsubscriberabsence	<String>	Read-write	Set this setting to yes if Citrix ADC needs to Hold pakcets till subscriber session is fetched from PCRF. Else set to NO. By default set to yes. If this setting is set to NO, then till Citrix ADC fetches subscriber from PCRF, default subscriber profile will be applied to this subscriber if configured. If default subscriber profile is also not configured an undef would be raised to expressions which use Subscriber attributes. . Default value: YES

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
			Possible values = YES, NO
requesttimeout	<Double>	Read-write	q!Time, in seconds, within which the Gx CCR request must complete. If the request does not complete within this time, the request is retransmitted for requestRetryAttempts time. If still reuqest is not complete then default subscriber profile will be applied to this subscriber if configured. If default subscriber profile is also not configured an undef would be raised to expressions which use Subscriber attributes. Zero disables the timeout. !. Default value: 10 Minimum value = 0 Maximum value = 86400
requestretryattempts	<Double>	Read-write	If the request does not complete within requestTimeout time, the request is retransmitted for requestRetryAttempts time.

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
			Default value: 3
idlettl	<Double>	Read-write	q!Idle Time, in seconds, after which the Gx CCR-U request will be sent after any PCRF activity on a session. Any RAR or CCA message resets the timer. Zero value disables the idle timeout. !. Default value: 900 Minimum value = 0 Maximum value = 86400
revalidationtimeout	<Double>	Read-write	q!Revalidation Timeout, in seconds, after which the Gx CCR-U request will be sent after any PCRF activity on a session. Any RAR or CCA message resets the timer. Zero value disables the idle timeout. !. Default value: 0 Minimum value = 0 Maximum value = 86400

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
healthcheck	<String>	Read-write	q!Set this setting to yes if Citrix ADC should send DWR packets to PCRF server. When the session is idle, healthcheck timer expires and DWR packets are initiated in order to check that PCRF server is active. By default set to No. !. Default value: NO Possible values = YES, NO
healthcheckttl	<Double>	Read-write	q!Healthcheck timeout, in seconds, after which the DWR will be sent in order to ensure the state of the PCRF server. Any CCR, CCA, RAR or RRA message resets the timer. !. Default value: 30 Minimum value = 6 Maximum value = 86400
cerrequesttimeout	<Double>	Read-write	q!Healthcheck request timeout, in seconds, after which the Citrix ADC considers that no CCA packet received

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
			to the initiated CCR. After this time Citrix ADC should send again CCR to PCRF server. !. Default value: 0 Minimum value = 0 Maximum value = 86400
negativettl	<Double>	Read-write	q!Negative TTL, in seconds, after which the Gx CCR-I request will be resent for sessions that have not been resolved by PCRF due to server being down or no response or failed response. Instead of polling the PCRF server constantly, negative-TTL makes Citrix ADC stick to un-resolved session. Meanwhile Citrix ADC installs a negative session to avoid going to PCRF. For Negative Sessions, Netcaler inherits the attributes from default subscriber profile if default subscriber is configured. A

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
			default subscriber could be configured as 'add subscriber profile *'. Or these attributes can be inherited from Radius as well if Radius is configued. Zero value disables the Negative Sessions. And Citrix ADC does not install Negative sessions even if subscriber session could not be fetched. !. Default value: 600 Minimum value = 0 Maximum value = 86400
negativettllimitedsuccess	<String>	Read-write	Set this to YES if Citrix ADC should create negative session for Result-Code DIAMETER_LIMITED_SUCCESS (2002) received in CCA-I. If set to NO, regular session is created. Default value: NO

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
			Possible values = YES, NO
purgesdbongxfailure	<String>	Read-write	Set this setting to YES if needed to purge Subscriber Database in case of Gx failure. By default set to NO. . Default value: NO Possible values = YES, NO
servicepathavp	<Double[]>	Read-write	The AVP code in which PCRF sends service path applicable for subscriber. Minimum value = 1
servicepathvendorid	<Double>	Read-write	The vendorid of the AVP in which PCRF sends service path for subscriber.
nodeid	<Double>	Read-write	Unique number that identifies the cluster node. Minimum value = 0 Maximum value = 31
svrstate	<String>	Read-only	The state of the gx service. Possible values = UP, DOWN, UNKNOWN, BUSY, OUT OF SERVICE,

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
			GOING OUT OF SERVICE, DOWN WHEN GOING OUT OF SERVICE, NS_EMPTY_STR, Unknown, DISABLED
identity	<String>	Read-only	DiameterIdentity to be used by NS. DiameterIdentity is used to identify a Diameter node uniquely. Before setting up diameter configuration, Citrix ADC (as a Diameter node) MUST be assigned a unique DiameterIdentity. example => set ns diameter -identity Citrix ADC.com Now whenever Citrix ADC system needs to use identity in diameter messages. It will use 'netscaler.com' as Origin-Host AVP as defined in RFC3588 . Minimum length = 1

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
realm	<String>	Read-only	Diameter Realm to be used by NS. example => set ns diameter -realm com Now whenever Citrix ADC system needs to use realm in diameter messages. It will use 'com' as Origin-Realm AVP as defined in RFC3588 . Minimum length = 1
status	<String>	Read-only	Citrix ADC PCRF connection Status. (Gx Protocol State).
servicepathinfomode	<String>	Read-only	The type of info in which service path is passed from PCRF, SERVICE_FUNCTION: Denotes the service chain is passed as servicefunction names in-order in AVPS with code servicepathAVP SERVICE_PATH: Denotes the service path name is

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
			passed in AVPS with code servicepathAVP.q. Default value: SERVICEPATH Possible values = SERVICEFUNCTIONS, SERVICEPATH
gxreportingavp1	<Double[]>	Read-only	The AVP code to report as 1st custom AVP. Minimum value = 1
gxreportingavp1vendorid	<Double>	Read-only	The vendorid of the AVP which will be reported as 1st custom AVP.
gxreportingavp1type	<String>	Read-only	The type of the AVP which will be reported as 1st custom AVP. Default value: OctetString Possible values = OctetString, Integer32, Integer64, Unsigned32, Unsigned64, Float32, Float64
gxreportingavp2	<Double[]>	Read-only	The AVP code to report as 2nd custom AVP.

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
			Minimum value = 1
gxreportingavp2vendorid	<Double>	Read-only	The vendorid of the AVP which will be reported as 2nd custom AVP.
gxreportingavp2type	<String>	Read-only	The type of the AVP which will be reported as 2nd custom AVP. Default value: OctetString Possible values = OctetString, Integer32, Integer64, Unsigned32, Unsigned64, Float32, Float64
gxreportingavp3	<Double[]>	Read-only	The AVP code to report as 3rd custom AVP. Minimum value = 1
gxreportingavp3vendorid	<Double>	Read-only	The vendorid of the AVP which will be reported as 3rd custom AVP.
gxreportingavp3type	<String>	Read-only	The type of the AVP which will be reported as 3rd custom AVP. Default value: OctetString

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
			Possible values = OctetString, Integer32, Integer64, Unsigned32, Unsigned64, Float32, Float64
gxreportingavp4	<Double[]>	Read-only	The AVP code to report as 4th custom AVP. Minimum value = 1
gxreportingavp4vendorid	<Double>	Read-only	The vendorid of the AVP which will be reported as 4th custom AVP.
gxreportingavp4type	<String>	Read-only	The type of the AVP which will be reported as 4th custom AVP. Default value: OctetString Possible values = OctetString, Integer32, Integer64, Unsigned32, Unsigned64, Float32, Float64
gxreportingavp5	<Double[]>	Read-only	The AVP code to report as 5th custom AVP. Minimum value = 1

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
gxreportingavp5vendorid	<Double>	Read-only	The vendorid of the AVP which will be reported as 5th custom AVP.
gxreportingavp5type	<String>	Read-only	The type of the AVP which will be reported as 5th custom AVP. Default value: OctetString Possible values = OctetString, Integer32, Integer64, Unsigned32, Unsigned64, Float32, Float64

Operations

(click to see Properties)

UPDATE| UNSET| GET (ALL)

Some options that you can use for each operations:

- **Getting warnings in response:**NITRO allows you to get warnings in an operation by specifying the "warning" query parameter as "yes". For example, to get warnings while connecting to the Citrix ADC appliance, the URL is as follows:

  http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/login?warning=yes

  If any, the warnings are displayed in the response payload with the HTTP code "209 X-NITRO-WARNING".

- **Authenticated access for individual NITRO operations:**NITRO allows you to logon to the Citrix ADC appliance to perform individual operations. You can use this option instead of

creating a NITRO session (using the login object) and then using that session to perform all operations,

To do this, you must specify the username and password in the request header of the NITRO request as follows:

X-NITRO-USER:<username>

X-NITRO-PASS:<password>

Note: In such cases, make sure that the request header DOES not include the following:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue>

`*Note:`*

Mandatory parameters are marked in redand placeholder content is marked in <green>.

update

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/subscribergxinterface

HTTP Method:PUT

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Content-Type:application/json

Request Payload:

```
{"subscribergxinterface":{

"vserver":<String_value>,

"service":<String_value>,

"pcrfrealm":<String_value>,

"holdonsubscriberabsence":<String_value>,

"requesttimeout":<Double_value>,

"requestretryattempts":<Double_value>,
```

```
"idlettl":<Double_value>,

"revalidationtimeout":<Double_value>,

"healthcheck":<String_value>,

"healthchecktttl":<Double_value>,

"cerrequesttimeout":<Double_value>,

"negativettl":<Double_value>,

"negativettllimitedsuccess":<String_value>,

"purgesdbongxfailure":<String_value>,

"servicepathavp":<Double[]_value>,

"servicepathvendorid":<Double_value>

}}
```

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error

unset

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/subscribergxinterface?**action=unset**

HTTP Method:POST

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Content-Type:application/json

Request Payload:

```
{"subscribergxinterface":{
"vserver":true,
"service":true,
"holdonsubscriberabsence":true,
"requesttimeout":true,
"requestretryattempts":true,
"idlettl":true,
"revalidationtimeout":true,
"healthcheck":true,
"healthchecktttl":true,
"cerrequesttimeout":true,
"negativettl":true,
"negativettllimitedsuccess":true,
"purgesdbongxfailure":true,
"servicepathavp":true,
"servicepathvendorid":true
}}
```


Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error

get (all)

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/subscribergxinterface

Query-parameters:

args

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/subscribergxinterface?**args=nodeid:<Double_value>**

Use this query-parameter to get subscribergxinterface resources based on additional properties.

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "subscribergxinterface": [ {

nodeid:<Double_value>"vserver":<String_value>,

"service":<String_value>,

"svrstate":<String_value>,

"pcrfrealm":<String_value>,

"holdonsubscriberabsence":<String_value>,

"requesttimeout":<Double_value>,
```

```
"requestretryattempts":<Double_value>,

"idlettl":<Double_value>,

"revalidationtimeout":<Double_value>,

"healthcheck":<String_value>,

"healthchecktttl":<Double_value>,

"cerrequesttimeout":<Double_value>,

"negativettl":<Double_value>,

"negativettllimitedsuccess":<String_value>,

"purgesdbongxfailure":<String_value>,

"identity":<String_value>,

"realm":<String_value>,

"status":<String_value>,

"servicepathavp":<Double[]_value>,

"servicepathvendorid":<Double_value>,

"servicepathinfomode":<String_value>,

"gxreportingavp1":<Double[]_value>,

"gxreportingavp1vendorid":<Double_value>,

"gxreportingavp1type":<String_value>,

"gxreportingavp2":<Double[]_value>,

"gxreportingavp2vendorid":<Double_value>,

"gxreportingavp2type":<String_value>,

"gxreportingavp3":<Double[]_value>,
```

```
"gxreportingavp3vendorid":<Double_value>,

"gxreportingavp3type":<String_value>,

"gxreportingavp4":<Double[]_value>,

"gxreportingavp4vendorid":<Double_value>,

"gxreportingavp4type":<String_value>,

"gxreportingavp5":<Double[]_value>,

"gxreportingavp5vendorid":<Double_value>,

"gxreportingavp5type":<String_value>

}]}
```

subscriberparam

Configuration for Subscriber Params resource.

Properties

(click to see Operations)

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
keytype	<String>	Read-write	Type of subscriber key type IP or IPANDVLAN. IPANDVLAN option can be used only when the interfaceType is set to gxOnly. Changing the lookup method should result to the subscriber session

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
			database being flushed. Default value: IP Possible values = IP, IPANDVLAN
interfacetype	<String>	Read-write	Subscriber Interface refers to Citrix ADC interaction with control plane protocols, RADIUS and GX. Types of subscriber interface: NONE, RadiusOnly, RadiusAndGx, GxOnly. NONE: Only static subscribers can be configured. RadiusOnly: GX interface is absent. Subscriber information is obtained through RADIUS Accounting messages. RadiusAndGx: Subscriber ID obtained through RADIUS Accounting is used to query PCRF. Subscriber information is

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
			obtained from both RADIUS and PCRF. GxOnly: RADIUS interface is absent. Subscriber information is queried using Subscriber IP or IP+VLAN. Default value: None Possible values = None, RadiusOnly, RadiusAndGx, GxOnly
idlettl	<Double>	Read-write	q!Idle Timeout, in seconds, after which Citrix ADC will take an idleAction on a subscriber session (refer to 'idleAction' arguement in 'set subscriber param' for more details on idleAction). Any data-plane or control plane activity updates the idleTimeout on subscriber session. idleAction could be to 'just delete the session' or 'delete and CCR-T' (if PCRF is configured) or 'do not delete but send a CCR-U'.

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
			Zero value disables the idle timeout. !. Default value: 0 Minimum value = 0 Maximum value = 172800
idleaction	<String>	Read-write	q!Once idleTTL exprires on a subscriber session, Citrix ADC will take an idle action on that session. idleAction could be chosen from one of these ==> 1. ccrTerminate: (default) send CCR-T to inform PCRF about session termination and delete the session. 2. delete: Just delete the subscriber session without informing PCRF. 3. ccrUpdate: Do not delete the session and instead send a CCR-U to PCRF requesting

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
			for an updated session. !. Default value: ccrTerminate Possible values = ccrTerminate, delete, ccrUpdate
ipv6prefixlookuplist	<Double[]>	Read-write	The ipv6PrefixLookupList should consist of all the ipv6 prefix lengths assigned to the UE's'. Minimum value = 1 Maximum value = 128

Operations

(click to see Properties)

UPDATE| UNSET| GET (ALL)

Some options that you can use for each operations:

- **Getting warnings in response:**NITRO allows you to get warnings in an operation by specifying the "warning" query parameter as "yes". For example, to get warnings while connecting to the Citrix ADC appliance, the URL is as follows:

  http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/login?warning=yes

  If any, the warnings are displayed in the response payload with the HTTP code "209 X-NITRO-WARNING".

- **Authenticated access for individual NITRO operations:**NITRO allows you to logon to the Citrix ADC appliance to perform individual operations. You can use this option instead of

creating a NITRO session (using the login object) and then using that session to perform all operations,

To do this, you must specify the username and password in the request header of the NITRO request as follows:

X-NITRO-USER:<username>

X-NITRO-PASS:<password>

Note: In such cases, make sure that the request header DOES not include the following:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue>

`*Note:`*

Mandatory parameters are marked in redand placeholder content is marked in <green>.

update

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/subscriberparam

HTTP Method:PUT

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Content-Type:application/json

Request Payload:

```
{"subscriberparam":{

"keytype":<String_value>,

"interfacetype":<String_value>,

"idlettl":<Double_value>,

"idleaction":<String_value>,

"ipv6prefixlookuplist":<Double[]_value>

}}
```

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error

unset

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/subscriberparam?**action=unset**

HTTP Method:POST

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Content-Type:application/json

Request Payload:

```
{"subscriberparam":{

"keytype":true,

"interfacetype":true,

"idlettl":true,

"idleaction":true

}}
```

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error

get (all)

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/subscriberparam

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "subscriberparam": [ {

"keytype":<String_value>,

"interfacetype":<String_value>,

"idlettl":<Double_value>,

"idleaction":<String_value>,

"ipv6prefixlookuplist":<Double[]_value>

}]}
```

subscriberprofile

Configuration for Subscriber Profile resource.

Properties

(click to see Operations)

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
ip	<String>	Read-write	Subscriber ip address.
vlan	<Double>	Read-write	The vlan number on which the subscriber is located. Default value: 0 Minimum value = 0 Maximum value = 4096
subscriberrules	<String[]>	Read-write	Rules configured for this subscriber. This is similar to rules received from PCRF for dynamic subscriber sessions.
subscriptionidtype	<String>	Read-write	Subscription-Id type. Possible values = E164, IMSI, SIP_URI, NAI, PRIVATE
subscriptionidvalue	<String>	Read-write	Subscription-Id value.
servicepath	<String>	Read-write	Name of the servicepath to be taken for this subscriber.
flags	<Double>	Read-only	Subscriber Session flags.
ttl	<Double>	Read-only	Subscriber Session TTL.

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
avpdisplaybuffer	<String>	Read-only	Subscriber Attributes Display.
__count	<Double>	Read-only	count parameter

Operations

(click to see Properties)

ADD| UPDATE| UNSET| DELETE| GET (ALL)| COUNT

Some options that you can use for each operations:

- **Getting warnings in response:**NITRO allows you to get warnings in an operation by specifying the "warning" query parameter as "yes". For example, to get warnings while connecting to the Citrix ADC appliance, the URL is as follows:

  http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/login?warning=yes

  If any, the warnings are displayed in the response payload with the HTTP code "209 X-NITRO-WARNING".

- **Authenticated access for individual NITRO operations:**NITRO allows you to logon to the Citrix ADC appliance to perform individual operations. You can use this option instead of creating a NITRO session (using the login object) and then using that session to perform all operations,

  To do this, you must specify the username and password in the request header of the NITRO request as follows:

  X-NITRO-USER:<username>

  X-NITRO-PASS:<password>

  Note: In such cases, make sure that the request header DOES not include the following:

  Cookie:NITRO_AUTH_TOKEN=<tokenvalue>

`*Note:`*

Mandatory parameters are marked in redand placeholder content is marked in <green>.

add

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/subscriberprofile

HTTP Method:POST

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Content-Type:application/json

Request Payload:

```
{"subscriberprofile":{

<b>"ip":<String_value>,

</b>"vlan":<Double_value>,

"subscriberrules":<String[]_value>,

"subscriptionidtype":<String_value>,

"subscriptionidvalue":<String_value>,

"servicepath":<String_value>

}}
```

Response:

HTTP Status Code on Success: 201 Created HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error

update

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/subscriberprofile

HTTP Method:PUT

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Content-Type:application/json

Request Payload:

```
{"subscriberprofile":{

<b>"ip":<String_value>,

</b>"vlan":<Double_value>,

"subscriberrules":<String[]_value>,

"subscriptionidtype":<String_value>,

"subscriptionidvalue":<String_value>,

"servicepath":<String_value>

}}
```

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error

unset

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/subscriberprofile?**action=unset**

HTTP Method:POST

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Content-Type:application/json

Request Payload:

```
{"subscriberprofile":{
```

```
<b>"ip":<String_value>,

</b>"servicepath":true

}}
```

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error

delete

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/subscriberprofile/ip_value<String>

HTTP Method:DELETE

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue>

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error

get (all)

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/subscriberprofile

Query-parameters:

attrs

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/subscriberprofile?**attrs=property-name1,property-name2**

Use this query parameter to specify the resource details that you want to retrieve.

filter

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/subscriberprofile?**filter=property-name1:property-val1,property-name2:property-val2**

Use this query-parameter to get the filtered set of subscriberprofile resources configured on Citrix ADC. Filtering can be done on any of the properties of the resource.

view

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/subscriberprofile?**view=summary**

Note: By default, the retrieved results are displayed in detail view (?view=detail).

pagination

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/subscriberprofile?**pagesize=#no;pageno=#no**

Use this query-parameter to get the subscriberprofile resources in chunks.

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "subscriberprofile": [ {

"ip":<String_value>,

"vlan":<Double_value>,

"subscriptionidtype":<String_value>,
```

```
"subscriptionidvalue":<String_value>,

"subscriberrules":<String[]_value>,

"flags":<Double_value>,

"ttl":<Double_value>,

"avpdisplaybuffer":<String_value>,

"servicepath":<String_value>

}]}
```

count

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/subscriberprofile?**count=yes**

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "subscriberprofile": [ { "__count": "#no"} ] }
```

subscriberradiusinterface

Configuration for RADIUS interface Parameters resource.

Properties

(click to see Operations)

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
listeningservice	<String>	Read-write	Name of RADIUS LISTENING service that will process RADIUS accounting requests. Minimum length = 1
radiusinterimasstart	<String>	Read-write	Treat radius interim message as start radius messages. Default value: DISABLED Possible values = ENABLED, DISABLED
svrstate	<String>	Read-only	The state of the radius service. Possible values = UP, DOWN, UNKNOWN, BUSY, OUT OF SERVICE, GOING OUT OF SERVICE, DOWN WHEN GOING OUT OF SERVICE, NS_EMPTY_STR, Unknown, DISABLED

Operations

(click to see Properties)

UPDATE| UNSET| GET (ALL)

Some options that you can use for each operations:

- **Getting warnings in response:**NITRO allows you to get warnings in an operation by specifying the "warning" query parameter as "yes". For example, to get warnings while connecting to the Citrix ADC appliance, the URL is as follows:

  http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/login?warning=yes

  If any, the warnings are displayed in the response payload with the HTTP code "209 X-NITRO-WARNING".

- **Authenticated access for individual NITRO operations:**NITRO allows you to logon to the Citrix ADC appliance to perform individual operations. You can use this option instead of creating a NITRO session (using the login object) and then using that session to perform all operations,

  To do this, you must specify the username and password in the request header of the NITRO request as follows:

  X-NITRO-USER:<username>

  X-NITRO-PASS:<password>

  Note: In such cases, make sure that the request header DOES not include the following:

  Cookie:NITRO_AUTH_TOKEN=<tokenvalue>

`*Note:`*

Mandatory parameters are marked in redand placeholder content is marked in <green>.

update

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/subscriberradiusinterface

HTTP Method:PUT

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Content-Type:application/json

Request Payload:

```
{"subscriberradiusinterface":{

"listeningservice":<String_value>,

"radiusinterimasstart":<String_value>

}}
```

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error

unset

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/subscriberradiusinterface?**action=unset**

HTTP Method:POST

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Content-Type:application/json

Request Payload:

```
{"subscriberradiusinterface":{

"radiusinterimasstart":true

}}
```

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error

get (all)

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/subscriberradiusinterface

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "subscriberradiusinterface": [ {

"listeningservice":<String_value>,

"svrstate":<String_value>,

"radiusinterimasstart":<String_value>

}]}
```

subscribersessions

Configuration for subscriber sesions resource.

Properties

(click to see Operations)

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
ip	<String>	Read-write	Subscriber IP Address. Default value: 0
vlan	<Double>	Read-write	The vlan number on which the subscriber is located. Default value: 0 Minimum value = 0 Maximum value = 4096
nodeid	<Double>	Read-write	Unique number that identifies the cluster node. Minimum value = 0 Maximum value = 31
subscriptionidtype	<String>	Read-only	Subscription-Id type. Possible values = E164, IMSI, SIP_URI, NAI, PRIVATE
subscriptionidvalue	<String>	Read-only	Subscription-Id value.
subscriberrules	<String[]>	Read-only	Rules stored in this session for this subscriber. When PCRF sends Charging-Rule-Name or Charging-Rule-Base-Name AVP for a subscriber, Citrix ADC stores these

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
			AVPs in the subscriber session. These Rules can be retreived using 'Subscriber.Rule_Active(<rule name>)' expression. For static subscriber profiles, these rules are configured using '-subscriberRules <list of rules>'.
flags	<Double>	Read-only	Subscriber Session flags.
ttl	<Double>	Read-only	Subscriber Session revalidation Timeout remaining. This TTL gets refreshed when a radius or CCA or RAR message is received for this subscriber session. Citrix ADC will send a CCR-U after revalidation timer expires. If subscriber sessions is a negative session, then Citrix ADC will send a CCR-I after TTL expires. Negative Sessions are sessions which have not been resolved by PCRF

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
			and instead of polling PCRF continously, Citrix ADC has installed a negative session. If default subscriber is configued, then Negative Sessions inherits default subscriber profile. .
idlettl	<Double>	Read-only	Subscriber Session Activity Timeout remaining. Citrix ADC will take an idleAction after ttl expires. idleaction could be --> 1. ccrTerminate: (default) send CCR-T to inform PCRF about session termination and delete the session. 2. delete: Just delete the subscriber session without informing PCRF. 3. ccrUpdate: Do not delete the session and instead send a CCR-U to PCRF requesting

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
			for an updated session. But if this is a negative session and idleaction is ccrUpdate then Citrix ADC won't take any action. Also on negative sessions ccrTerminate translates to delete.
avpdisplaybuffer	<String>	Read-only	Subscriber Attributes Display.
servicepath	<String>	Read-only	Name of the servicepath to be taken for this subscriber.
__count	<Double>	Read-only	count parameter

Operations

(click to see Properties)

CLEAR| GET (ALL)| COUNT

Some options that you can use for each operations:

- **Getting warnings in response:**NITRO allows you to get warnings in an operation by specifying the "warning" query parameter as "yes". For example, to get warnings while connecting to the Citrix ADC appliance, the URL is as follows:

  http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/login?warning=yes

  If any, the warnings are displayed in the response payload with the HTTP code "209 X-NITRO-WARNING".

- **Authenticated access for individual NITRO operations:**NITRO allows you to logon to the Citrix ADC appliance to perform individual operations. You can use this option instead of creating a NITRO session (using the login object) and then using that session to perform all operations,

  To do this, you must specify the username and password in the request header of the NITRO request as follows:

  X-NITRO-USER:<username>

  X-NITRO-PASS:<password>

  Note: In such cases, make sure that the request header DOES not include the following:

  Cookie:NITRO_AUTH_TOKEN=<tokenvalue>

`*Note:`*

Mandatory parameters are marked in redand placeholder content is marked in <green>.

clear

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/subscribersessions?**action=clear**

HTTP Method:POST

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Content-Type:application/json

Request Payload:

```
{"subscribersessions":{

"ip":<String_value>,

"vlan":<Double_value>

}}
```

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error

get (all)

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/subscribersessions

Query-parameters:

args

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/subscribersessions?**args=ip:<String_value>,vlan:<Double_value>,nodeid:<Double_value>**

Use this query-parameter to get subscribersessions resources based on additional properties.

attrs

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/subscribersessions?**attrs=property-name1,property-name2**

Use this query parameter to specify the resource details that you want to retrieve.

filter

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/subscribersessions?**filter=property-name1:property-val1,property-name2:property-val2**

Use this query-parameter to get the filtered set of subscribersessions resources configured on Citrix ADC. Filtering can be done on any of the properties of the resource.

view

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/subscribersessions?**view=summary**

Note: By default, the retrieved results are displayed in detail view (?view=detail).

pagination

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/subscribersessions?**pagesize=#no;pageno=#no**

Use this query-parameter to get the subscribersessions resources in chunks.

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "subscribersessions": [ {

ip:<String_value>,vlan:<Double_value>,nodeid:<Double_value>"subscriptionid
type":<String_value>,

"subscriptionidvalue":<String_value>,

"subscriberrules":<String[]_value>,

"flags":<Double_value>,

"ttl":<Double_value>,

"idlettl":<Double_value>,

"avpdisplaybuffer":<String_value>,

"servicepath":<String_value>

}]}
```

count

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/subscribersessions?**count=yes**

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "subscribersessions": [ { "__count": "#no"} ] }
```

SureConnect

Sure Connect

SureConnect configuration. SureConnect allows the system to deliver alternate content to clients when the system is under heavy load for the original URL. The original content will be served to the client, automatically, after an estimated delay period.

NAME	DESCRIPTION
scparameter	Configuration for SC parameter
scpolicy	Configuration for SureConnect policy

scparameter

Configuration for SC parameter resource.

Properties

(click to see Operations)

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
sessionlife	<Double>	Read-write	Time, in seconds, between the first time and the next time the SureConnect alternative content window is displayed. The alternative content window is displayed only once during a session for the same browser accessing a configured URL, so this parameter determines the length of a session. Default value: 300 Minimum value = 1 Maximum value = 4294967294
vsr	<String>	Read-write	File containing the customized response to be displayed when the ACTION in the SureConnect policy is set to NS. Default value: "DEFAULT" Minimum length = 1

Operations

(click to see Properties)

UPDATE| UNSET| GET (ALL)

Some options that you can use for each operations:

- **Getting warnings in response:**NITRO allows you to get warnings in an operation by specifying the "warning" query parameter as "yes". For example, to get warnings while connecting to the Citrix ADC appliance, the URL is as follows:

  http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/login?warning=yes

  If any, the warnings are displayed in the response payload with the HTTP code "209 X-NITRO-WARNING".

- **Authenticated access for individual NITRO operations:**NITRO allows you to logon to the Citrix ADC appliance to perform individual operations. You can use this option instead of creating a NITRO session (using the login object) and then using that session to perform all operations,

  To do this, you must specify the username and password in the request header of the NITRO request as follows:

  X-NITRO-USER:<username>

  X-NITRO-PASS:<password>

  Note: In such cases, make sure that the request header DOES not include the following:

  Cookie:NITRO_AUTH_TOKEN=<tokenvalue>

`*Note:`*

Mandatory parameters are marked in redand placeholder content is marked in <green>.

update

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/scparameter

HTTP Method:PUT

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Content-Type:application/json

Request Payload:

```
{"scparameter":{

"sessionlife":<Double_value>,

"vsr":<String_value>

}}
```

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error

unset

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/scparameter?**action=unset**

HTTP Method:POST

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Content-Type:application/json

Request Payload:

```
{"scparameter":{

"sessionlife":true,

"vsr":true

}}
```

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error

get (all)

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/scparameter

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "scparameter": [ {

"sessionlife":<Double_value>,

"vsr":<String_value>

}]}
```

scpolicy

Configuration for SureConnect policy resource.

Properties

(click to see Operations)

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
name	<String>	Read-write	Name for the policy. Must begin with an ASCII alphabetic or underscore (_) character, and must contain only ASCII alphanumeric, underscore, hash (#), period (.), space, colon (:), at (@), equals (=), and hyphen (-) characters. Minimum length = 1 Maximum length = 31
url	<String>	Read-write	URL against which to match incoming client request. Maximum length = 127
rule	<String>	Read-write	Expression against which the traffic is evaluated. Maximum length of a string literal in the expression is 255 characters. A longer string can be split into smaller strings of up to 255 characters each, and the smaller strings concatenated with the + operator. For

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
			example, you can create a 500-character string as follows: "'<string of 255 characters>" + "<string of 245 characters>'" The following requirements apply only to the Citrix ADC CLI: * If the expression includes one or more spaces, enclose the entire expression in double quotation marks. * If the expression itself includes double quotation marks, escape the quotations by using the character. * Alternatively, you can use single quotation marks to enclose the rule, in which case you do not have to escape the double quotation marks.

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
			Maximum length = 1499
delay	<Double>	Read-write	Delay threshold, in microseconds, for requests that match the policy's URL or rule. If the delay statistics gathered for the matching request exceed the specified delay, SureConnect is triggered for that request. Minimum value = 1 Maximum value = 599999999
maxconn	<Double>	Read-write	Maximum number of concurrent connections that can be open for requests that match the policy's URL or rule. Minimum value = 1 Maximum value = 4294967294
action	<String>	Read-write	Action to be taken when the delay or maximum-connections threshold is reached. Available settings function as follows:

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
			ACS - Serve content from an alternative content service. NS - Serve alternative content from the Citrix ADC. NO ACTION - Serve no alternative content. However, delay statistics are still collected for the configured URLs, and, if the Maximum Client Connections parameter is set, the number of connections is limited to the value specified by that parameter. (However, alternative content is not served even if the maxConn threshold is met). Possible values = ACS, NS, NOACTION
altcontentsvcname	<String>	Read-write	Name of the alternative content service to be used in the ACS action.

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
			Minimum length = 1 Maximum length = 127
altcontentpath	<String>	Read-write	Path to the alternative content service to be used in the ACS action. Minimum length = 1 Maximum length = 127
__count	<Double>	Read-only	count parameter

Operations

(click to see Properties)

ADD| DELETE| UPDATE| UNSET| GET (ALL)| GET| COUNT

Some options that you can use for each operations:

- **Getting warnings in response:**NITRO allows you to get warnings in an operation by specifying the "warning" query parameter as "yes". For example, to get warnings while connecting to the Citrix ADC appliance, the URL is as follows:

  http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/login?warning=yes

  If any, the warnings are displayed in the response payload with the HTTP code "209 X-NITRO-WARNING".

- **Authenticated access for individual NITRO operations:**NITRO allows you to logon to the Citrix ADC appliance to perform individual operations. You can use this option instead of creating a NITRO session (using the login object) and then using that session to perform all operations,

  To do this, you must specify the username and password in the request header of the NITRO request as follows:

  X-NITRO-USER:<username>

X-NITRO-PASS:<password>

Note: In such cases, make sure that the request header DOES not include the following:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue>

`*Note:`*

Mandatory parameters are marked in redand placeholder content is marked in <green>.

add

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/scpolicy

HTTP Method:POST

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Content-Type:application/json

Request Payload:

```
{"scpolicy":{

<b>"name":<String_value>,

</b>"url":<String_value>,

"rule":<String_value>,

"delay":<Double_value>,

"maxconn":<Double_value>,

"action":<String_value>,

"altcontentsvcname":<String_value>,

"altcontentpath":<String_value>

}}
```

Response:

HTTP Status Code on Success: 201 Created HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error

delete

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/scpolicy/name_value<String>

HTTP Method:DELETE

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue>

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error

update

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/scpolicy

HTTP Method:PUT

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Content-Type:application/json

Request Payload:

```
{"scpolicy":{

<b>"name":<String_value>,

</b>"url":<String_value>,

"rule":<String_value>,

"delay":<Double_value>,
```

```
"maxconn":<Double_value>,

"action":<String_value>,

"altcontentsvcname":<String_value>,

"altcontentpath":<String_value>

}}
```

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error

unset

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/scpolicy?**action=unset**

HTTP Method:POST

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Content-Type:application/json

Request Payload:

```
{"scpolicy":{

<b>"name":<String_value>,

</b>"delay":true,

"maxconn":true

}}
```

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error

get (all)

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/scpolicy

Query-parameters:

attrs

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/scpolicy?**attrs=property-name1,property-name2**

Use this query parameter to specify the resource details that you want to retrieve.

filter

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/scpolicy?**filter=property-name1:property-val1,property-name2:property-val2**

Use this query-parameter to get the filtered set of scpolicy resources configured on Citrix ADC. Filtering can be done on any of the properties of the resource.

view

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/scpolicy?**view=summary**

Note: By default, the retrieved results are displayed in detail view (?view=detail).

pagination

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/scpolicy?**pagesize=#no;pageno=#no**

Use this query-parameter to get the scpolicy resources in chunks.

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "scpolicy": [ {

"name":<String_value>,

"url":<String_value>,

"rule":<String_value>,

"delay":<Double_value>,

"maxconn":<Double_value>,

"action":<String_value>,

"altcontentsvcname":<String_value>,

"altcontentpath":<String_value>

}]}
```

get

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/scpolicy/name_value<String>

Query-parameters:

attrs

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/scpolicy/name_value<String>?**attrs=property-name1,property-name2**

Use this query parameter to specify the resource details that you want to retrieve.

view

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/scpolicy/name_value<String>?**view=summary**

Note: By default, the retrieved results are displayed in detail view (?view=detail).

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "scpolicy": [ {

"name":<String_value>,

"url":<String_value>,

"rule":<String_value>,

"delay":<Double_value>,

"maxconn":<Double_value>,

"action":<String_value>,

"altcontentsvcname":<String_value>,

"altcontentpath":<String_value>

}]}
```

count

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/scpolicy?**count=yes**

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "scpolicy": [ { "__count": "#no"} ] }
```

System

System

System configuration.

NAME	DESCRIPTION
systemautorestorefeature	Configuration for Enable/Disable the autorestore feature use to create restorepoint
systembackup	Configuration for Backup Data for ns backup and restore
systemcmdpolicy	Configuration for command policy

NAME	DESCRIPTION
systemcollectionparam	Configuration for collection parameter
systemcore	Configuration for core
systemcountergroup	Configuration for counter group
systemcounters	Configuration for counters
systemdatasource	Configuration for historical datasource
systementity	Configuration for entity
systementitydata	Configuration for entity data
systementitytype	Configuration for entity type
systemeventhistory	Configuration for event history
systemextramgmtcpu	
systemfile	Configuration for file
systemglobal_auditnslogpolicy_binding	Binding object showing the auditnslogpolicy that can be bound to systemglobal
systemglobal_auditsyslogpolicy_binding	Binding object showing the auditsyslogpolicy that can be bound to systemglobal
systemglobal_authenticationldappolicy_binding	Binding object showing the authenticationldappolicy that can be bound to systemglobal
systemglobal_authenticationlocalpolicy_binding	Binding object showing the authenticationlocalpolicy that can be bound to systemglobal
systemglobal_authenticationpolicy_binding	Binding object showing the authenticationpolicy that can be bound to systemglobal

NAME	DESCRIPTION
systemglobal_authenticationradiuspolicy_binding	Binding object showing the authenticationradiuspolicy that can be bound to systemglobal
systemglobal_authenticationtacacspolicy_binding	Binding object showing the authenticationtacacspolicy that can be bound to systemglobal
systemglobal_binding	Binding object showing the resources that can be bound to systemglobal
systemglobaldata	Configuration for global counter data
systemgroup	Configuration for system group
systemgroup_binding	Binding object showing the resources that can be bound to systemgroup
systemgroup_nspartition_binding	Binding object showing the nspartition that can be bound to systemgroup
systemgroup_systemcmdpolicy_binding	Binding object showing the systemcmdpolicy that can be bound to systemgroup
systemgroup_systemuser_binding	Binding object showing the systemuser that can be bound to systemgroup
systemhwerror	Configuration for Hardware errors
systemkek	
systemparameter	Configuration for system parameter
systemrestorepoint	Configuration for Setting restorepoints for auto restore
systemsession	Configuration for system session
systemsshkey	
systemuser	Configuration for system user

NAME	DESCRIPTION
systemuser_binding	Binding object showing the resources that can be bound to systemuser
systemuser_nspartition_binding	Binding object showing the nspartition that can be bound to systemuser
systemuser_systemcmdpolicy_binding	Binding object showing the systemcmdpolicy that can be bound to systemuser
systemuser_systemgroup_binding	Binding object showing the systemgroup that can be bound to systemuser

systembackup

Configuration for Backup Data for ns backup and restore resource.

Properties

(click to see Operations)

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
filename	<String>	Read-write	Name of the backup file(*.tgz) to be restored.
uselocaltimezone	<Boolean>	Read-write	This option will create backup file with local timezone timestamp.
level	<String>	Read-write	Level of data to be backed up. Default value: basic Possible values = basic, full
includekernel	<String>	Read-write	Use this option to add kernel in the backup file.

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
			Default value: NO Possible values = NO, YES
comment	<String>	Read-write	Comment specified at the time of creation of the backup file(*.tgz).
skipbackup	<Boolean>	Read-write	Use this option to skip taking backup during restore operation.
size	<Double>	Read-only	Size of the backup file(*.tgz) in KB.
creationtime	<String>	Read-only	Creation time of the backup file(*.tgz).
version	<String>	Read-only	Build version of the backup file(*.tgz).
createdby	<String>	Read-only	Name of user who created the backup file(*.tgz).
ipaddress	<String>	Read-only	Ip of Citrix ADC box where the backup file(*.tgz) was created.
__count	<Double>	Read-only	count parameter

Operations

(click to see Properties)

CREATE| RESTORE| ADD| DELETE| GET (ALL)| GET| COUNT

Some options that you can use for each operations:

- **Getting warnings in response:**NITRO allows you to get warnings in an operation by specifying the "warning" query parameter as "yes". For example, to get warnings while connecting to the Citrix ADC appliance, the URL is as follows:

  http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/login?warning=yes

  If any, the warnings are displayed in the response payload with the HTTP code "209 X-NITRO-WARNING".

- **Authenticated access for individual NITRO operations:**NITRO allows you to logon to the Citrix ADC appliance to perform individual operations. You can use this option instead of creating a NITRO session (using the login object) and then using that session to perform all operations,

  To do this, you must specify the username and password in the request header of the NITRO request as follows:

  X-NITRO-USER:<username>

  X-NITRO-PASS:<password>

  Note: In such cases, make sure that the request header DOES not include the following:

  Cookie:NITRO_AUTH_TOKEN=<tokenvalue>

`*Note:`*

Mandatory parameters are marked in redand placeholder content is marked in <green>.

create

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/systembackup?**action=create**

HTTP Method:POST

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Content-Type:application/json

Request Payload:

```
{"systembackup":{
```

```
"filename":<String_value>,

"uselocaltimezone":<Boolean_value>,

"level":<String_value>,

"includekernel":<String_value>,

"comment":<String_value>

}}
```

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error

restore

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/systembackup?**action=restore**

HTTP Method:POST

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Content-Type:application/json

Request Payload:

```
{"systembackup":{

<b>"filename":<String_value>,

</b>"skipbackup":<Boolean_value>

}}
```

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error

add

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/systembackup

HTTP Method:POST

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Content-Type:application/json

Request Payload:

```
{"systembackup":{

<b>"filename":<String_value>

</b>}}
```

Response:

HTTP Status Code on Success: 201 Created HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error

delete

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/systembackup/filename_value<String>

HTTP Method:DELETE

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue>

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error

get (all)

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/systembackup

Query-parameters:

attrs

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/systembackup?**attrs=property-name1,property-name2**

Use this query parameter to specify the resource details that you want to retrieve.

filter

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/systembackup?**filter=property-name1:property-val1,property-name2:property-val2**

Use this query-parameter to get the filtered set of systembackup resources configured on Citrix ADC. Filtering can be done on any of the properties of the resource.

view

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/systembackup?**view=summary**

Note: By default, the retrieved results are displayed in detail view (?view=detail).

pagination

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/systembackup?**pagesize=#no;pageno=#no**

Use this query-parameter to get the systembackup resources in chunks.

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "systembackup": [ {

"filename":<String_value>,

"level":<String_value>,

"includekernel":<String_value>,

"comment":<String_value>,

"size":<Double_value>,

"uselocaltimezone":<Boolean_value>,

"creationtime":<String_value>,

"version":<String_value>,

"createdby":<String_value>,

"ipaddress":<String_value>

}]}
```

get

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/systembackup/filename_value<String>

Query-parameters:

attrs

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/systembackup/filename_value<String>?**attrs=property-name1,property-name2**

Use this query parameter to specify the resource details that you want to retrieve.

view

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/systembackup/filename_value<String>?**view=summary**

Note: By default, the retrieved results are displayed in detail view (?view=detail).

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "systembackup": [ {

"filename":<String_value>,

"level":<String_value>,

"includekernel":<String_value>,

"comment":<String_value>,

"size":<Double_value>,

"uselocaltimezone":<Boolean_value>,
```

```
"creationtime":<String_value>,

"version":<String_value>,

"createdby":<String_value>,

"ipaddress":<String_value>

}]}
```

count

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/systembackup?**count=yes**

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "systembackup": [ { "__count": "#no"} ] }
```

systemcmdpolicy

Configuration for command policy resource.

Properties

(click to see Operations)

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
policyname	<String>	Read-write	Name for a command policy. Must begin with a letter, number, or the underscore (_) character, and must contain only alphanumeric, hyphen (-), period (.), hash (#), space ( ), at (@), equal (=), colon (:), and underscore characters. Cannot be changed after the policy is created. CLI Users: If the name includes one or more spaces, enclose the name in double or single quotation marks (for example, "my policy" or 'my policy'). Minimum length = 1
action	<String>	Read-write	Action to perform when a request matches the policy. Possible values = ALLOW, DENY
cmdspec	<String>	Read-write	Regular expression specifying the data that matches the policy. Minimum length = 1

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
__count	<Double>	Read-only	count parameter

Operations

(click to see Properties)

ADD| DELETE| UPDATE| GET (ALL)| GET| COUNT

Some options that you can use for each operations:

- **Getting warnings in response:**NITRO allows you to get warnings in an operation by specifying the "warning" query parameter as "yes". For example, to get warnings while connecting to the Citrix ADC appliance, the URL is as follows:

  http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/login?warning=yes

  If any, the warnings are displayed in the response payload with the HTTP code "209 X-NITRO-WARNING".

- **Authenticated access for individual NITRO operations:**NITRO allows you to logon to the Citrix ADC appliance to perform individual operations. You can use this option instead of creating a NITRO session (using the login object) and then using that session to perform all operations,

  To do this, you must specify the username and password in the request header of the NITRO request as follows:

  X-NITRO-USER:<username>

  X-NITRO-PASS:<password>

  Note: In such cases, make sure that the request header DOES not include the following:

  Cookie:NITRO_AUTH_TOKEN=<tokenvalue>

`*Note:`*

Mandatory parameters are marked in redand placeholder content is marked in <green>.

add

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/systemcmdpolicy

HTTP Method:POST

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Content-Type:application/json

Request Payload:

```
{"systemcmdpolicy":{

<b>"policyname":<String_value>,

</b><b>"action":<String_value>,

</b><b>"cmdspec":<String_value>

</b>}}
```

Response:

HTTP Status Code on Success: 201 Created HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error

delete

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/systemcmdpolicy/policyname_value<String>

HTTP Method:DELETE

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue>

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error

update

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/systemcmdpolicy

HTTP Method:PUT

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Content-Type:application/json

Request Payload:

```
{"systemcmdpolicy":{

<b>"policyname":<String_value>,

</b><b>"action":<String_value>,

</b><b>"cmdspec":<String_value>

</b>}}
```

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error

get (all)

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/systemcmdpolicy

Query-parameters:

attrs

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/systemcmdpolicy?**attrs=property-name1,property-name2**

Use this query parameter to specify the resource details that you want to retrieve.

filter

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/systemcmdpolicy?**filter=property-name1:property-val1,property-name2:property-val2**

Use this query-parameter to get the filtered set of systemcmdpolicy resources configured on Citrix ADC. Filtering can be done on any of the properties of the resource.

view

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/systemcmdpolicy?**view=summary**

Note: By default, the retrieved results are displayed in detail view (?view=detail).

pagination

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/systemcmdpolicy?**pagesize=#no;pageno=#no**

Use this query-parameter to get the systemcmdpolicy resources in chunks.

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "systemcmdpolicy": [ {

"policyname":<String_value>,

"action":<String_value>,

"cmdspec":<String_value>

}]}
```

get

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/systemcmdpolicy/policyname_value<String>

Query-parameters:

attrs

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/systemcmdpolicy/policyname_value<String>?**attrs=property-name1,property-name2**

Use this query parameter to specify the resource details that you want to retrieve.

view

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/systemcmdpolicy/policyname_value<String>?**view=summary**

Note: By default, the retrieved results are displayed in detail view (?view=detail).

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "systemcmdpolicy": [ {

"policyname":<String_value>,
```

```
"action":<String_value>,

"cmdspec":<String_value>

}]}
```

count

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/systemcmdpolicy?**count=yes**

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "systemcmdpolicy": [ { "__count": "#no"} ] }
```

systemcollectionparam

Configuration for collection parameter resource.

Properties

(click to see Operations)

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
communityname	<String>	Read-write	SNMPv1 community name for authentication.
loglevel	<String>	Read-write	specify the log level. Possible values CRITICAL,WARNING,INFO,DEBUG1,DEBUG2. Minimum length = 1
datapath	<String>	Read-write	specify the data path to the database. Minimum length = 1

Operations

(click to see Properties)

UPDATE| UNSET| GET (ALL)

Some options that you can use for each operations:

- **Getting warnings in response:**NITRO allows you to get warnings in an operation by specifying the "warning" query parameter as "yes". For example, to get warnings while connecting to the Citrix ADC appliance, the URL is as follows:

  http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/login?warning=yes

  If any, the warnings are displayed in the response payload with the HTTP code "209 X-NITRO-WARNING".

- **Authenticated access for individual NITRO operations:**NITRO allows you to logon to the Citrix ADC appliance to perform individual operations. You can use this option instead of creating a NITRO session (using the login object) and then using that session to perform all operations,

  To do this, you must specify the username and password in the request header of the NITRO request as follows:

  X-NITRO-USER:<username>

X-NITRO-PASS:<password>

Note: In such cases, make sure that the request header DOES not include the following:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue>

`*Note:`*

Mandatory parameters are marked in redand placeholder content is marked in <green>.

update

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/systemcollectionparam

HTTP Method:PUT

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Content-Type:application/json

Request Payload:

```
{"systemcollectionparam":{

"communityname":<String_value>,

"loglevel":<String_value>,

"datapath":<String_value>

}}
```

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error

unset

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/systemcollectionparam?**action=unset**

HTTP Method:POST

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Content-Type:application/json

Request Payload:

```
{"systemcollectionparam":{

"communityname":true,

"loglevel":true,

"datapath":true

}}
```

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error

get (all)

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/systemcollectionparam

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "systemcollectionparam": [ {
"communityname":<String_value>,
"loglevel":<String_value>,
"datapath":<String_value>
}]}
```

systemcore

Configuration for core resource.

Properties

(click to see Operations)

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
datasource	<String>	Read-write	Specifies the source which contains all the stored counter values.
response	<String>	Read-only	.

Operations

(click to see Properties)

GET (ALL)

Some options that you can use for each operations:

- **Getting warnings in response:**NITRO allows you to get warnings in an operation by specifying the "warning" query parameter as "yes". For example, to get warnings while connecting to the Citrix ADC appliance, the URL is as follows:

  http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/login?warning=yes

If any, the warnings are displayed in the response payload with the HTTP code "209 X-NITRO-WARNING".

- **Authenticated access for individual NITRO operations:**NITRO allows you to logon to the Citrix ADC appliance to perform individual operations. You can use this option instead of creating a NITRO session (using the login object) and then using that session to perform all operations,

  To do this, you must specify the username and password in the request header of the NITRO request as follows:

  X-NITRO-USER:<username>

  X-NITRO-PASS:<password>

  Note: In such cases, make sure that the request header DOES not include the following:

  Cookie:NITRO_AUTH_TOKEN=<tokenvalue>

`*Note:`*

Mandatory parameters are marked in redand placeholder content is marked in <green>.

get (all)

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/systemcore

Query-parameters:

args

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/systemcore?
args=datasource:<String_value>

Use this query-parameter to get systemcore resources based on additional properties.

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "systemcore": [ {

datasource:<String_value>"response":<String_value>

}]}
```

systemcountergroup

Configuration for counter group resource.

Properties

(click to see Operations)

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
datasource	<String>	Read-write	Specifies the source which contains all the stored counter values.
response	<String>	Read-only	.

Operations

(click to see Properties)

GET (ALL)

Some options that you can use for each operations:

- **Getting warnings in response:**NITRO allows you to get warnings in an operation by specifying the "warning" query parameter as "yes". For example, to get warnings while connecting to the Citrix ADC appliance, the URL is as follows:

  http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/login?warning=yes

  If any, the warnings are displayed in the response payload with the HTTP code "209 X-NITRO-WARNING".

- **Authenticated access for individual NITRO operations:**NITRO allows you to logon to the Citrix ADC appliance to perform individual operations. You can use this option instead of creating a NITRO session (using the login object) and then using that session to perform all operations,

  To do this, you must specify the username and password in the request header of the NITRO request as follows:

  X-NITRO-USER:<username>

  X-NITRO-PASS:<password>

  Note: In such cases, make sure that the request header DOES not include the following:

  Cookie:NITRO_AUTH_TOKEN=<tokenvalue>

`*Note:`*

Mandatory parameters are marked in redand placeholder content is marked in <green>.

get (all)

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/systemcountergroup

Query-parameters:

args

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/systemcountergroup?**args=datasource:<String_value>**

Use this query-parameter to get systemcountergroup resources based on additional properties.

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "systemcountergroup": [ {

datasource:<String_value>"response":<String_value>

}]}
```

systemcounters

Configuration for counters resource.

Properties

(click to see Operations)

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
countergroup	<String>	Read-write	Specify the (counter) group name which contains all the counters specific tot his particular group.
datasource	<String>	Read-write	Specifies the source which contains all the stored counter values.
response	<String>	Read-only	.

Operations

(click to see Properties)

GET (ALL)

Some options that you can use for each operations:

- **Getting warnings in response:**NITRO allows you to get warnings in an operation by specifying the "warning" query parameter as "yes". For example, to get warnings while connecting to the Citrix ADC appliance, the URL is as follows:

  http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/login?warning=yes

  If any, the warnings are displayed in the response payload with the HTTP code "209 X-NITRO-WARNING".

- **Authenticated access for individual NITRO operations:**NITRO allows you to logon to the Citrix ADC appliance to perform individual operations. You can use this option instead of creating a NITRO session (using the login object) and then using that session to perform all operations,

  To do this, you must specify the username and password in the request header of the NITRO request as follows:

  X-NITRO-USER:<username>

  X-NITRO-PASS:<password>

  Note: In such cases, make sure that the request header DOES not include the following:

  Cookie:NITRO_AUTH_TOKEN=<tokenvalue>

`*Note:`*

Mandatory parameters are marked in redand placeholder content is marked in <green>.

get (all)

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/systemcounters

Query-parameters:

args

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/systemcounters?**args=countergroup:<String_value>,datasource:<String_value>**

Use this query-parameter to get systemcounters resources based on additional properties.

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "systemcounters": [ {

countergroup:<String_value>,datasource:<String_value>"response":<String_va
lue>

}]}
```

systemdatasource

Configuration for historical datasource resource.

Properties

(click to see Operations)

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
datasource	<String>	Read-write	Specifies the source which contains all

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
			the stored counter values.
response	<String>	Read-only	.

Operations

(click to see Properties)

GET (ALL)

Some options that you can use for each operations:

- **Getting warnings in response:**NITRO allows you to get warnings in an operation by specifying the "warning" query parameter as "yes". For example, to get warnings while connecting to the Citrix ADC appliance, the URL is as follows:

  http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/login?warning=yes

  If any, the warnings are displayed in the response payload with the HTTP code "209 X-NITRO-WARNING".
- **Authenticated access for individual NITRO operations:**NITRO allows you to logon to the Citrix ADC appliance to perform individual operations. You can use this option instead of creating a NITRO session (using the login object) and then using that session to perform all operations,

  To do this, you must specify the username and password in the request header of the NITRO request as follows:

  X-NITRO-USER:<username>

  X-NITRO-PASS:<password>

  Note: In such cases, make sure that the request header DOES not include the following:

  Cookie:NITRO_AUTH_TOKEN=<tokenvalue>

`*Note:`*

Mandatory parameters are marked in redand placeholder content is marked in <green>.

get (all)

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/systemdatasource

Query-parameters:

args

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/systemdatasource?**args=datasource:<String_value>**

Use this query-parameter to get systemdatasource resources based on additional properties.

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "systemdatasource": [ {

datasource:<String_value>"response":<String_value>

}]}
```

systementity

Configuration for entity resource.

Properties

(click to see Operations)

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
type	<String>	Read-write	Specify the entity type.
datasource	<String>	Read-write	Specifies the source which contains all the stored counter values.
core	<Integer>	Read-write	Specify core ID of the PE in nCore.
response	<String>	Read-only	.

Operations

(click to see Properties)

GET (ALL)

Some options that you can use for each operations:

- **Getting warnings in response:**NITRO allows you to get warnings in an operation by specifying the "warning" query parameter as "yes". For example, to get warnings while connecting to the Citrix ADC appliance, the URL is as follows:

  http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/login?warning=yes

  If any, the warnings are displayed in the response payload with the HTTP code "209 X-NITRO-WARNING".
- **Authenticated access for individual NITRO operations:**NITRO allows you to logon to the Citrix ADC appliance to perform individual operations. You can use this option instead of creating a NITRO session (using the login object) and then using that session to perform all operations,

  To do this, you must specify the username and password in the request header of the NITRO request as follows:

  X-NITRO-USER:<username>

  X-NITRO-PASS:<password>

Note: In such cases, make sure that the request header DOES not include the following:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue>

`*Note:` *

Mandatory parameters are marked in redand placeholder content is marked in <green>.

get (all)

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/systementity

Query-parameters:

args

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/systementity?
args=type:<String_value>,datasource:<String_value>,core:<Integer_value>

Use this query-parameter to get systementity resources based on additional properties.

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "systementity": [ {

<b>type:<String_value>,</
b>datasource:<String_value>,core:<Integer_value>"response":<String_value>

}]}
```

systementitydata

Configuration for entity data resource.

Properties

(click to see Operations)

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
type	<String>	Read-write	Specify the entity type.
name	<String>	Read-write	Specify the entity name.
alldeleted	<Boolean>	Read-write	Specify this if you would like to delete information about all deleted entities from the database.
allinactive	<Boolean>	Read-write	Specify this if you would like to delete information about all inactive entities from the database.
datasource	<String>	Read-write	Specifies the source which contains all the stored counter values.
core	<Integer>	Read-write	Specify core ID of the PE in nCore.
counters	<String>	Read-write	Specify the counters to be collected.
starttime	<String>	Read-write	Specify start time in mmddyyyyhhmm to start collecting

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
			values from that timestamp.
endtime	<String>	Read-write	Specify end time in mmddyyyyhhmm upto which values have to be collected.
last	<Integer>	Read-write	Last is literal way of saying a certain time period from the current moment. Example: -last 1 hour, -last 1 day, et cetera. Default value: 1
unit	<String>	Read-write	Specify the time period from current moment. Example 1 x where x = hours/ days/ years. Possible values = HOURS, DAYS, MONTHS
response	<String>	Read-only	.
startupdate	<Double>	Read-only	.
lastupdate	<Double>	Read-only	.

Operations

(click to see Properties)

DELETE| GET (ALL)

Some options that you can use for each operations:

- **Getting warnings in response:**NITRO allows you to get warnings in an operation by specifying the "warning" query parameter as "yes". For example, to get warnings while connecting to the Citrix ADC appliance, the URL is as follows:

  http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/login?warning=yes

  If any, the warnings are displayed in the response payload with the HTTP code "209 X-NITRO-WARNING".

- **Authenticated access for individual NITRO operations:**NITRO allows you to logon to the Citrix ADC appliance to perform individual operations. You can use this option instead of creating a NITRO session (using the login object) and then using that session to perform all operations,

  To do this, you must specify the username and password in the request header of the NITRO request as follows:

  X-NITRO-USER:<username>

  X-NITRO-PASS:<password>

  Note: In such cases, make sure that the request header DOES not include the following:

  Cookie:NITRO_AUTH_TOKEN=<tokenvalue>

`*Note:`*

Mandatory parameters are marked in redand placeholder content is marked in <green>.

delete

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/systementitydata

HTTP Method:DELETE

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue>

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error

get (all)

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/systementitydata

Query-parameters:

args

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/systementitydata?
args=type:<String_value>,name:<String_value>,counters:<String_value>,starttime:<String_value>,endt

Use this query-parameter to get systementitydata resources based on additional properties.

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "systementitydata": [ {

<b>type:<String_value>,</b><b>name:<String_value>,</
b><b>counters:<String_value>,</
b>starttime:<String_value>,endtime:<String_value>,last:<Integer_value>,uni
t:<String_value>,datasource:<String_value>,core:<Integer_value>"response":
<String_value>,

"startupdate":<Double_value>,

"lastupdate":<Double_value>

}]}
```

systementitytype

Configuration for entity type resource.

Properties

(click to see Operations)

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
datasource	<String>	Read-write	Specifies the source which contains all the stored counter values.
response	<String>	Read-only	.

Operations

(click to see Properties)

GET (ALL)

Some options that you can use for each operations:

- **Getting warnings in response:**NITRO allows you to get warnings in an operation by specifying the "warning" query parameter as "yes". For example, to get warnings while connecting to the Citrix ADC appliance, the URL is as follows:

  http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/login?warning=yes

  If any, the warnings are displayed in the response payload with the HTTP code "209 X-NITRO-WARNING".
- **Authenticated access for individual NITRO operations:**NITRO allows you to logon to the Citrix ADC appliance to perform individual operations. You can use this option instead of creating a NITRO session (using the login object) and then using that session to perform all operations,

  To do this, you must specify the username and password in the request header of the NITRO request as follows:

X-NITRO-USER:<username>

X-NITRO-PASS:<password>

Note: In such cases, make sure that the request header DOES not include the following:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue>

`*Note:`*

Mandatory parameters are marked in redand placeholder content is marked in <green>.

get (all)

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/systementitytype

Query-parameters:

args

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/systementitytype?
args=datasource:<String_value>

Use this query-parameter to get systementitytype resources based on additional properties.

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "systementitytype": [ {

datasource:<String_value>"response":<String_value>
```

```
}]}
```

systemeventhistory

Configuration for event history resource.

Properties

(click to see Operations)

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
starttime	<String>	Read-write	Specify start time in mmddyyyyhhmm to start collecting values from that timestamp.
endtime	<String>	Read-write	Specify end time in mmddyyyyhhmm upto which values have to be collected.
last	<Integer>	Read-write	Last is literal way of saying a certain time period from the current moment. Example: -last 1 hour, -last 1 day, et cetera. Default value: 1
unit	<String>	Read-write	Specify the time period from current moment. Example 1 x where x = hours/ days/ years. Possible values = HOURS, DAYS, MONTHS

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
datasource	<String>	Read-write	Specifies the source which contains all the stored counter values.
response	<String>	Read-only	.

Operations

(click to see Properties)

GET (ALL)

Some options that you can use for each operations:

- **Getting warnings in response:**NITRO allows you to get warnings in an operation by specifying the "warning" query parameter as "yes". For example, to get warnings while connecting to the Citrix ADC appliance, the URL is as follows:

  http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/login?warning=yes

  If any, the warnings are displayed in the response payload with the HTTP code "209 X-NITRO-WARNING".

- **Authenticated access for individual NITRO operations:**NITRO allows you to logon to the Citrix ADC appliance to perform individual operations. You can use this option instead of creating a NITRO session (using the login object) and then using that session to perform all operations,

  To do this, you must specify the username and password in the request header of the NITRO request as follows:

  X-NITRO-USER:<username>

  X-NITRO-PASS:<password>

  Note: In such cases, make sure that the request header DOES not include the following:

  Cookie:NITRO_AUTH_TOKEN=<tokenvalue>

`*Note:`*

Mandatory parameters are marked in redand placeholder content is marked in <green>.

get (all)

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/systemeventhistory

Query-parameters:

args

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/systemeventhistory?
args=starttime:<String_value>,endtime:<String_value>,last:<Integer_value>,unit:<String_value>,dataso

Use this query-parameter to get systemeventhistory resources based on additional properties.

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "systemeventhistory": [ {

starttime:<String_value>,endtime:<String_value>,last:<Integer_value>,unit:
<String_value>,<b>datasource:<String_value>,</b>"response":<String_value>

}]}
```

systemextramgmtcpu

Configuration for 0 resource.

Properties

(click to see Operations)

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
nodeid	<Double>	Read-write	Unique number that identifies the cluster node. Minimum value = 0 Maximum value = 31
configuredstate	<String>	Read-only	Get the configured state of extra manahement cpu. Default value: DISABLED Possible values = ENABLED, DISABLED
effectivestate	<String>	Read-only	Get the current running state of extra management cpu. Default value: DISABLED Possible values = ENABLED, DISABLED

Operations

(click to see Properties)

ENABLE| DISABLE| GET (ALL)

Some options that you can use for each operations:

- **Getting warnings in response:**NITRO allows you to get warnings in an operation by specifying the "warning" query parameter as "yes". For example, to get warnings while connecting to the Citrix ADC appliance, the URL is as follows:

  http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/login?warning=yes

  If any, the warnings are displayed in the response payload with the HTTP code "209 X-NITRO-WARNING".

- **Authenticated access for individual NITRO operations:**NITRO allows you to logon to the Citrix ADC appliance to perform individual operations. You can use this option instead of creating a NITRO session (using the login object) and then using that session to perform all operations,

  To do this, you must specify the username and password in the request header of the NITRO request as follows:

  X-NITRO-USER:<username>

  X-NITRO-PASS:<password>

  Note: In such cases, make sure that the request header DOES not include the following:

  Cookie:NITRO_AUTH_TOKEN=<tokenvalue>

`*Note:`*

Mandatory parameters are marked in redand placeholder content is marked in <green>.

enable

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/systemextramgmtcpu?**action=enable**

HTTP Method:POST

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Content-Type:application/json

Request Payload:

```
{"systemextramgmtcpu":{
```

```
}}
```

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error

disable

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/systemextramgmtcpu?**action=disable**

HTTP Method:POST

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Content-Type:application/json

Request Payload:

```
{"systemextramgmtcpu":{

}}
```

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error

get (all)

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/systemextramgmtcpu

Query-parameters:

args

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/systemextramgmtcpu?**args=nodeid:<Double_value>**

Use this query-parameter to get systemextramgmtcpu resources based on additional properties.

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "systemextramgmtcpu": [ {

nodeid:<Double_value>"configuredstate":<String_value>,

"effectivestate":<String_value>

}]}
```

systemfile

Configuration for file resource.

Properties

(click to see Operations)

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
filename	<String>	Read-write	Name of the file. It should not include filepath.

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
			Maximum length = 63
filecontent	<String>	Read-write	file content in Base64 format.
filelocation	<String>	Read-write	location of the file on Citrix ADC. Maximum length = 127
fileencoding	<String>	Read-write	encoding type of the file content. Default value: "BASE64"
fileaccesstime	<String>	Read-only	Last access time of the file.
filemodifiedtime	<String>	Read-only	last modified time of the file.
filemode	<String[]>	Read-only	file mode. Possible values = DIRECTORY
filesize	<Double>	Read-only	Size of the file in BYTES.
__count	<Double>	Read-only	count parameter

Operations

(click to see Properties)

ADD| DELETE| GET (ALL)

Some options that you can use for each operations:

- **Getting warnings in response:**NITRO allows you to get warnings in an operation by specifying the "warning" query parameter as "yes". For example, to get warnings while connecting to the Citrix ADC appliance, the URL is as follows:

  http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/login?warning=yes

  If any, the warnings are displayed in the response payload with the HTTP code "209 X-NITRO-WARNING".

- **Authenticated access for individual NITRO operations:**NITRO allows you to logon to the Citrix ADC appliance to perform individual operations. You can use this option instead of creating a NITRO session (using the login object) and then using that session to perform all operations,

  To do this, you must specify the username and password in the request header of the NITRO request as follows:

  X-NITRO-USER:<username>

  X-NITRO-PASS:<password>

  Note: In such cases, make sure that the request header DOES not include the following:

  Cookie:NITRO_AUTH_TOKEN=<tokenvalue>

`*Note:`*

Mandatory parameters are marked in redand placeholder content is marked in <green>.

add

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/systemfile

HTTP Method:POST

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Content-Type:application/json

Request Payload:

```
{"systemfile":{
```

```
<b>"filename":<String_value>,

</b><b>"filecontent":<String_value>,

</b><b>"filelocation":<String_value>,

</b>"fileencoding":<String_value>

}}
```

Response:

HTTP Status Code on Success: 201 Created HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error

delete

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/systemfile/filename_value<String>

HTTP Method:DELETE

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue>

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error

get (all)

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/systemfile

Query-parameters:

args

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/systemfile?
args=filename:<String_value>,filelocation:<String_value>,

Use this query-parameter to get systemfile resources based on additional properties.

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "systemfile": [ {

filename:<String_value>,<b>filelocation:<String_value>,</
b>"filecontent":<String_value>,

"fileencoding":<String_value>,

"fileaccesstime":<String_value>,

"filemodifiedtime":<String_value>,

"filemode":<String[]_value>,

"filesize":<Double_value>

}]}
```

systemglobaldata

Configuration for global counter data resource.

Properties

(click to see Operations)

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
counters	<String>	Read-write	Specify the counters to be collected.
countergroup	<String>	Read-write	Specify the (counter) group name which contains all the counters specific to this particular group.
starttime	<String>	Read-write	Specify start time in mmddyyyyhhmm to start collecting values from that timestamp.
endtime	<String>	Read-write	Specify end time in mmddyyyyhhmm upto which values have to be collected.
last	<Integer>	Read-write	Last is literal way of saying a certain time period from the current moment. Example: -last 1 hour, -last 1 day, et cetera. Default value: 1
unit	<String>	Read-write	Specify the time period from current moment. Example 1 x where x = hours/ days/ years. Possible values = HOURS, DAYS, MONTHS
datasource	<String>	Read-write	Specifies the source which contains all

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
			the stored counter values.
core	<Integer>	Read-write	Specify core ID of the PE in nCore.
response	<String>	Read-only	.
startupdate	<Double>	Read-only	.
lastupdate	<Double>	Read-only	.

Operations

(click to see Properties)

GET (ALL)

Some options that you can use for each operations:

- **Getting warnings in response:**NITRO allows you to get warnings in an operation by specifying the "warning" query parameter as "yes". For example, to get warnings while connecting to the Citrix ADC appliance, the URL is as follows:

  http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/login?warning=yes

  If any, the warnings are displayed in the response payload with the HTTP code "209 X-NITRO-WARNING".

- **Authenticated access for individual NITRO operations:**NITRO allows you to logon to the Citrix ADC appliance to perform individual operations. You can use this option instead of creating a NITRO session (using the login object) and then using that session to perform all operations,

  To do this, you must specify the username and password in the request header of the NITRO request as follows:

  X-NITRO-USER:<username>

  X-NITRO-PASS:<password>

  Note: In such cases, make sure that the request header DOES not include the following:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue>

`*Note:` *

Mandatory parameters are marked in redand placeholder content is marked in <green>.

get (all)

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/systemglobaldata

Query-parameters:

args

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/systemglobaldata?
args=counters:<String_value>,countergroup:<String_value>,starttime:<String_value>,endtime:<String_

Use this query-parameter to get systemglobaldata resources based on additional properties.

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "systemglobaldata": [ {

<b>counters:<String_value>,</
b>countergroup:<String_value>,starttime:<String_value>,endtime:<String_val
ue>,last:<Integer_value>,unit:<String_value>,datasource:<String_value>,cor
e:<Integer_value>"response":<String_value>,

"startupdate":<Double_value>,
```

```
"lastupdate":<Double_value>

}]}
```

systemglobal_auditnslogpolicy_binding

Binding object showing the auditnslogpolicy that can be bound to systemglobal.

Properties

(click to see Operations)

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
priority	<Double>	Read-write	The priority of the command policy.
globalbindtype	<String>	Read-write	. Default value: SYSTEM_GLOBAL Possible values = SYSTEM_GLOBAL, VPN_GLOBAL, RNAT_GLOBAL
nextfactor	<String>	Read-write	On success invoke label. Applicable for advanced authentication policy binding.
gotopriorityexpression	<String>	Read-write	Applicable only to advance authentication policy. Expression or other value specifying the next policy to be evaluated if the current policy evaluates to TRUE.

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
			Specify one of the following values: * NEXT - Evaluate the policy with the next higher priority number. * END - End policy evaluation.
policyname	<String>	Read-write	The name of the command policy.
__count	<Double>	Read-write	count parameter

Operations

(click to see Properties)

ADD:| DELETE:| GET| GET (ALL)| COUNT

Some options that you can use for each operations:

- **Getting warnings in response:**NITRO allows you to get warnings in an operation by specifying the "warning" query parameter as "yes". For example, to get warnings while connecting to the Citrix ADC appliance, the URL is as follows:

  http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/login?warning=yes

  If any, the warnings are displayed in the response payload with the HTTP code "209 X-NITRO-WARNING".

- **Authenticated access for individual NITRO operations:**NITRO allows you to logon to the Citrix ADC appliance to perform individual operations. You can use this option instead of creating a NITRO session (using the login object) and then using that session to perform all operations,

  To do this, you must specify the username and password in the request header of the NITRO request as follows:

  X-NITRO-USER:<username>

  X-NITRO-PASS:<password>

  Note: In such cases, make sure that the request header DOES not include the following:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue>

`*Note:` *

Mandatory parameters are marked in redand placeholder content is marked in <green>.

add

URL:http://<Citrix-ADC-IP-address(NSIP)/nitro/v1/config/systemglobal_auditnslogpolicy_binding

HTTP Method:PUT

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Content-Type:application/json

Request Payload:

```
{
"systemglobal_auditnslogpolicy_binding":{
"policyname":<String_value>,
"priority":<Double_value>,
"nextfactor":<String_value>,
"gotopriorityexpression":<String_value>
}}
```

Response:

HTTP Status Code on Success: 201 Created HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error

delete

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/systemglobal_auditnslogpolicy_binding

Query-parameters:

args

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/systemglobal_auditnslogpolicy_binding?**args=policyname:<String_value>**

HTTP Method:DELETE

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue>

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error

get

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/systemglobal_auditnslogpolicy_binding

Query-parameters:

filter

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/systemglobal_auditnslogpolicy_binding?**filter=property-name1:property-value1,property-name2:property-value2**

Use this query-parameter to get the filtered set of systemglobal_auditnslogpolicy_binding resources configured on Citrix ADC. Filtering can be done on any of the properties of the resource.

pagination

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/systemglobal_auditnslogpolicy_binding?**pagesize=#no;pageno=#no**

Use this query-parameter to get the systemglobal_auditnslogpolicy_binding resources in chunks.

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "systemglobal_auditnslogpolicy_binding": [ {

"priority":<Double_value>,

"globalbindtype":<String_value>,

"nextfactor":<String_value>,

"gotopriorityexpression":<String_value>,

"policyname":<String_value>

}]}
```

get (all)

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/systemglobal_auditnslogpolicy_binding

Query-parameters:

bulkbindings

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/systemglobal_auditnslogpolicy_binding?**bulkbindings=yes**

NITRO allows you to fetch bindings in bulk.

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "systemglobal_auditnslogpolicy_binding": [ {

"priority":<Double_value>,

"globalbindtype":<String_value>,

"nextfactor":<String_value>,

"gotopriorityexpression":<String_value>,

"policyname":<String_value>

}]}
```

count

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/systemglobal_auditnslogpolicy_binding?**count=yes**

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "systemglobal_auditnslogpolicy_binding": [ { "__count": "#no"} ] }
```

systemglobal_auditsyslogpolicy_binding

Binding object showing the auditsyslogpolicy that can be bound to systemglobal.

Properties

(click to see Operations)

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
priority	<Double>	Read-write	The priority of the command policy.
globalbindtype	<String>	Read-write	. Default value: SYSTEM_GLOBAL Possible values = SYSTEM_GLOBAL, VPN_GLOBAL, RNAT_GLOBAL
nextfactor	<String>	Read-write	On success invoke label. Applicable for advanced authentication policy binding.

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
gotopriorityexpression	<String>	Read-write	Applicable only to advance authentication policy. Expression or other value specifying the next policy to be evaluated if the current policy evaluates to TRUE. Specify one of the following values: * NEXT - Evaluate the policy with the next higher priority number. * END - End policy evaluation.
policyname	<String>	Read-write	The name of the command policy.
__count	<Double>	Read-write	count parameter

Operations

(click to see Properties)

ADD:| DELETE:| GET| GET (ALL)| COUNT

Some options that you can use for each operations:

- **Getting warnings in response:**NITRO allows you to get warnings in an operation by specifying the "warning" query parameter as "yes". For example, to get warnings while connecting to the Citrix ADC appliance, the URL is as follows:

  http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/login?warning=yes

  If any, the warnings are displayed in the response payload with the HTTP code "209 X-NITRO-WARNING".
- **Authenticated access for individual NITRO operations:**NITRO allows you to logon to the Citrix ADC appliance to perform individual operations. You can use this option instead of

creating a NITRO session (using the login object) and then using that session to perform all operations,

To do this, you must specify the username and password in the request header of the NITRO request as follows:

X-NITRO-USER:<username>

X-NITRO-PASS:<password>

Note: In such cases, make sure that the request header DOES not include the following:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue>

`*Note:`*

Mandatory parameters are marked in redand placeholder content is marked in <green>.

add

URL:http://<Citrix-ADC-IP-address(NSIP)/nitro/v1/config/systemglobal_auditsyslogpolicy_binding

HTTP Method:PUT

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Content-Type:application/json

Request Payload:

```
{

"systemglobal_auditsyslogpolicy_binding":{

"policyname":<String_value>,

"priority":<Double_value>,

"nextfactor":<String_value>,

"gotopriorityexpression":<String_value>

}}
```

Response:

HTTP Status Code on Success: 201 Created HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error

delete

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/systemglobal_auditsyslogpolicy_binding

Query-parameters:

args

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/systemglobal_auditsyslogpolicy_binding?**args=policyname:<String_value>**

HTTP Method:DELETE

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue>

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error

get

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/systemglobal_auditsyslogpolicy_binding

Query-parameters:

filter

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/systemglobal_auditsyslogpolicy_binding?**filter=property-name1:property-value1,property-name2:property-value2**

Use this query-parameter to get the filtered set of systemglobal_auditsyslogpolicy_binding resources configured on Citrix ADC. Filtering can be done on any of the properties of the resource.

pagination

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/systemglobal_auditsyslogpolicy_binding?**pagesize=#no;pageno=#no**

Use this query-parameter to get the systemglobal_auditsyslogpolicy_binding resources in chunks.

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "systemglobal_auditsyslogpolicy_binding": [ {

"priority":<Double_value>,

"globalbindtype":<String_value>,

"nextfactor":<String_value>,

"gotopriorityexpression":<String_value>,

"policyname":<String_value>

}]}
```

get (all)

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/systemglobal_auditsyslogpolicy_binding

Query-parameters:

bulkbindings

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/systemglobal_auditsyslogpolicy_binding?**bulkbindings=yes**

NITRO allows you to fetch bindings in bulk.

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "systemglobal_auditsyslogpolicy_binding": [ {

"priority":<Double_value>,

"globalbindtype":<String_value>,

"nextfactor":<String_value>,

"gotopriorityexpression":<String_value>,

"policyname":<String_value>

}]}
```

count

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/systemglobal_auditsyslogpolicy_binding?**count=yes**

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "systemglobal_auditsyslogpolicy_binding": [ { "__count": "#no"} ] }
```

systemglobal_authenticationldappolicy_binding

Binding object showing the authenticationldappolicy that can be bound to systemglobal.

Properties

(click to see Operations)

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
priority	<Double>	Read-write	The priority of the command policy.
globalbindtype	<String>	Read-write	. Default value: SYSTEM_GLOBAL Possible values = SYSTEM_GLOBAL, VPN_GLOBAL, RNAT_GLOBAL

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
nextfactor	<String>	Read-write	On success invoke label. Applicable for advanced authentication policy binding.
gotopriorityexpression	<String>	Read-write	Applicable only to advance authentication policy. Expression or other value specifying the next policy to be evaluated if the current policy evaluates to TRUE. Specify one of the following values: * NEXT - Evaluate the policy with the next higher priority number. * END - End policy evaluation.
policyname	<String>	Read-write	The name of the command policy.
__count	<Double>	Read-write	count parameter

Operations

(click to see Properties)

ADD:| DELETE:| GET| GET (ALL)| COUNT

Some options that you can use for each operations:

- **Getting warnings in response:**NITRO allows you to get warnings in an operation by specifying the "warning" query parameter as "yes". For example, to get warnings while connecting to the Citrix ADC appliance, the URL is as follows:

  http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/login?warning=yes

If any, the warnings are displayed in the response payload with the HTTP code "209 X-NITRO-WARNING".

- **Authenticated access for individual NITRO operations:**NITRO allows you to logon to the Citrix ADC appliance to perform individual operations. You can use this option instead of creating a NITRO session (using the login object) and then using that session to perform all operations,

  To do this, you must specify the username and password in the request header of the NITRO request as follows:

  X-NITRO-USER:<username>

  X-NITRO-PASS:<password>

  Note: In such cases, make sure that the request header DOES not include the following:

  Cookie:NITRO_AUTH_TOKEN=<tokenvalue>

`*Note:`*

Mandatory parameters are marked in redand placeholder content is marked in <green>.

add

URL:http://<Citrix-ADC-IP-address(NSIP)/nitro/v1/config/systemglobal_authenticationldappolicy_binding

HTTP Method:PUT

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Content-Type:application/json

Request Payload:

```
{

"systemglobal_authenticationldappolicy_binding":{

"policyname":<String_value>,

"priority":<Double_value>,
```

```
"nextfactor":<String_value>,

"gotopriorityexpression":<String_value>

}}
```

Response:

HTTP Status Code on Success: 201 Created HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error

delete

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/systemglobal_authenticationldappolicy_binding

Query-parameters:

args

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/systemglobal_authenticationldappolicy_binding?**args=policyname:<String_value>**

HTTP Method:DELETE

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue>

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error

get

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/systemglobal_authenticationldappolicy_binding

Query-parameters:

filter

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/systemglobal_authenticationldappolicy_binding?**filter=property-name1:property-value1,property-name2:property-value2**

Use this query-parameter to get the filtered set of systemglobal_authenticationldappolicy_binding resources configured on Citrix ADC. Filtering can be done on any of the properties of the resource.

pagination

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/systemglobal_authenticationldappolicy_binding?**pagesize=#no;pageno=#no**

Use this query-parameter to get the systemglobal_authenticationldappolicy_binding resources in chunks.

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "systemglobal_authenticationldappolicy_binding": [ {

"priority":<Double_value>,

"globalbindtype":<String_value>,

"nextfactor":<String_value>,

"gotopriorityexpression":<String_value>,
```

```
"policyname":<String_value>

}]}
```

get (all)

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/systemglobal_authenticationldappolicy_binding

Query-parameters:

bulkbindings

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/systemglobal_authenticationldappolicy_binding?**bulkbindings=yes**

NITRO allows you to fetch bindings in bulk.

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "systemglobal_authenticationldappolicy_binding": [ {

"priority":<Double_value>,

"globalbindtype":<String_value>,

"nextfactor":<String_value>,
```

```
"gotopriorityexpression":<String_value>,

"policyname":<String_value>

}]}
```

count

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/systemglobal_authenticationldappolicy_binding?**count=yes**

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "systemglobal_authenticationldappolicy_binding": [ { "__count":
"#no"} ] }
```

systemglobal_authenticationlocalpolicy_binding

Binding object showing the authenticationlocalpolicy that can be bound to systemglobal.

Properties

(click to see Operations)

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
priority	<Double>	Read-write	The priority of the command policy.
globalbindtype	<String>	Read-write	. Default value: SYSTEM_GLOBAL Possible values = SYSTEM_GLOBAL, VPN_GLOBAL, RNAT_GLOBAL
nextfactor	<String>	Read-write	On success invoke label. Applicable for advanced authentication policy binding.
gotopriorityexpression	<String>	Read-write	Applicable only to advance authentication policy. Expression or other value specifying the next policy to be evaluated if the current policy evaluates to TRUE. Specify one of the following values: * NEXT - Evaluate the policy with the next higher priority number. * END - End policy evaluation.
policyname	<String>	Read-write	The name of the command policy.
__count	<Double>	Read-write	count parameter

Operations

(click to see Properties)

ADD:| DELETE:| GET| GET (ALL)| COUNT

Some options that you can use for each operations:

- **Getting warnings in response:**NITRO allows you to get warnings in an operation by specifying the "warning" query parameter as "yes". For example, to get warnings while connecting to the Citrix ADC appliance, the URL is as follows:

  http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/login?warning=yes

  If any, the warnings are displayed in the response payload with the HTTP code "209 X-NITRO-WARNING".

- **Authenticated access for individual NITRO operations:**NITRO allows you to logon to the Citrix ADC appliance to perform individual operations. You can use this option instead of creating a NITRO session (using the login object) and then using that session to perform all operations,

  To do this, you must specify the username and password in the request header of the NITRO request as follows:

  X-NITRO-USER:<username>

  X-NITRO-PASS:<password>

  Note: In such cases, make sure that the request header DOES not include the following:

  Cookie:NITRO_AUTH_TOKEN=<tokenvalue>

`*Note:`*

Mandatory parameters are marked in redand placeholder content is marked in <green>.

add

URL:http://<Citrix-ADC-IP-address(NSIP)/nitro/v1/config/systemglobal_authenticationlocalpolicy_binding

HTTP Method:PUT

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Content-Type:application/json

Request Payload:

```
{
"systemglobal_authenticationlocalpolicy_binding":{
"policyname":<String_value>,
"priority":<Double_value>,
"nextfactor":<String_value>,
"gotopriorityexpression":<String_value>
}}
```

Response:

HTTP Status Code on Success: 201 Created HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error

delete

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/systemglobal_authenticationlocalpolicy_binding

Query-parameters:

args

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/systemglobal_authenticationlocalpolicy_binding?**args=policyname:<String_value>**

HTTP Method:DELETE

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue>

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error

get

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/systemglobal_authenticationlocalpolicy_binding

Query-parameters:

filter

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/systemglobal_authenticationlocalpolicy_binding?**filter=property-name1:property-value1,property-name2:property-value2**

Use this query-parameter to get the filtered set of systemglobal_authenticationlocalpolicy_binding resources configured on Citrix ADC. Filtering can be done on any of the properties of the resource.

pagination

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/systemglobal_authenticationlocalpolicy_binding?**pagesize=#no;pageno=#no**

Use this query-parameter to get the systemglobal_authenticationlocalpolicy_binding resources in chunks.

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "systemglobal_authenticationlocalpolicy_binding": [ {

"priority":<Double_value>,

"globalbindtype":<String_value>,

"nextfactor":<String_value>,

"gotopriorityexpression":<String_value>,

"policyname":<String_value>

}]}
```

get (all)

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/systemglobal_authenticationlocalpolicy_binding

Query-parameters:

bulkbindings

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/systemglobal_authenticationlocalpolicy_binding?**bulkbindings=yes**

NITRO allows you to fetch bindings in bulk.

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "systemglobal_authenticationlocalpolicy_binding": [ {

"priority":<Double_value>,

"globalbindtype":<String_value>,

"nextfactor":<String_value>,

"gotopriorityexpression":<String_value>,

"policyname":<String_value>

}]}
```

count

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/systemglobal_authenticationlocalpolicy_binding?**count=yes**

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "systemglobal_authenticationlocalpolicy_binding": [ { "__count":
"#no"} ] }
```

systemglobal_authenticationpolicy_binding

Binding object showing the authenticationpolicy that can be bound to systemglobal.

Properties

(click to see Operations)

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
priority	<Double>	Read-write	The priority of the command policy.
globalbindtype	<String>	Read-write	. Default value: SYSTEM_GLOBAL Possible values = SYSTEM_GLOBAL, VPN_GLOBAL, RNAT_GLOBAL
nextfactor	<String>	Read-write	On success invoke label. Applicable for advanced authentication policy binding.
gotopriorityexpression	<String>	Read-write	Applicable only to advance authentication policy. Expression or other value specifying the next policy to be evaluated if the current policy evaluates to TRUE. Specify one of the following values: * NEXT - Evaluate the policy with the next

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
			higher priority number. * END - End policy evaluation.
policyname	<String>	Read-write	The name of the command policy.
__count	<Double>	Read-write	count parameter

Operations

(click to see Properties)

ADD:| DELETE:| GET| GET (ALL)| COUNT

Some options that you can use for each operations:

- **Getting warnings in response:**NITRO allows you to get warnings in an operation by specifying the "warning" query parameter as "yes". For example, to get warnings while connecting to the Citrix ADC appliance, the URL is as follows:

  http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/login?warning=yes

  If any, the warnings are displayed in the response payload with the HTTP code "209 X-NITRO-WARNING".

- **Authenticated access for individual NITRO operations:**NITRO allows you to logon to the Citrix ADC appliance to perform individual operations. You can use this option instead of creating a NITRO session (using the login object) and then using that session to perform all operations,

  To do this, you must specify the username and password in the request header of the NITRO request as follows:

  X-NITRO-USER:<username>

  X-NITRO-PASS:<password>

  Note: In such cases, make sure that the request header DOES not include the following:

  Cookie:NITRO_AUTH_TOKEN=<tokenvalue>

`*Note:` *

Mandatory parameters are marked in redand placeholder content is marked in <green>.

add

URL:http://<Citrix-ADC-IP-address(NSIP)/nitro/v1/config/systemglobal_authenticationpolicy_binding

HTTP Method:PUT

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Content-Type:application/json

Request Payload:

```
{

"systemglobal_authenticationpolicy_binding":{

"policyname":<String_value>,

"priority":<Double_value>,

"nextfactor":<String_value>,

"gotopriorityexpression":<String_value>

}}
```

Response:

HTTP Status Code on Success: 201 Created HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error

delete

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/systemglobal_authenticationpolicy_binding

Query-parameters:

args

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/systemglobal_authenticationpolicy_binding?**args=policyname:<String_value>**

HTTP Method:DELETE

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue>

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error

get

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/systemglobal_authenticationpolicy_binding

Query-parameters:

filter

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/systemglobal_authenticationpolicy_binding?**filter=property-name1:property-value1,property-name2:property-value2**

Use this query-parameter to get the filtered set of systemglobal_authenticationpolicy_binding resources configured on Citrix ADC. Filtering can be done on any of the properties of the resource.

pagination

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/systemglobal_authenticationpolicy_binding?**pagesize=#no;pageno=#no**

Use this query-parameter to get the systemglobal_authenticationpolicy_binding resources in chunks.

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "systemglobal_authenticationpolicy_binding": [ {

"priority":<Double_value>,

"globalbindtype":<String_value>,

"nextfactor":<String_value>,

"gotopriorityexpression":<String_value>,

"policyname":<String_value>

}]}
```

get (all)

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/systemglobal_authenticationpolicy_binding

Query-parameters:

bulkbindings

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/systemglobal_authenticationpolicy_binding?**bulkbindings=yes**

NITRO allows you to fetch bindings in bulk.

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "systemglobal_authenticationpolicy_binding": [ {

"priority":<Double_value>,

"globalbindtype":<String_value>,

"nextfactor":<String_value>,

"gotopriorityexpression":<String_value>,

"policyname":<String_value>

}]}
```

count

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/systemglobal_authenticationpolicy_binding?**count=yes**

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "systemglobal_authenticationpolicy_binding": [ { "__count": "#no"} ] }
```

systemglobal_authenticationradiuspolicy_binding

Binding object showing the authenticationradiuspolicy that can be bound to systemglobal.

Properties

(click to see Operations)

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
priority	<Double>	Read-write	The priority of the command policy.
globalbindtype	<String>	Read-write	. Default value: SYSTEM_GLOBAL Possible values = SYSTEM_GLOBAL, VPN_GLOBAL, RNAT_GLOBAL
nextfactor	<String>	Read-write	On success invoke label. Applicable for advanced authentication policy binding.

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
gotopriorityexpression	<String>	Read-write	Applicable only to advance authentication policy. Expression or other value specifying the next policy to be evaluated if the current policy evaluates to TRUE. Specify one of the following values: * NEXT - Evaluate the policy with the next higher priority number. * END - End policy evaluation.
policyname	<String>	Read-write	The name of the command policy.
__count	<Double>	Read-write	count parameter

Operations

(click to see Properties)

ADD:| DELETE:| GET| GET (ALL)| COUNT

Some options that you can use for each operations:

- **Getting warnings in response:**NITRO allows you to get warnings in an operation by specifying the "warning" query parameter as "yes". For example, to get warnings while connecting to the Citrix ADC appliance, the URL is as follows:

  http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/login?warning=yes

  If any, the warnings are displayed in the response payload with the HTTP code "209 X-NITRO-WARNING".
- **Authenticated access for individual NITRO operations:**NITRO allows you to logon to the Citrix ADC appliance to perform individual operations. You can use this option instead of

creating a NITRO session (using the login object) and then using that session to perform all operations,

To do this, you must specify the username and password in the request header of the NITRO request as follows:

X-NITRO-USER:<username>

X-NITRO-PASS:<password>

Note: In such cases, make sure that the request header DOES not include the following:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue>

`*Note:`*

Mandatory parameters are marked in redand placeholder content is marked in <green>.

add

URL:http://<Citrix-ADC-IP-address(NSIP)/nitro/v1/config/systemglobal_authenticationradiuspolicy_binding

HTTP Method:PUT

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Content-Type:application/json

Request Payload:

```
{

"systemglobal_authenticationradiuspolicy_binding":{

"policyname":<String_value>,

"priority":<Double_value>,

"nextfactor":<String_value>,

"gotopriorityexpression":<String_value>
```

```
}}
```

Response:

HTTP Status Code on Success: 201 Created HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error

delete

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/systemglobal_authenticationradiuspolicy_binding

Query-parameters:

args

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/systemglobal_authenticationradiuspolicy_binding?**args=policyname:<String_value>**

HTTP Method:DELETE

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue>

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error

get

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/systemglobal_authenticationradiuspolicy_binding

Query-parameters:

filter

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/systemglobal_authenticationradiuspolicy_binding?**filter=property-name1:property-value1,property-name2:property-value2**

Use this query-parameter to get the filtered set of systemglobal_authenticationradiuspolicy_binding resources configured on Citrix ADC. Filtering can be done on any of the properties of the resource.

pagination

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/systemglobal_authenticationradiuspolicy_binding?**pagesize=#no;pageno=#no**

Use this query-parameter to get the systemglobal_authenticationradiuspolicy_binding resources in chunks.

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "systemglobal_authenticationradiuspolicy_binding": [ {

"priority":<Double_value>,

"globalbindtype":<String_value>,

"nextfactor":<String_value>,

"gotopriorityexpression":<String_value>,

"policyname":<String_value>
```

```
}]}
```

get (all)

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/systemglobal_authenticationradiuspolicy_binding

Query-parameters:

bulkbindings

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/systemglobal_authenticationradiuspolicy_binding?**bulkbindings=yes**

NITRO allows you to fetch bindings in bulk.

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "systemglobal_authenticationradiuspolicy_binding": [ {

"priority":<Double_value>,

"globalbindtype":<String_value>,

"nextfactor":<String_value>,

"gotopriorityexpression":<String_value>,
```

```
"policyname":<String_value>

}]}
```

count

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/systemglobal_authenticationradiuspolicy_binding?**count=yes**

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "systemglobal_authenticationradiuspolicy_binding": [ { "__count":
"#no"} ] }
```

systemglobal_authenticationtacacspolicy_binding

Binding object showing the authenticationtacacspolicy that can be bound to systemglobal.

Properties

(click to see Operations)

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
priority	<Double>	Read-write	The priority of the command policy.
globalbindtype	<String>	Read-write	. Default value: SYSTEM_GLOBAL Possible values = SYSTEM_GLOBAL, VPN_GLOBAL, RNAT_GLOBAL
nextfactor	<String>	Read-write	On success invoke label. Applicable for advanced authentication policy binding.
gotopriorityexpression	<String>	Read-write	Applicable only to advance authentication policy. Expression or other value specifying the next policy to be evaluated if the current policy evaluates to TRUE. Specify one of the following values: * NEXT - Evaluate the policy with the next higher priority number. * END - End policy evaluation.
policyname	<String>	Read-write	The name of the command policy.
__count	<Double>	Read-write	count parameter

Operations

(click to see Properties)

ADD:| DELETE:| GET| GET (ALL)| COUNT

Some options that you can use for each operations:

- **Getting warnings in response:**NITRO allows you to get warnings in an operation by specifying the "warning" query parameter as "yes". For example, to get warnings while connecting to the Citrix ADC appliance, the URL is as follows:

  http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/login?warning=yes

  If any, the warnings are displayed in the response payload with the HTTP code "209 X-NITRO-WARNING".

- **Authenticated access for individual NITRO operations:**NITRO allows you to logon to the Citrix ADC appliance to perform individual operations. You can use this option instead of creating a NITRO session (using the login object) and then using that session to perform all operations,

  To do this, you must specify the username and password in the request header of the NITRO request as follows:

  X-NITRO-USER:<username>

  X-NITRO-PASS:<password>

  Note: In such cases, make sure that the request header DOES not include the following:

  Cookie:NITRO_AUTH_TOKEN=<tokenvalue>

`*Note:`*

Mandatory parameters are marked in redand placeholder content is marked in <green>.

add

URL:http://<Citrix-ADC-IP-address(NSIP)/nitro/v1/config/systemglobal_authenticationtacacspolicy_binding

HTTP Method:PUT

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Content-Type:application/json

Request Payload:

```
{
"systemglobal_authenticationtacacspolicy_binding":{
"policyname":<String_value>,
"priority":<Double_value>,
"nextfactor":<String_value>,
"gotopriorityexpression":<String_value>
}}
```

Response:

HTTP Status Code on Success: 201 Created HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error

delete

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/systemglobal_authenticationtacacspolicy_binding

Query-parameters:

args

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/systemglobal_authenticationtacacspolicy_binding?**args=policyname:<String_value>**

HTTP Method:DELETE

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue>

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error

get

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/systemglobal_authenticationtacacspolicy_binding

Query-parameters:

filter

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/systemglobal_authenticationtacacspolicy_binding?**filter=property-name1:property-value1,property-name2:property-value2**

Use this query-parameter to get the filtered set of systemglobal_authenticationtacacspolicy_binding resources configured on Citrix ADC. Filtering can be done on any of the properties of the resource.

pagination

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/systemglobal_authenticationtacacspolicy_binding?**pagesize=#no;pageno=#no**

Use this query-parameter to get the systemglobal_authenticationtacacspolicy_binding resources in chunks.

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "systemglobal_authenticationtacacspolicy_binding": [ {

"priority":<Double_value>,

"globalbindtype":<String_value>,

"nextfactor":<String_value>,

"gotopriorityexpression":<String_value>,

"policyname":<String_value>

}]}
```

get (all)

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/systemglobal_authenticationtacacspolicy_binding

Query-parameters:

bulkbindings

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/systemglobal_authenticationtacacspolicy_binding?**bulkbindings=yes**

NITRO allows you to fetch bindings in bulk.

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "systemglobal_authenticationtacacspolicy_binding": [ {

"priority":<Double_value>,

"globalbindtype":<String_value>,

"nextfactor":<String_value>,

"gotopriorityexpression":<String_value>,

"policyname":<String_value>

}]}
```

count

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/systemglobal_authenticationtacacspolicy_binding?**count=yes**

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "systemglobal_authenticationtacacspolicy_binding": [ { "__count":
"#no"} ] }
```

systemglobal_binding

Binding object which returns the resources bound to systemglobal.

Properties

(click to see Operations)

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
systemglobal_authenticationradiuspolicy_binding	<systemglobal_authenticationradiuspolicy_binding[]>	Read-only	authenticationradiuspolicy that can be bound to systemglobal.
systemglobal_authenticationldappolicy_binding	<systemglobal_authenticationldappolicy_binding[]>	Read-only	authenticationldappolicy that can be bound to systemglobal.
systemglobal_authenticationlocalpolicy_binding	<systemglobal_authenticationlocalpolicy_binding[]>	Read-only	authenticationlocalpolicy that can be bound to systemglobal.
systemglobal_auditsyslogpolicy_binding	<systemglobal_auditsyslogpolicy_binding[]>	Read-only	auditsyslogpolicy that can be bound to systemglobal.
systemglobal_auditnslogpolicy_binding	<systemglobal_auditnslogpolicy_binding[]>	Read-only	auditnslogpolicy that can be bound to systemglobal.
systemglobal_authenticationpolicy_binding	<systemglobal_authenticationpolicy_binding[]>	Read-only	authenticationpolicy that can be bound to systemglobal.

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
systemglobal_authenticationtacacspolicy_binding	<systemglobal_authenticationtacacspolicy_binding[]>	Read-only	authenticationtacacs policy that can be bound to systemglobal.

Operations

(click to see Properties)

GET| GET (ALL)

Some options that you can use for each operations:

- **Getting warnings in response:**NITRO allows you to get warnings in an operation by specifying the "warning" query parameter as "yes". For example, to get warnings while connecting to the Citrix ADC appliance, the URL is as follows:

  http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/login?warning=yes

  If any, the warnings are displayed in the response payload with the HTTP code "209 X-NITRO-WARNING".

- **Authenticated access for individual NITRO operations:**NITRO allows you to logon to the Citrix ADC appliance to perform individual operations. You can use this option instead of creating a NITRO session (using the login object) and then using that session to perform all operations,

  To do this, you must specify the username and password in the request header of the NITRO request as follows:

  X-NITRO-USER:<username>

  X-NITRO-PASS:<password>

  Note: In such cases, make sure that the request header DOES not include the following:

  Cookie:NITRO_AUTH_TOKEN=<tokenvalue>

`*Note:`*

Mandatory parameters are marked in redand placeholder content is marked in <green>.

get

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/systemglobal_binding

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "systemglobal_binding": [ {

"systemglobal_authenticationldappolicy_binding":<systemglobal_authenticati
onldappolicy_binding[]_value>,

"systemglobal_auditsyslogpolicy_binding":<systemglobal_auditsyslogpolicy_b
inding[]_value>,

"systemglobal_authenticationlocalpolicy_binding":<systemglobal_authenticat
ionlocalpolicy_binding[]_value>,

"systemglobal_auditnslogpolicy_binding":<systemglobal_auditnslogpolicy_bin
ding[]_value>,

"systemglobal_authenticationtacacspolicy_binding":<systemglobal_authentica
tiontacacspolicy_binding[]_value>,

"systemglobal_authenticationpolicy_binding":<systemglobal_authenticationpo
licy_binding[]_value>,

"systemglobal_authenticationradiuspolicy_binding":<systemglobal_authentica
tionradiuspolicy_binding[]_value>
```

```
}]}
```

get (all)

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/systemglobal_binding

Query-parameters:

bulkbindings

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/systemglobal_binding?**bulkbindings=yes**

NITRO allows you to fetch bindings in bulk.

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "systemglobal_binding": [ {

"systemglobal_authenticationldappolicy_binding":<systemglobal_authenticati
onldappolicy_binding[]_value>,

"systemglobal_auditsyslogpolicy_binding":<systemglobal_auditsyslogpolicy_b
inding[]_value>,

"systemglobal_authenticationlocalpolicy_binding":<systemglobal_authenticat
ionlocalpolicy_binding[]_value>,
```

```
"systemglobal_auditnslogpolicy_binding":<systemglobal_auditnslogpolicy_binding[]_value>,

"systemglobal_authenticationtacacspolicy_binding":<systemglobal_authenticationtacacspolicy_binding[]_value>,

"systemglobal_authenticationpolicy_binding":<systemglobal_authenticationpolicy_binding[]_value>,

"systemglobal_authenticationradiuspolicy_binding":<systemglobal_authenticationradiuspolicy_binding[]_value>

}]}
```

systemgroup

Configuration for system group resource.

Properties

(click to see Operations)

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
groupname	<String>	Read-write	Name for the group. Must begin with a letter, number, or the underscore (_) character, and must contain only alphanumeric, hyphen (-), period (.), hash (#), space ( ), at (@), equal (=), colon (:), and underscore characters. Cannot be changed after the group is created.

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
			CLI Users: If the name includes one or more spaces, enclose the name in double or single quotation marks (for example, "my group" or 'my group'). Minimum length = 1
promptstring	<String>	Read-write	String to display at the command-line prompt. Can consist of letters, numbers, hyphen (-), period (.), hash (#), space ( ), at (@), equal (=), colon (:), underscore (_), and the following variables: * %u - Will be replaced by the user name. * %h - Will be replaced by the hostname of the Citrix ADC. * %t - Will be replaced by the current time in 12-hour format. * %T - Will be replaced by the

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
			current time in 24-hour format. * %d - Will be replaced by the current date. * %s - Will be replaced by the state of the Citrix ADC. Note: The 63-character limit for the length of the string does not apply to the characters that replace the variables. Minimum length = 1
timeout	<Double>	Read-write	CLI session inactivity timeout, in seconds. If Restrictedtimeout argument of system parameter is enabled, Timeout can have values in the range [300-86400] seconds.If Restrictedtimeout argument of system parameter is disabled, Timeout can have values in the range [0, 10-100000000]

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
			seconds. Default value is 900 seconds.
__count	<Double>	Read-only	count parameter

Operations

(click to see Properties)

ADD| DELETE| UPDATE| UNSET| GET (ALL)| GET| COUNT

Some options that you can use for each operations:

- **Getting warnings in response:**NITRO allows you to get warnings in an operation by specifying the "warning" query parameter as "yes". For example, to get warnings while connecting to the Citrix ADC appliance, the URL is as follows:

  http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/login?warning=yes

  If any, the warnings are displayed in the response payload with the HTTP code "209 X-NITRO-WARNING".

- **Authenticated access for individual NITRO operations:**NITRO allows you to logon to the Citrix ADC appliance to perform individual operations. You can use this option instead of creating a NITRO session (using the login object) and then using that session to perform all operations,

  To do this, you must specify the username and password in the request header of the NITRO request as follows:

  X-NITRO-USER:<username>

  X-NITRO-PASS:<password>

  Note: In such cases, make sure that the request header DOES not include the following:

  Cookie:NITRO_AUTH_TOKEN=<tokenvalue>

`*Note:`*

Mandatory parameters are marked in redand placeholder content is marked in <green>.

add

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/systemgroup

HTTP Method:POST

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Content-Type:application/json

Request Payload:

```
{"systemgroup":{

<b>"groupname":<String_value>,

</b>"promptstring":<String_value>,

"timeout":<Double_value>

}}
```

Response:

HTTP Status Code on Success: 201 Created HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error

delete

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/systemgroup/groupname_value<String>

HTTP Method:DELETE

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue>

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error

update

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/systemgroup

HTTP Method:PUT

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Content-Type:application/json

Request Payload:

```
{"systemgroup":{

<b>"groupname":<String_value>,

</b>"promptstring":<String_value>,

"timeout":<Double_value>

}}
```

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error

unset

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/systemgroup?**action=unset**

HTTP Method:POST

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Content-Type:application/json

Request Payload:

```
{"systemgroup":{

<b>"groupname":<String_value>,

</b>"promptstring":true,

"timeout":true

}}
```

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error

get (all)

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/systemgroup

Query-parameters:

attrs

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/systemgroup?**attrs=property-name1,property-name2**

Use this query parameter to specify the resource details that you want to retrieve.

filter

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/systemgroup?**filter=property-name1:property-val1,property-name2:property-val2**

Use this query-parameter to get the filtered set of systemgroup resources configured on Citrix ADC. Filtering can be done on any of the properties of the resource.

view

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/systemgroup?**view=summary**

Note: By default, the retrieved results are displayed in detail view (?view=detail).

pagination

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/systemgroup?**pagesize=#no;pageno=#no**

Use this query-parameter to get the systemgroup resources in chunks.

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "systemgroup": [ {

"groupname":<String_value>,

"promptstring":<String_value>,

"timeout":<Double_value>

}]}
```

get

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/systemgroup/groupname_value<String>

Query-parameters:

attrs

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/systemgroup/groupname_value<String>?**attrs=property-name1,property-name2**

Use this query parameter to specify the resource details that you want to retrieve.

view

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/systemgroup/groupname_value<String>?**view=summary**

Note: By default, the retrieved results are displayed in detail view (?view=detail).

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "systemgroup": [ {

"groupname":<String_value>,

"promptstring":<String_value>,

"timeout":<Double_value>

}]}
```

count

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/systemgroup?**count=yes**

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "systemgroup": [ { "__count": "#no"} ] }
```

systemgroup_binding

Binding object which returns the resources bound to systemgroup.

Properties

(click to see Operations)

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
groupname	<String>	Read-write	Name of the system group about which to display information. Minimum length = 1
systemgroup_systemuser_binding	<systemgroup_systemuser_binding[]>	Read-only	systemuser that can be bound to systemgroup.

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
systemgroup_nspartition_binding	<systemgroup_nspartition_binding[]>	Read-only	nspartition that can be bound to systemgroup.
systemgroup_systemcmdpolicy_binding	<systemgroup_systemcmdpolicy_binding[]>	Read-only	systemcmdpolicy that can be bound to systemgroup.

Operations

(click to see Properties)

GET| GET (ALL)

Some options that you can use for each operations:

- **Getting warnings in response:**NITRO allows you to get warnings in an operation by specifying the "warning" query parameter as "yes". For example, to get warnings while connecting to the Citrix ADC appliance, the URL is as follows:

  http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/login?warning=yes

  If any, the warnings are displayed in the response payload with the HTTP code "209 X-NITRO-WARNING".

- **Authenticated access for individual NITRO operations:**NITRO allows you to logon to the Citrix ADC appliance to perform individual operations. You can use this option instead of creating a NITRO session (using the login object) and then using that session to perform all operations,

  To do this, you must specify the username and password in the request header of the NITRO request as follows:

  X-NITRO-USER:<username>

  X-NITRO-PASS:<password>

  Note: In such cases, make sure that the request header DOES not include the following:

  Cookie:NITRO_AUTH_TOKEN=<tokenvalue>

`*Note:`*

Mandatory parameters are marked in redand placeholder content is marked in <green>.

get

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/systemgroup_binding/groupname_value<String>

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "systemgroup_binding": [ {

"groupname":<String_value>,

"systemgroup_systemcmdpolicy_binding":<systemgroup_systemcmdpolicy_binding
[]_value>,

"systemgroup_nspartition_binding":<systemgroup_nspartition_binding[]_value
>,

"systemgroup_systemuser_binding":<systemgroup_systemuser_binding[]_value>

}]}
```

get (all)

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/systemgroup_binding

Query-parameters:

bulkbindings

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/systemgroup_binding?**bulkbindings=yes**

NITRO allows you to fetch bindings in bulk.

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "systemgroup_binding": [ {

"groupname":<String_value>,

"systemgroup_systemcmdpolicy_binding":<systemgroup_systemcmdpolicy_binding
[]_value>,

"systemgroup_nspartition_binding":<systemgroup_nspartition_binding[]_value
>,

"systemgroup_systemuser_binding":<systemgroup_systemuser_binding[]_value>

}]}
```

systemgroup_nspartition_binding

Binding object showing the nspartition that can be bound to systemgroup.

Properties

(click to see Operations)

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
partitionname	<String>	Read-write	Name of the Partition to bind to the system group. Minimum length = 1
groupname	<String>	Read-write	Name of the system group. Minimum length = 1
__count	<Double>	Read-write	count parameter

Operations

(click to see Properties)

ADD:| DELETE:| GET| GET (ALL)| COUNT

Some options that you can use for each operations:

- **Getting warnings in response:**NITRO allows you to get warnings in an operation by specifying the "warning" query parameter as "yes". For example, to get warnings while connecting to the Citrix ADC appliance, the URL is as follows:

  http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/login?warning=yes

  If any, the warnings are displayed in the response payload with the HTTP code "209 X-NITRO-WARNING".

- **Authenticated access for individual NITRO operations:**NITRO allows you to logon to the Citrix ADC appliance to perform individual operations. You can use this option instead of creating a NITRO session (using the login object) and then using that session to perform all operations,

  To do this, you must specify the username and password in the request header of the NITRO request as follows:

  X-NITRO-USER:<username>

X-NITRO-PASS:<password>

Note: In such cases, make sure that the request header DOES not include the following:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue>

`*Note:`*

Mandatory parameters are marked in redand placeholder content is marked in <green>.

add

URL:http://<Citrix-ADC-IP-address(NSIP)/nitro/v1/config/systemgroup_nspartition_binding

HTTP Method:PUT

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Content-Type:application/json

Request Payload:

```
{

"systemgroup_nspartition_binding":{

<b>"groupname":<String_value>,

</b>"partitionname":<String_value>

}}
```

Response:

HTTP Status Code on Success: 201 Created HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error

delete

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/systemgroup_nspartition_binding/groupname_value<String>

Query-parameters:

args

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/systemgroup_nspartition_binding/groupname_value<String>?**args=partitionname:<String_value>**

HTTP Method:DELETE

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue>

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error

get

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/systemgroup_nspartition_binding/groupname_value<String>

Query-parameters:

filter

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/systemgroup_nspartition_binding/groupname_value<String>?**filter=property-name1:property-value1,property-name2:property-value2**

Use this query-parameter to get the filtered set of systemgroup_nspartition_binding resources configured on Citrix ADC. Filtering can be done on any of the properties of the resource.

pagination

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/systemgroup_nspartition_binding/groupname_value<String>?**pagesize=#no;pageno=#no**

Use this query-parameter to get the systemgroup_nspartition_binding resources in chunks.

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "systemgroup_nspartition_binding": [ {

"partitionname":<String_value>,

"groupname":<String_value>

}]}
```

get (all)

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/systemgroup_nspartition_binding

Query-parameters:

bulkbindings

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/systemgroup_nspartition_binding?**bulkbindings=yes**

NITRO allows you to fetch bindings in bulk.

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "systemgroup_nspartition_binding": [ {

"partitionname":<String_value>,

"groupname":<String_value>

}]}
```

count

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/systemgroup_nspartition_binding/groupname_value<String>?**count=yes**

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{"systemgroup_nspartition_binding": [ { "__count": "#no"} ] }
```

systemgroup_systemcmdpolicy_binding

Binding object showing the systemcmdpolicy that can be bound to systemgroup.

Properties

(click to see Operations)

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
priority	<Double>	Read-write	The priority of the command policy.
policyname	<String>	Read-write	The name of command policy.
groupname	<String>	Read-write	Name of the system group. Minimum length = 1
__count	<Double>	Read-write	count parameter

Operations

(click to see Properties)

ADD:| DELETE:| GET| GET (ALL)| COUNT

Some options that you can use for each operations:

- **Getting warnings in response:**NITRO allows you to get warnings in an operation by specifying the "warning" query parameter as "yes". For example, to get warnings while connecting to the Citrix ADC appliance, the URL is as follows:

  http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/login?warning=yes

  If any, the warnings are displayed in the response payload with the HTTP code "209 X-NITRO-WARNING".
- **Authenticated access for individual NITRO operations:**NITRO allows you to logon to the Citrix ADC appliance to perform individual operations. You can use this option instead of

creating a NITRO session (using the login object) and then using that session to perform all operations,

To do this, you must specify the username and password in the request header of the NITRO request as follows:

X-NITRO-USER:<username>

X-NITRO-PASS:<password>

Note: In such cases, make sure that the request header DOES not include the following:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue>

`*Note:`*

Mandatory parameters are marked in redand placeholder content is marked in <green>.

add

URL:http://<Citrix-ADC-IP-address(NSIP)/nitro/v1/config/systemgroup_systemcmdpolicy_binding

HTTP Method:PUT

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Content-Type:application/json

Request Payload:

```
{

"systemgroup_systemcmdpolicy_binding":{

<b>"groupname":<String_value>,

</b>"policyname":<String_value>,

"priority":<Double_value>

}}
```

Response:

HTTP Status Code on Success: 201 Created HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error

delete

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/systemgroup_systemcmdpolicy_binding/groupname_value<String>

Query-parameters:

args

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/systemgroup_systemcmdpolicy_binding/groupname_value<String>?**args=policyname:<String_value>**

HTTP Method:DELETE

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue>

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error

get

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/systemgroup_systemcmdpolicy_binding/groupname_value<String>

Query-parameters:

filter

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/systemgroup_systemcmdpolicy_binding/groupname_value<String>?**filter=property-name1:property-value1,property-name2:property-value2**

Use this query-parameter to get the filtered set of systemgroup_systemcmdpolicy_binding resources configured on Citrix ADC. Filtering can be done on any of the properties of the resource.

pagination

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/systemgroup_systemcmdpolicy_binding/groupname_value<String>?**pagesize=#no;pageno=#no**

Use this query-parameter to get the systemgroup_systemcmdpolicy_binding resources in chunks.

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "systemgroup_systemcmdpolicy_binding": [ {

"priority":<Double_value>,

"policyname":<String_value>,

"groupname":<String_value>

}]}
```

get (all)

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/systemgroup_systemcmdpolicy_binding

Query-parameters:

bulkbindings

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/systemgroup_systemcmdpolicy_binding?**bulkbindings=yes**

NITRO allows you to fetch bindings in bulk.

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "systemgroup_systemcmdpolicy_binding": [ {

"priority":<Double_value>,

"policyname":<String_value>,

"groupname":<String_value>

}]}
```

count

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/systemgroup_systemcmdpolicy_binding/groupname_value<String>?**count=yes**

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{"systemgroup_systemcmdpolicy_binding": [ { "__count": "#no"} ] }
```

systemgroup_systemuser_binding

Binding object showing the systemuser that can be bound to systemgroup.

Properties

(click to see Operations)

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
username	<String>	Read-write	The system user.
groupname	<String>	Read-write	Name of the system group. Minimum length = 1
__count	<Double>	Read-write	count parameter

Operations

(click to see Properties)

ADD:| DELETE:| GET| GET (ALL)| COUNT

Some options that you can use for each operations:

- **Getting warnings in response:**NITRO allows you to get warnings in an operation by specifying the "warning" query parameter as "yes". For example, to get warnings while connecting to the Citrix ADC appliance, the URL is as follows:

  http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/login?warning=yes

  If any, the warnings are displayed in the response payload with the HTTP code "209 X-NITRO-WARNING".

- **Authenticated access for individual NITRO operations:**NITRO allows you to logon to the Citrix ADC appliance to perform individual operations. You can use this option instead of creating a NITRO session (using the login object) and then using that session to perform all operations,

  To do this, you must specify the username and password in the request header of the NITRO request as follows:

  X-NITRO-USER:<username>

  X-NITRO-PASS:<password>

  Note: In such cases, make sure that the request header DOES not include the following:

  Cookie:NITRO_AUTH_TOKEN=<tokenvalue>

`*Note:`*

Mandatory parameters are marked in redand placeholder content is marked in <green>.

add

URL:http://<Citrix-ADC-IP-address(NSIP)/nitro/v1/config/systemgroup_systemuser_binding

HTTP Method:PUT

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Content-Type:application/json

Request Payload:

```
{
```

```
"systemgroup_systemuser_binding":{

<b>"groupname":<String_value>,

</b>"username":<String_value>

}}
```

Response:

HTTP Status Code on Success: 201 Created HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error

delete

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/systemgroup_systemuser_binding/groupname_value<String>

Query-parameters:

args

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/systemgroup_systemuser_binding/groupname_value<String>?**args=username:<String_value>**

HTTP Method:DELETE

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue>

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error

get

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/systemgroup_systemuser_binding/groupname_value<String>

Query-parameters:

filter

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/systemgroup_systemuser_binding/groupname_value<String>?**filter=property-name1:property-value1,property-name2:property-value2**

Use this query-parameter to get the filtered set of systemgroup_systemuser_binding resources configured on Citrix ADC. Filtering can be done on any of the properties of the resource.

pagination

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/systemgroup_systemuser_binding/groupname_value<String>?**pagesize=#no;pageno=#no**

Use this query-parameter to get the systemgroup_systemuser_binding resources in chunks.

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "systemgroup_systemuser_binding": [ {

"username":<String_value>,

"groupname":<String_value>

}]}
```

get (all)

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/systemgroup_systemuser_binding

Query-parameters:

bulkbindings

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/systemgroup_systemuser_binding?**bulkbindings=yes**

NITRO allows you to fetch bindings in bulk.

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "systemgroup_systemuser_binding": [ {

"username":<String_value>,

"groupname":<String_value>

}]}
```

count

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/systemgroup_systemuser_binding/groupname_value<String>?**count=yes**

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{"systemgroup_systemuser_binding": [ { "__count": "#no"} ] }
```

systemhwerror

Configuration for Hardware errors resource.

Properties

(click to see Operations)

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
diskcheck	<Boolean>	Read-write	Perform only disk error checking.
response	<String>	Read-only	Response captured during hardware examination.

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
hwerrorcount	<Integer>	Read-only	Number of errors found. Check response field for more information on errors if this value is non zero.

Operations

(click to see Properties)

CHECK

Some options that you can use for each operations:

- **Getting warnings in response:**NITRO allows you to get warnings in an operation by specifying the "warning" query parameter as "yes". For example, to get warnings while connecting to the Citrix ADC appliance, the URL is as follows:

  http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/login?warning=yes

  If any, the warnings are displayed in the response payload with the HTTP code "209 X-NITRO-WARNING".

- **Authenticated access for individual NITRO operations:**NITRO allows you to logon to the Citrix ADC appliance to perform individual operations. You can use this option instead of creating a NITRO session (using the login object) and then using that session to perform all operations,

  To do this, you must specify the username and password in the request header of the NITRO request as follows:

  X-NITRO-USER:<username>

  X-NITRO-PASS:<password>

  Note: In such cases, make sure that the request header DOES not include the following:

  Cookie:NITRO_AUTH_TOKEN=<tokenvalue>

`*Note:`*

Mandatory parameters are marked in redand placeholder content is marked in <green>.

check

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/systemhwerror?**action=check**

HTTP Method:POST

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Content-Type:application/json

Request Payload:

```
{"systemhwerror":{

<b>"diskcheck":<Boolean_value>

</b>}}
```

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Payload:**

```
{ "systemhwerror": [ {

"response":<String_value>,

"hwerrorcount":<Integer_value>



}]}
```

systemkek

Configuration for 0 resource.

Properties

(click to see Operations)

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
passphrase	<String>	Read-write	Passphrase required to generate the key encryption key. Minimum length = 8 Maximum length = 32
password	<String>	Read-write	Password required to import the key encryption key. Minimum length = 8 Maximum length = 32

Operations

(click to see Properties)

CREATE| EXPORT| IMPORT

Some options that you can use for each operations:

- **Getting warnings in response:**NITRO allows you to get warnings in an operation by specifying the "warning" query parameter as "yes". For example, to get warnings while connecting to the Citrix ADC appliance, the URL is as follows:

  http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/login?warning=yes

  If any, the warnings are displayed in the response payload with the HTTP code "209 X-NITRO-WARNING".
- **Authenticated access for individual NITRO operations:**NITRO allows you to logon to the Citrix ADC appliance to perform individual operations. You can use this option instead of creating a NITRO session (using the login object) and then using that session to perform all operations,

To do this, you must specify the username and password in the request header of the NITRO request as follows:

X-NITRO-USER:<username>

X-NITRO-PASS:<password>

Note: In such cases, make sure that the request header DOES not include the following:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue>

`*Note:`*

Mandatory parameters are marked in redand placeholder content is marked in <green>.

create

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/systemkek?**action=create**

HTTP Method:POST

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Content-Type:application/json

Request Payload:

```
{"systemkek":{

<b>"passphrase":<String_value>

</b>}}
```

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error

export

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/systemkek?**action=export**

HTTP Method:POST

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Content-Type:application/json

Request Payload:

```
{"systemkek":{

"password":<String_value>

}}
```

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error

Import

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/systemkek?**action=Import**

HTTP Method:POST

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Content-Type:application/json

Request Payload:

```
{"systemkek":{

"password":<String_value>

}}
```

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error

systemparameter

Configuration for system parameter resource.

Properties

(click to see Operations)

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
rbaonresponse	<String>	Read-write	Enable or disable Role-Based Authentication (RBA) on responses. Default value: ENABLED Possible values = ENABLED, DISABLED
promptstring	<String>	Read-write	String to display at the command-line prompt. Can consist of letters, numbers, hyphen (-), period (.), hash (#), space ( ), at (@), equal (=), colon (:), underscore (_), and the following variables: * %u - Will be replaced by the user name.

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
			* %h - Will be replaced by the hostname of the Citrix ADC. * %t - Will be replaced by the current time in 12-hour format. * %T - Will be replaced by the current time in 24-hour format. * %d - Will be replaced by the current date. * %s - Will be replaced by the state of the Citrix ADC. Note: The 63-character limit for the length of the string does not apply to the characters that replace the variables. Minimum length = 1
natpcbforceflushlimit	<Double>	Read-write	Flush the system if the number of Network Address Translation Protocol

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
			Control Blocks (NATPCBs) exceeds this value. Default value: 2147483647 Minimum value = 1000
natpcbrstontimeout	<String>	Read-write	Send a reset signal to client and server connections when their NATPCBs time out. Avoids the buildup of idle TCP connections on both the sides. Default value: DISABLED Possible values = ENABLED, DISABLED
timeout	<Double>	Read-write	CLI session inactivity timeout, in seconds. If Restrictedtimeout argument is enabled, Timeout can have values in the range [300-86400] seconds. If Restrictedtimeout argument is disabled, Timeout can have values in the range [0, 10-100000000] seconds. Default

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
			value is 900 seconds.
localauth	<String>	Read-write	When enabled, local users can access Citrix ADC even when external authentication is configured. When disabled, local users are not allowed to access the Citrix ADC, Local users can access the Citrix ADC only when the configured external authentication servers are unavailable. This parameter is not applicable to SSH Key-based authentication. Default value: ENABLED Possible values = ENABLED, DISABLED
minpasswordlen	<Double>	Read-write	Minimum length of system user password. When strong password is enabled default minimum length is 4. User entered value can be greater than or equal to 4. Default mininum value is 1 when strong

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
			password is disabled. Maximum value is 127 in both cases. Minimum value = 1 Maximum value = 127
strongpassword	<String>	Read-write	After enabling strong password (enableall / enablelocal - not included in exclude list), all the passwords / sensitive information must have - Atleast 1 Lower case character, Atleast 1 Upper case character, Atleast 1 numeric character, Atleast 1 special character (~, `, !, @, #, $, %, \^, ;, *, -, _, =, +, {, }, [,], |, \, :, <, >, /, ., ,, " "). Exclude list in case of enablelocal is - NS_FIPS, NS_CRL, NS_RSAKEY, NS_PKCS12, NS_PKCS8, NS_LDAP, NS_TACACS, NS_TACACSACTION, NS_RADIUS, NS_RADIUSACTION, NS_ENCRYPTION_PARAMS. So no

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
			Strong Password checks will be performed on these ObjectType commands for enablelocal case. Default value: disabled Possible values = enableall, enablelocal, disabled
restrictedtimeout	<String>	Read-write	Enable/Disable the restricted timeout behaviour. When enabled, timeout cannot be configured beyond admin configured timeout and also it will have the [minimum - maximum] range check. When disabled, timeout will have the old behaviour. By default the value is disabled. Default value: DISABLED Possible values = ENABLED, DISABLED
fipsusermode	<String>	Read-write	Use this option to set the FIPS mode for key user-land processes. When

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
			enabled, these user-land processes will operate in FIPS mode. In this mode, theses processes will use FIPS 140-2 Level-1 certified crypto algorithms. Default is disabled, wherein, these user-land processes will not operate in FIPS mode. Default value: DISABLED Possible values = ENABLED, DISABLED
doppler	<String>	Read-write	Enable or disable Doppler. Default value: 0 Possible values = ENABLED, DISABLED
googleanalytics	<String>	Read-write	Enable or disable Google analytics. Default value: DISABLED Possible values = ENABLED, DISABLED
totalauthtimeout	<Double>	Read-write	Total time a request can take for

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
			authentication/ authorization. Default value: 20 Minimum value = 5 Maximum value = 120
cliloglevel	<String>	Read-write	Audit log level, which specifies the types of events to log for cli executed commands. Available values function as follows: * EMERGENCY - Events that indicate an immediate crisis on the server. * ALERT - Events that might require action. * CRITICAL - Events that indicate an imminent server crisis. * ERROR - Events that indicate some type of error. * WARNING - Events that require action in the near future.

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
			* NOTICE - Events that the administrator should know about. * INFORMATIONAL - All but low-level events. * DEBUG - All events, in extreme detail. Default value: INFORMATIONAL Possible values = EMERGENCY, ALERT, CRITICAL, ERROR, WARNING, NOTICE, INFORMATIONAL, DEBUG
forcepasswordchange	<String>	Read-write	Enable or disable force password change for nsroot user. Default value: DISABLED Possible values = ENABLED, DISABLED
basicauth	<String>	Read-write	Enable or disable basic authentication for Nitro API.

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
			Default value: ENABLED Possible values = ENABLED, DISABLED
reauthonauthparamchange	<String>	Read-write	Enable or disable External user reauthentication when authentication parameter changes. Default value: DISABLED Possible values = ENABLED, DISABLED
removesensitivefiles	<String>	Read-write	Use this option to remove the sensitive files from the system like authorise keys, public keys etc. The commands which will remove sensitive files when this system paramter is enabled are rm cluster instance, rm cluster node, rm ha node, clear config full, join cluster and add cluster instance. Default value: DISABLED Possible values = ENABLED, DISABLED

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
maxclient	<Double>	Read-only	Maximum number of client connection allowed by the system. Minimum value = 20 Maximum value = 40
allowdefaultpartition	<String>	Read-only	Enable/Disable the allowing partition users to access default partition. Default value: NO Possible values = YES, NO

Operations

(click to see Properties)

UPDATE| UNSET| GET (ALL)

Some options that you can use for each operations:

- **Getting warnings in response:**NITRO allows you to get warnings in an operation by specifying the "warning" query parameter as "yes". For example, to get warnings while connecting to the Citrix ADC appliance, the URL is as follows:

  http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/login?warning=yes

  If any, the warnings are displayed in the response payload with the HTTP code "209 X-NITRO-WARNING".
- **Authenticated access for individual NITRO operations:**NITRO allows you to logon to the Citrix ADC appliance to perform individual operations. You can use this option instead of creating a NITRO session (using the login object) and then using that session to perform all operations,

To do this, you must specify the username and password in the request header of the NITRO request as follows:

X-NITRO-USER:<username>

X-NITRO-PASS:<password>

Note: In such cases, make sure that the request header DOES not include the following:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue>

`*Note:`*

Mandatory parameters are marked in redand placeholder content is marked in <green>.

update

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/systemparameter

HTTP Method:PUT

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Content-Type:application/json

Request Payload:

```
{"systemparameter":{

"rbaonresponse":<String_value>,

"promptstring":<String_value>,

"natpcbforceflushlimit":<Double_value>,

"natpcbrstontimeout":<String_value>,

"timeout":<Double_value>,

"localauth":<String_value>,

"minpasswordlen":<Double_value>,
```

```
"strongpassword":<String_value>,

"restrictedtimeout":<String_value>,

"fipsusermode":<String_value>,

"doppler":<String_value>,

"googleanalytics":<String_value>,

"totalauthtimeout":<Double_value>,

"cliloglevel":<String_value>,

"forcepasswordchange":<String_value>,

"basicauth":<String_value>,

"reauthonauthparamchange":<String_value>,

"removesensitivefiles":<String_value>

}}
```

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error

unset

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/systemparameter?**action=unset**

HTTP Method:POST

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Content-Type:application/json

Request Payload:

```
{"systemparameter":{
"minpasswordlen":true,
"rbaonresponse":true,
"promptstring":true,
"natpcbforceflushlimit":true,
"natpcbrstontimeout":true,
"timeout":true,
"localauth":true,
"strongpassword":true,
"restrictedtimeout":true,
"fipsusermode":true,
"doppler":true,
"googleanalytics":true,
"totalauthtimeout":true,
"cliloglevel":true,
"forcepasswordchange":true,
"basicauth":true,
"reauthonauthparamchange":true,
"removesensitivefiles":true
}}
```

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error

get (all)

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/systemparameter

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "systemparameter": [ {

"rbaonresponse":<String_value>,

"promptstring":<String_value>,

"natpcbforceflushlimit":<Double_value>,

"natpcbrstontimeout":<String_value>,

"timeout":<Double_value>,

"maxclient":<Double_value>,

"localauth":<String_value>,

"minpasswordlen":<Double_value>,
```

```
"strongpassword":<String_value>,

"restrictedtimeout":<String_value>,

"fipsusermode":<String_value>,

"doppler":<String_value>,

"googleanalytics":<String_value>,

"cliloglevel":<String_value>,

"forcepasswordchange":<String_value>,

"totalauthtimeout":<Double_value>,

"basicauth":<String_value>,

"allowdefaultpartition":<String_value>,

"reauthonauthparamchange":<String_value>,

"removesensitivefiles":<String_value>

}]}
```

systemsession

Configuration for system session resource.

Properties

(click to see Operations)

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
sid	<Double>	Read-write	ID of the system session about which to display information. Minimum value = 1

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
all	<Boolean>	Read-write	Terminate all the system sessions except the current session.
username	<String>	Read-only	user name of the session.
logintime	<String>	Read-only	logged-in time of this session.
logintimelocal	<String>	Read-only	logged-in time (in local ADC timezone) of this session.
lastactivitytime	<String>	Read-only	last activity time of this session.
lastactivitytimelocal	<String>	Read-only	last activity time (in local ADC timezone) of this session.
expirytime	<Integer>	Read-only	Time left in expire the session in seconds.
numofconnections	<Double>	Read-only	number of connection using this token.
currentconn	<Boolean>	Read-only	True if the token is used for current session.
clienttype	<String>	Read-only	client type of the session. Possible values = CLI, CONSOLE, API, GUI

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
partitionname	<String>	Read-only	Name of the partition where the user is currently present.
__count	<Double>	Read-only	count parameter

Operations

(click to see Properties)

KILL| GET (ALL)| GET| COUNT

Some options that you can use for each operations:

- **Getting warnings in response:**NITRO allows you to get warnings in an operation by specifying the "warning" query parameter as "yes". For example, to get warnings while connecting to the Citrix ADC appliance, the URL is as follows:

  http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/login?warning=yes

  If any, the warnings are displayed in the response payload with the HTTP code "209 X-NITRO-WARNING".

- **Authenticated access for individual NITRO operations:**NITRO allows you to logon to the Citrix ADC appliance to perform individual operations. You can use this option instead of creating a NITRO session (using the login object) and then using that session to perform all operations,

  To do this, you must specify the username and password in the request header of the NITRO request as follows:

  X-NITRO-USER:<username>

  X-NITRO-PASS:<password>

  Note: In such cases, make sure that the request header DOES not include the following:

  Cookie:NITRO_AUTH_TOKEN=<tokenvalue>

`*Note:` *

Mandatory parameters are marked in redand placeholder content is marked in <green>.

kill

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/systemsession?**action=kill**

HTTP Method:POST

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Content-Type:application/json

Request Payload:

```
{"systemsession":{

"sid":<Double_value>,

"all":<Boolean_value>

}}
```

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error

get (all)

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/systemsession

Query-parameters:

attrs

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/systemsession?**attrs=property-name1,property-name2**

Use this query parameter to specify the resource details that you want to retrieve.

filter

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/systemsession?**filter=property-name1:property-val1,property-name2:property-val2**

Use this query-parameter to get the filtered set of systemsession resources configured on Citrix ADC. Filtering can be done on any of the properties of the resource.

view

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/systemsession?**view=summary**

Note: By default, the retrieved results are displayed in detail view (?view=detail).

pagination

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/systemsession?**pagesize=#no;pageno=#no**

Use this query-parameter to get the systemsession resources in chunks.

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "systemsession": [ {

"sid":<Double_value>,

"username":<String_value>,

"logintime":<String_value>,

"logintimelocal":<String_value>,
```

```
"lastactivitytime":<String_value>,

"lastactivitytimelocal":<String_value>,

"expirytime":<Integer_value>,

"numofconnections":<Double_value>,

"currentconn":<Boolean_value>,

"clienttype":<String_value>,

"partitionname":<String_value>

}]}
```

get

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/systemsession/sid_value<Double>

Query-parameters:

attrs

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/systemsession/sid_value<Double>?**attrs=property-name1,property-name2**

Use this query parameter to specify the resource details that you want to retrieve.

view

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/systemsession/sid_value<Double>?**view=summary**

Note: By default, the retrieved results are displayed in detail view (?view=detail).

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "systemsession": [ {

"sid":<Double_value>,

"username":<String_value>,

"logintime":<String_value>,

"logintimelocal":<String_value>,

"lastactivitytime":<String_value>,

"lastactivitytimelocal":<String_value>,

"expirytime":<Integer_value>,

"numofconnections":<Double_value>,

"currentconn":<Boolean_value>,

"clienttype":<String_value>,

"partitionname":<String_value>

}]}
```

count

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/systemsession?**count=yes**

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "systemsession": [ { "__count": "#no"} ] }
```

systemsshkey

Configuration for 0 resource.

Properties

(click to see Operations)

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
name	<String>	Read-write	URL \(protocol, host, path, and file name\) from where the location file will be imported. NOTE: The import fails if the object to be imported is on an HTTPS server that requires client certificate authentication for access.

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
			Minimum length = 1 Maximum length = 255
sshkeytype	<String>	Read-write	The type of the ssh key whether public or private key. Possible values = PRIVATE, PUBLIC
src	<String>	Read-write	URL \(protocol, host, path, and file name\) from where the location file will be imported. NOTE: The import fails if the object to be imported is on an HTTPS server that requires client certificate authentication for access. Minimum length = 1 Maximum length = 2047

Operations

(click to see Properties)

DELETE| IMPORT| GET (ALL)

Some options that you can use for each operations:

- **Getting warnings in response:**NITRO allows you to get warnings in an operation by specifying the "warning" query parameter as "yes". For example, to get warnings while connecting to the Citrix ADC appliance, the URL is as follows:

  http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/login?warning=yes

  If any, the warnings are displayed in the response payload with the HTTP code "209 X-NITRO-WARNING".

- **Authenticated access for individual NITRO operations:**NITRO allows you to logon to the Citrix ADC appliance to perform individual operations. You can use this option instead of creating a NITRO session (using the login object) and then using that session to perform all operations,

  To do this, you must specify the username and password in the request header of the NITRO request as follows:

  X-NITRO-USER:<username>

  X-NITRO-PASS:<password>

  Note: In such cases, make sure that the request header DOES not include the following:

  Cookie:NITRO_AUTH_TOKEN=<tokenvalue>

`*Note:`*

Mandatory parameters are marked in redand placeholder content is marked in <green>.

delete

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/systemsshkey/name_value<String>

HTTP Method:DELETE

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue>

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error

Import

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/systemsshkey?**action=Import**

HTTP Method:POST

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Content-Type:application/json

Request Payload:

```
{"systemsshkey":{

<b>"name":<String_value>,

</b><b>"src":<String_value>,

</b><b>"sshkeytype":<String_value>

</b>}}
```

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error

get (all)

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/systemsshkey

Query-parameters:

args

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/systemsshkey?**args=sshkeytype:<String_value>**

Use this query-parameter to get systemsshkey resources based on additional properties.

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "systemsshkey": [ {

sshkeytype:<String_value>"name":<String_value>

}]}
```

systemuser

Configuration for system user resource.

Properties

(click to see Operations)

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
username	<String>	Read-write	Name for a user. Must begin with a letter, number, or the underscore (_) character, and must contain only alphanumeric, hyphen (-), period (.), hash (#), space ( ), at (@), equal (=),

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
			colon (:), and underscore characters. Cannot be changed after the user is added. CLI Users: If the name includes one or more spaces, enclose the name in double or single quotation marks (for example, "my user" or 'my user'). Minimum length = 1
password	<String>	Read-write	Password for the system user. Can include any ASCII character. Minimum length = 1
externalauth	<String>	Read-write	Whether to use external authentication servers for the system user authentication or not. Default value: ENABLED Possible values = ENABLED, DISABLED
promptstring	<String>	Read-write	String to display at the command-line prompt. Can consist of letters, numbers, hyphen (-), period (.),

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
			hash (#), space ( ), at (@), equal (=), colon (:), underscore (_), and the following variables: * %u - Will be replaced by the user name. * %h - Will be replaced by the hostname of the Citrix ADC. * %t - Will be replaced by the current time in 12-hour format. * %T - Will be replaced by the current time in 24-hour format. * %d - Will be replaced by the current date. * %s - Will be replaced by the state of the Citrix ADC. Note: The 63-character limit for the length of the string does not apply to the characters that

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
			replace the variables. Minimum length = 1
timeout	<Double>	Read-write	CLI session inactivity timeout, in seconds. If Restrictedtimeout argument of system parameter is enabled, Timeout can have values in the range [300-86400] seconds. If Restrictedtimeout argument of system parameter is disabled, Timeout can have values in the range [0, 10-100000000] seconds. Default value is 900 seconds.
logging	<String>	Read-write	Users logging privilege. Default value: DISABLED Possible values = ENABLED, DISABLED
maxsession	<Double>	Read-write	Maximum number of client connection allowed per user. Default value: 20

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
			Minimum value = 1 Maximum value = 40
encrypted	<Boolean>	Read-only	.
hashmethod	<String>	Read-only	. Possible values = SHA1, SHA512
promptinheritedfrom	<String>	Read-only	From where the prompt has been inherited. Possible values = User, Group, Global, Climode
timeoutkind	<String>	Read-only	From where the timeout has been inherited. Possible values = User, Group, Global, Climode
__count	<Double>	Read-only	count parameter

Operations

(click to see Properties)

ADD| DELETE| UPDATE| UNSET| GET (ALL)| GET| COUNT

Some options that you can use for each operations:

- **Getting warnings in response:**NITRO allows you to get warnings in an operation by specifying the "warning" query parameter as "yes". For example, to get warnings while connecting to the Citrix ADC appliance, the URL is as follows:

  http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/login?warning=yes

If any, the warnings are displayed in the response payload with the HTTP code "209 X-NITRO-WARNING".

- **Authenticated access for individual NITRO operations:**NITRO allows you to logon to the Citrix ADC appliance to perform individual operations. You can use this option instead of creating a NITRO session (using the login object) and then using that session to perform all operations,

  To do this, you must specify the username and password in the request header of the NITRO request as follows:

  X-NITRO-USER:<username>

  X-NITRO-PASS:<password>

  Note: In such cases, make sure that the request header DOES not include the following:

  Cookie:NITRO_AUTH_TOKEN=<tokenvalue>

`*Note:`*

Mandatory parameters are marked in redand placeholder content is marked in <green>.

add

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/systemuser

HTTP Method:POST

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Content-Type:application/json

Request Payload:

```
{"systemuser":{

<b>"username":<String_value>,

</b><b>"password":<String_value>,

</b>"externalauth":<String_value>,
```

```
"promptstring":<String_value>,

"timeout":<Double_value>,

"logging":<String_value>,

"maxsession":<Double_value>

}}
```

Response:

HTTP Status Code on Success: 201 Created HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error

delete

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/systemuser/username_value<String>

HTTP Method:DELETE

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue>

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error

update

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/systemuser

HTTP Method:PUT

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Content-Type:application/json

Request Payload:

```
{"systemuser":{

<b>"username":<String_value>,

</b>"password":<String_value>,

"externalauth":<String_value>,

"promptstring":<String_value>,

"timeout":<Double_value>,

"logging":<String_value>,

"maxsession":<Double_value>

}}
```

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error

unset

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/systemuser?**action=unset**

HTTP Method:POST

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Content-Type:application/json

Request Payload:

```
{"systemuser":{
```

```
<b>"username":<String_value>,

</b>"externalauth":true,

"promptstring":true,

"timeout":true,

"logging":true,

"maxsession":true

}}
```

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error

get (all)

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/systemuser

Query-parameters:

attrs

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/systemuser?**attrs=property-name1,property-name2**

Use this query parameter to specify the resource details that you want to retrieve.

filter

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/systemuser?**filter=property-name1:property-val1,property-name2:property-val2**

Use this query-parameter to get the filtered set of systemuser resources configured on Citrix ADC. Filtering can be done on any of the properties of the resource.

view

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/systemuser?**view=summary**

Note: By default, the retrieved results are displayed in detail view (?view=detail).

pagination

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/systemuser?**pagesize=#no;pageno=#no**

Use this query-parameter to get the systemuser resources in chunks.

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "systemuser": [ {

"username":<String_value>,

"password":<String_value>,

"encrypted":<Boolean_value>,

"hashmethod":<String_value>,

"externalauth":<String_value>,

"promptstring":<String_value>,

"promptinheritedfrom":<String_value>,

"timeout":<Double_value>,
```

```
"timeoutkind":<String_value>,

"logging":<String_value>,

"maxsession":<Double_value>

}]}
```

get

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/systemuser/username_value<String>

Query-parameters:

attrs

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/systemuser/username_value<String>?**attrs=property-name1,property-name2**

Use this query parameter to specify the resource details that you want to retrieve.

view

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/systemuser/username_value<String>?**view=summary**

Note: By default, the retrieved results are displayed in detail view (?view=detail).

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "systemuser": [ {

"username":<String_value>,

"password":<String_value>,

"encrypted":<Boolean_value>,

"hashmethod":<String_value>,

"externalauth":<String_value>,

"promptstring":<String_value>,

"promptinheritedfrom":<String_value>,

"timeout":<Double_value>,

"timeoutkind":<String_value>,

"logging":<String_value>,

"maxsession":<Double_value>

}]}
```

count

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/systemuser?**count=yes**

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "systemuser": [ { "__count": "#no"} ] }
```

systemuser_binding

Binding object which returns the resources bound to systemuser.

Properties

(click to see Operations)

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
username	<String>	Read-write	Name of a system user about whom to display information. Minimum length = 1
systemuser_systemgroup_binding	<systemuser_systemgroup_binding[]>	Read-only	systemgroup that can be bound to systemuser.
systemuser_nspartition_binding	<systemuser_nspartition_binding[]>	Read-only	nspartition that can be bound to systemuser.
systemuser_systemcmdpolicy_binding	<systemuser_systemcmdpolicy_binding[]>	Read-only	systemcmdpolicy that can be bound to systemuser.

Operations

(click to see Properties)

GET| GET (ALL)

Some options that you can use for each operations:

- **Getting warnings in response:**NITRO allows you to get warnings in an operation by specifying the "warning" query parameter as "yes". For example, to get warnings while connecting to the Citrix ADC appliance, the URL is as follows:

  http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/login?warning=yes

  If any, the warnings are displayed in the response payload with the HTTP code "209 X-NITRO-WARNING".

- **Authenticated access for individual NITRO operations:**NITRO allows you to logon to the Citrix ADC appliance to perform individual operations. You can use this option instead of creating a NITRO session (using the login object) and then using that session to perform all operations,

  To do this, you must specify the username and password in the request header of the NITRO request as follows:

  X-NITRO-USER:<username>

  X-NITRO-PASS:<password>

  Note: In such cases, make sure that the request header DOES not include the following:

  Cookie:NITRO_AUTH_TOKEN=<tokenvalue>

`*Note:`*

Mandatory parameters are marked in redand placeholder content is marked in <green>.

get

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/systemuser_binding/username_value<String>

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "systemuser_binding": [ {

"username":<String_value>,

"systemuser_systemgroup_binding":<systemuser_systemgroup_binding[]_value>,

"systemuser_nspartition_binding":<systemuser_nspartition_binding[]_value>,

"systemuser_systemcmdpolicy_binding":<systemuser_systemcmdpolicy_binding[]
_value>

}]}
```

get (all)

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/systemuser_binding

Query-parameters:

bulkbindings

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/systemuser_binding?**bulkbindings=yes**

NITRO allows you to fetch bindings in bulk.

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "systemuser_binding": [ {

"username":<String_value>,

"systemuser_systemgroup_binding":<systemuser_systemgroup_binding[]_value>,

"systemuser_nspartition_binding":<systemuser_nspartition_binding[]_value>,

"systemuser_systemcmdpolicy_binding":<systemuser_systemcmdpolicy_binding[]
_value>

}]}
```

systemuser_nspartition_binding

Binding object showing the nspartition that can be bound to systemuser.

Properties

(click to see Operations)

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
username	<String>	Read-write	Name of the system-user entry to which to bind the command policy. Minimum length = 1
partitionname	<String>	Read-write	Name of the Partition to bind to the system user.
__count	<Double>	Read-write	count parameter

Operations

(click to see Properties)

ADD:| DELETE:| GET| GET (ALL)| COUNT

Some options that you can use for each operations:

- **Getting warnings in response:**NITRO allows you to get warnings in an operation by specifying the "warning" query parameter as "yes". For example, to get warnings while connecting to the Citrix ADC appliance, the URL is as follows:

  http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/login?warning=yes

  If any, the warnings are displayed in the response payload with the HTTP code "209 X-NITRO-WARNING".

- **Authenticated access for individual NITRO operations:**NITRO allows you to logon to the Citrix ADC appliance to perform individual operations. You can use this option instead of creating a NITRO session (using the login object) and then using that session to perform all operations,

  To do this, you must specify the username and password in the request header of the NITRO request as follows:

  X-NITRO-USER:<username>

  X-NITRO-PASS:<password>

  Note: In such cases, make sure that the request header DOES not include the following:

  Cookie:NITRO_AUTH_TOKEN=<tokenvalue>

`*Note:`*

Mandatory parameters are marked in redand placeholder content is marked in <green>.

add

URL:http://<Citrix-ADC-IP-address(NSIP)/nitro/v1/config/systemuser_nspartition_binding

HTTP Method:PUT

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Content-Type:application/json

Request Payload:

```
{

"systemuser_nspartition_binding":{

<b>"username":<String_value>,

</b>"partitionname":<String_value>

}}
```

Response:

HTTP Status Code on Success: 201 Created HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error

delete

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/systemuser_nspartition_binding/username_value<String>

Query-parameters:

args

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/systemuser_nspartition_binding/username_value<String>?**args=partitionname:<String_value>**

HTTP Method:DELETE

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue>

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error

get

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/systemuser_nspartition_binding/username_value<String>

Query-parameters:

filter

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/systemuser_nspartition_binding/username_value<String>?**filter=property-name1:property-value1,property-name2:property-value2**

Use this query-parameter to get the filtered set of systemuser_nspartition_binding resources configured on Citrix ADC. Filtering can be done on any of the properties of the resource.

pagination

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/systemuser_nspartition_binding/username_value<String>?**pagesize=#no;pageno=#no**

Use this query-parameter to get the systemuser_nspartition_binding resources in chunks.

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "systemuser_nspartition_binding": [ {

"username":<String_value>,
```

```
"partitionname":<String_value>

}]}
```

get (all)

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/systemuser_nspartition_binding

Query-parameters:

bulkbindings

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/systemuser_nspartition_binding?**bulkbindings=yes**

NITRO allows you to fetch bindings in bulk.

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "systemuser_nspartition_binding": [ {

"username":<String_value>,

"partitionname":<String_value>

}]}
```

count

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/systemuser_nspartition_binding/username_value<String>?**count=yes**

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{"systemuser_nspartition_binding": [ { "__count": "#no"} ] }
```

systemuser_systemcmdpolicy_binding

Binding object showing the systemcmdpolicy that can be bound to systemuser.

Properties

(click to see Operations)

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
priority	<Double>	Read-write	The priority of the policy.
policyname	<String>	Read-write	The name of command policy.

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
username	<String>	Read-write	Name of the system-user entry to which to bind the command policy. Minimum length = 1
__count	<Double>	Read-write	count parameter

Operations

(click to see Properties)

ADD:| DELETE:| GET| GET (ALL)| COUNT

Some options that you can use for each operations:

- **Getting warnings in response:**NITRO allows you to get warnings in an operation by specifying the "warning" query parameter as "yes". For example, to get warnings while connecting to the Citrix ADC appliance, the URL is as follows:

  http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/login?warning=yes

  If any, the warnings are displayed in the response payload with the HTTP code "209 X-NITRO-WARNING".

- **Authenticated access for individual NITRO operations:**NITRO allows you to logon to the Citrix ADC appliance to perform individual operations. You can use this option instead of creating a NITRO session (using the login object) and then using that session to perform all operations,

  To do this, you must specify the username and password in the request header of the NITRO request as follows:

  X-NITRO-USER:<username>

  X-NITRO-PASS:<password>

  Note: In such cases, make sure that the request header DOES not include the following:

  Cookie:NITRO_AUTH_TOKEN=<tokenvalue>

`*Note:` *

Mandatory parameters are marked in redand placeholder content is marked in <green>.

add

URL:http://<Citrix-ADC-IP-address(NSIP)/nitro/v1/config/systemuser_systemcmdpolicy_binding

HTTP Method:PUT

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Content-Type:application/json

Request Payload:

```
{

"systemuser_systemcmdpolicy_binding":{

<b>"username":<String_value>,

</b>"policyname":<String_value>,

"priority":<Double_value>

}}
```

Response:

HTTP Status Code on Success: 201 Created HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error

delete

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/systemuser_systemcmdpolicy_binding/username_value<String>

Query-parameters:

args

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/systemuser_systemcmdpolicy_binding/username_value<String>?**args=policyname:<String_value>**

HTTP Method:DELETE

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue>

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error

get

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/systemuser_systemcmdpolicy_binding/username_value<String>

Query-parameters:

filter

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/systemuser_systemcmdpolicy_binding/username_value<String>?**filter=property-name1:property-value1,property-name2:property-value2**

Use this query-parameter to get the filtered set of systemuser_systemcmdpolicy_binding resources configured on Citrix ADC. Filtering can be done on any of the properties of the resource.

pagination

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/systemuser_systemcmdpolicy_binding/username_value<String>?**pagesize=#no;pageno=#no**

Use this query-parameter to get the systemuser_systemcmdpolicy_binding resources in chunks.

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "systemuser_systemcmdpolicy_binding": [ {

"priority":<Double_value>,

"policyname":<String_value>,

"username":<String_value>

}]}
```

get (all)

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/systemuser_systemcmdpolicy_binding

Query-parameters:

bulkbindings

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/systemuser_systemcmdpolicy_binding?**bulkbindings=yes**

NITRO allows you to fetch bindings in bulk.

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "systemuser_systemcmdpolicy_binding": [ {

"priority":<Double_value>,

"policyname":<String_value>,

"username":<String_value>

}]}
```

count

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/systemuser_systemcmdpolicy_binding/username_value<String>?**count=yes**

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{"systemuser_systemcmdpolicy_binding": [ { "__count": "#no"} ] }
```

systemuser_systemgroup_binding

Binding object showing the systemgroup that can be bound to systemuser.

Properties

(click to see Operations)

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
username	<String>	Read-write	Name of the system-user entry to which to bind the command policy. Minimum length = 1
groupname	<String>	Read-write	The system group. .
__count	<Double>	Read-write	count parameter

Operations

(click to see Properties)

GET| GET (ALL)| COUNT

Some options that you can use for each operations:

- **Getting warnings in response:**NITRO allows you to get warnings in an operation by specifying the "warning" query parameter as "yes". For example, to get warnings while connecting to the Citrix ADC appliance, the URL is as follows:

  http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/login?warning=yes

  If any, the warnings are displayed in the response payload with the HTTP code "209 X-NITRO-WARNING".

- **Authenticated access for individual NITRO operations:**NITRO allows you to logon to the Citrix ADC appliance to perform individual operations. You can use this option instead of creating a NITRO session (using the login object) and then using that session to perform all operations,

  To do this, you must specify the username and password in the request header of the NITRO request as follows:

  X-NITRO-USER:<username>

  X-NITRO-PASS:<password>

  Note: In such cases, make sure that the request header DOES not include the following:

  Cookie:NITRO_AUTH_TOKEN=<tokenvalue>

`*Note:`*

Mandatory parameters are marked in redand placeholder content is marked in <green>.

get

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/systemuser_systemgroup_binding/username_value<String>

Query-parameters:

filter

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/systemuser_systemgroup_binding/username_value<String>?**filter=property-name1:property-value1,property-name2:property-value2**

Use this query-parameter to get the filtered set of systemuser_systemgroup_binding resources configured on Citrix ADC. Filtering can be done on any of the properties of the resource.

pagination

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/systemuser_systemgroup_binding/username_value<String>?**pagesize=#no;pageno=#no**

Use this query-parameter to get the systemuser_systemgroup_binding resources in chunks.

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "systemuser_systemgroup_binding": [ {

"username":<String_value>,

"groupname":<String_value>

}]}
```

get (all)

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/systemuser_systemgroup_binding

Query-parameters:

bulkbindings

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/systemuser_systemgroup_binding?**bulkbindings=yes**

NITRO allows you to fetch bindings in bulk.

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "systemuser_systemgroup_binding": [ {

"username":<String_value>,

"groupname":<String_value>

}]}
```

count

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/systemuser_systemgroup_binding/username_value<String>?**count=yes**

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{"systemuser_systemgroup_binding": [ { "__count": "#no"} ] }
```

Traffic Management

Traffic Management

TM session/policy configuration.

NAME	DESCRIPTION
tmformssoaction	Configuration for Form sso action
tmglobal_auditnslogpolicy_binding	Binding object showing the auditnslogpolicy that can be bound to tmglobal
tmglobal_auditsyslogpolicy_binding	Binding object showing the auditsyslogpolicy that can be bound to tmglobal
tmglobal_binding	Binding object showing the resources that can be bound to tmglobal
tmglobal_tmsessionpolicy_binding	Binding object showing the tmsessionpolicy that can be bound to tmglobal
tmglobal_tmtrafficpolicy_binding	Binding object showing the tmtrafficpolicy that can be bound to tmglobal
tmsamlssoprofile	Configuration for SAML sso action
tmsessionaction	Configuration for TM session action
tmsessionparameter	Configuration for session parameter
tmsessionpolicy	Configuration for TM session policy
tmsessionpolicy_aaagroup_binding	Binding object showing the aaagroup that can be bound to tmsessionpolicy
tmsessionpolicy_aaauser_binding	Binding object showing the aaauser that can be bound to tmsessionpolicy

NAME	DESCRIPTION
tmsessionpolicy_authenticationvserver_binding	Binding object showing the authenticationvserver that can be bound to tmsessionpolicy
tmsessionpolicy_binding	Binding object showing the resources that can be bound to tmsessionpolicy
tmsessionpolicy_tmglobal_binding	Binding object showing the tmglobal that can be bound to tmsessionpolicy
tmtrafficaction	Configuration for TM traffic action
tmtrafficpolicy	Configuration for TM traffic policy
tmtrafficpolicy_binding	Binding object showing the resources that can be bound to tmtrafficpolicy
tmtrafficpolicy_csvserver_binding	Binding object showing the csvserver that can be bound to tmtrafficpolicy
tmtrafficpolicy_lbvserver_binding	Binding object showing the lbvserver that can be bound to tmtrafficpolicy
tmtrafficpolicy_tmglobal_binding	Binding object showing the tmglobal that can be bound to tmtrafficpolicy

tmformssoaction

Configuration for Form sso action resource.

Properties

(click to see Operations)

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
name	<String>	Read-write	Name for the new form-based single sign-on profile. Must begin with an ASCII alphanumeric or underscore (_)

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
			character, and must contain only ASCII alphanumeric, underscore, hash (#), period (.), space, colon (:), at (@), equals (=), and hyphen (-) characters. Cannot be changed after an SSO action is created. The following requirement applies only to the Citrix ADC CLI: If the name includes one or more spaces, enclose the name in double or single quotation marks (for example, "my action" or 'my action'). Minimum length = 1
actionurl	<String>	Read-write	URL to which the completed form is submitted. Minimum length = 1
userfield	<String>	Read-write	Name of the form field in which the user types in the user ID. Minimum length = 1

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
passwdfield	<String>	Read-write	Name of the form field in which the user types in the password. Minimum length = 1
ssosuccessrule	<String>	Read-write	Expression, that checks to see if single sign-on is successful.
namevaluepair	<String>	Read-write	Name-value pair attributes to send to the server in addition to sending the username and password. Value names are separated by an ampersand (;) (for example, name1=value1;name2=value2).
responsesize	<Double>	Read-write	Number of bytes, in the response, to parse for extracting the forms. Default value: 8096
nvtype	<String>	Read-write	Type of processing of the name-value pair. If you specify STATIC, the values configured by the administrator are used. For DYNAMIC, the response is parsed, and the form is extracted and then submitted.

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
			Default value: DYNAMIC Possible values = STATIC, DYNAMIC
submitmethod	<String>	Read-write	HTTP method used by the single sign-on form to send the logon credentials to the logon server. Applies only to STATIC name-value type. Default value: GET Possible values = GET, POST
__count	<Double>	Read-only	count parameter

Operations

(click to see Properties)

ADD| DELETE| UPDATE| UNSET| GET (ALL)| GET| COUNT

Some options that you can use for each operations:

- **Getting warnings in response:**NITRO allows you to get warnings in an operation by specifying the "warning" query parameter as "yes". For example, to get warnings while connecting to the Citrix ADC appliance, the URL is as follows:

  http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/login?warning=yes

  If any, the warnings are displayed in the response payload with the HTTP code "209 X-NITRO-WARNING".

- **Authenticated access for individual NITRO operations:**NITRO allows you to logon to the Citrix ADC appliance to perform individual operations. You can use this option instead of

creating a NITRO session (using the login object) and then using that session to perform all operations,

To do this, you must specify the username and password in the request header of the NITRO request as follows:

X-NITRO-USER:<username>

X-NITRO-PASS:<password>

Note: In such cases, make sure that the request header DOES not include the following:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue>

`*Note:`*

Mandatory parameters are marked in redand placeholder content is marked in <green>.

add

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/tmformssoaction

HTTP Method:POST

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Content-Type:application/json

Request Payload:

```
{"tmformssoaction":{

<b>"name":<String_value>,

</b><b>"actionurl":<String_value>,

</b><b>"userfield":<String_value>,

</b><b>"passwdfield":<String_value>,

</b><b>"ssosuccessrule":<String_value>,

</b>"namevaluepair":<String_value>,
```

```
"responsesize":<Double_value>,

"nvtype":<String_value>,

"submitmethod":<String_value>

}}
```

Response:

HTTP Status Code on Success: 201 Created HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error

delete

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/tmformssoaction/name_value<String>

HTTP Method:DELETE

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue>

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error

update

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/tmformssoaction

HTTP Method:PUT

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Content-Type:application/json

Request Payload:

```
{"tmformssoaction":{

<b>"name":<String_value>,

</b>"actionurl":<String_value>,

"userfield":<String_value>,

"passwdfield":<String_value>,

"ssosuccessrule":<String_value>,

"responsesize":<Double_value>,

"namevaluepair":<String_value>,

"nvtype":<String_value>,

"submitmethod":<String_value>

}}
```

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error

unset

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/tmformssoaction?**action=unset**

HTTP Method:POST

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Content-Type:application/json

Request Payload:

```
{"tmformssoaction":{

<b>"name":<String_value>,

</b>"responsesize":true,

"namevaluepair":true,

"nvtype":true,

"submitmethod":true

}}
```

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error

get (all)

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/tmformssoaction

Query-parameters:

attrs

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/tmformssoaction?**attrs=property-name1,property-name2**

Use this query parameter to specify the resource details that you want to retrieve.

filter

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/tmformssoaction?**filter=property-name1:property-val1,property-name2:property-val2**

Use this query-parameter to get the filtered set of tmformssoaction resources configured on Citrix ADC. Filtering can be done on any of the properties of the resource.

view

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/tmformssoaction?**view=summary**

Note: By default, the retrieved results are displayed in detail view (?view=detail).

pagination

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/tmformssoaction?**pagesize=#no;pageno=#no**

Use this query-parameter to get the tmformssoaction resources in chunks.

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "tmformssoaction": [ {

"name":<String_value>,

"actionurl":<String_value>,

"userfield":<String_value>,

"passwdfield":<String_value>,

"responsesize":<Double_value>,

"namevaluepair":<String_value>,
```

```
"nvtype":<String_value>,

"ssosuccessrule":<String_value>,

"submitmethod":<String_value>

}]}
```

get

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/tmformssoaction/name_value<String>

Query-parameters:

attrs

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/tmformssoaction/name_value<String>?**attrs=property-name1,property-name2**

Use this query parameter to specify the resource details that you want to retrieve.

view

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/tmformssoaction/name_value<String>?**view=summary**

Note: By default, the retrieved results are displayed in detail view (?view=detail).

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "tmformssoaction": [ {

"name":<String_value>,

"actionurl":<String_value>,

"userfield":<String_value>,

"passwdfield":<String_value>,

"responsesize":<Double_value>,

"namevaluepair":<String_value>,

"nvtype":<String_value>,

"ssosuccessrule":<String_value>,

"submitmethod":<String_value>

}]}
```

count

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/tmformssoaction?**count=yes**

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "tmformssoaction": [ { "__count": "#no"} ] }
```

tmglobal_auditnslogpolicy_binding

Binding object showing the auditnslogpolicy that can be bound to tmglobal.

Properties

(click to see Operations)

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
priority	<Double>	Read-write	The priority of the policy.
policyname	<String>	Read-write	The name of the policy.
gotopriorityexpression	<String>	Read-write	Applicable only to advance tmsession policy. Expression or other value specifying the next policy to be evaluated if the current policy evaluates to TRUE. Specify one of the following values: * NEXT - Evaluate the policy with the next higher priority number. * END - End policy evaluation. * An expression that evaluates to a number. If you specify an expression, the number to which it

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
			evaluates determines the next policy to evaluate, as follows: * If the expression evaluates to a higher numbered priority, the policy with that priority is evaluated next. * If the expression evaluates to the priority of the current policy, the policy with the next higher numbered priority is evaluated next. * If the expression evaluates to a priority number that is numerically higher than the highest numbered priority, policy evaluation ends. An UNDEF event is triggered if: * The expression is invalid. * The expression evaluates to a priority number that is numerically lower than the current policy's priority. * The expression evaluates to a priority number that is between the current policy's priority number (say, 30) and the highest priority number (say, 100), but does not

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
			match any configured priority number (for example, the expression evaluates to the number 85). This example assumes that the priority number increments by 10 for every successive policy, and therefore a priority number of 85 does not exist in the policy label.
bindpolicytype	<Double>	Read-only	Bound policy type.
__count	<Double>	Read-only	count parameter

Operations

(click to see Properties)

ADD:| DELETE:| GET| GET (ALL)| COUNT

Some options that you can use for each operations:

- **Getting warnings in response:**NITRO allows you to get warnings in an operation by specifying the "warning" query parameter as "yes". For example, to get warnings while connecting to the Citrix ADC appliance, the URL is as follows:

  http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/login?warning=yes

  If any, the warnings are displayed in the response payload with the HTTP code "209 X-NITRO-WARNING".

- **Authenticated access for individual NITRO operations:**NITRO allows you to logon to the Citrix ADC appliance to perform individual operations. You can use this option instead of creating a NITRO session (using the login object) and then using that session to perform all operations,

To do this, you must specify the username and password in the request header of the NITRO request as follows:

X-NITRO-USER:<username>

X-NITRO-PASS:<password>

Note: In such cases, make sure that the request header DOES not include the following:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue>

`*Note:`*

Mandatory parameters are marked in redand placeholder content is marked in <green>.

add

URL:http://<Citrix-ADC-IP-address(NSIP)/nitro/v1/config/tmglobal_auditnslogpolicy_binding

HTTP Method:PUT

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Content-Type:application/json

Request Payload:

```
{
"tmglobal_auditnslogpolicy_binding":{
"policyname":<String_value>,
"priority":<Double_value>,
"gotopriorityexpression":<String_value>
}}
```

Response:

HTTP Status Code on Success: 201 Created HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error

delete

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/tmglobal_auditnslogpolicy_binding

Query-parameters:

args

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/tmglobal_auditnslogpolicy_binding?**args=policyname:<String_value>**

HTTP Method:DELETE

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue>

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error

get

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/tmglobal_auditnslogpolicy_binding

Query-parameters:

filter

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/tmglobal_auditnslogpolicy_binding?**filter=property-name1:property-value1,property-name2:property-value2**

Use this query-parameter to get the filtered set of tmglobal_auditnslogpolicy_binding resources configured on Citrix ADC. Filtering can be done on any of the properties of the resource.

pagination

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/tmglobal_auditnslogpolicy_binding?**pagesize=#no;pageno=#no**

Use this query-parameter to get the tmglobal_auditnslogpolicy_binding resources in chunks.

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "tmglobal_auditnslogpolicy_binding": [ {

"priority":<Double_value>,

"policyname":<String_value>,

"gotopriorityexpression":<String_value>,

"bindpolicytype":<Double_value>

}]}
```

get (all)

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/tmglobal_auditnslogpolicy_binding

Query-parameters:

bulkbindings

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/tmglobal_auditnslogpolicy_binding?**bulkbindings=yes**

NITRO allows you to fetch bindings in bulk.

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "tmglobal_auditnslogpolicy_binding": [ {

"priority":<Double_value>,

"policyname":<String_value>,

"gotopriorityexpression":<String_value>,

"bindpolicytype":<Double_value>

}]}
```

count

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/tmglobal_auditnslogpolicy_binding?**count=yes**

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "tmglobal_auditnslogpolicy_binding": [ { "__count": "#no"} ] }
```

tmglobal_auditsyslogpolicy_binding

Binding object showing the auditsyslogpolicy that can be bound to tmglobal.

Properties

(click to see Operations)

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
priority	<Double>	Read-write	The priority of the policy.
policyname	<String>	Read-write	The name of the policy.
gotopriorityexpression	<String>	Read-write	Applicable only to advance tmsession policy. Expression or other value specifying the next policy to be evaluated if the current policy evaluates to TRUE. Specify one of the following values: * NEXT - Evaluate the policy with the next higher priority

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
			number. * END - End policy evaluation. * An expression that evaluates to a number. If you specify an expression, the number to which it evaluates determines the next policy to evaluate, as follows: * If the expression evaluates to a higher numbered priority, the policy with that priority is evaluated next. * If the expression evaluates to the priority of the current policy, the policy with the next higher numbered priority is evaluated next. * If the expression evaluates to a priority number that is numerically higher than the highest numbered priority, policy evaluation ends. An UNDEF event is triggered if: * The expression is invalid. * The expression evaluates to a priority number that is numerically lower than the current policy's priority. * The

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
			expression evaluates to a priority number that is between the current policy's priority number (say, 30) and the highest priority number (say, 100), but does not match any configured priority number (for example, the expression evaluates to the number 85). This example assumes that the priority number increments by 10 for every successive policy, and therefore a priority number of 85 does not exist in the policy label.
bindpolicytype	<Double>	Read-only	Bound policy type.
__count	<Double>	Read-only	count parameter

Operations

(click to see Properties)

ADD:| DELETE:| GET| GET (ALL)| COUNT

Some options that you can use for each operations:

- **Getting warnings in response:**NITRO allows you to get warnings in an operation by specifying the "warning" query parameter as "yes". For example, to get warnings while connecting to the Citrix ADC appliance, the URL is as follows:

  http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/login?warning=yes

If any, the warnings are displayed in the response payload with the HTTP code "209 X-NITRO-WARNING".

- **Authenticated access for individual NITRO operations:**NITRO allows you to logon to the Citrix ADC appliance to perform individual operations. You can use this option instead of creating a NITRO session (using the login object) and then using that session to perform all operations,

  To do this, you must specify the username and password in the request header of the NITRO request as follows:

  X-NITRO-USER:<username>

  X-NITRO-PASS:<password>

  Note: In such cases, make sure that the request header DOES not include the following:

  Cookie:NITRO_AUTH_TOKEN=<tokenvalue>

`*Note:`*

Mandatory parameters are marked in redand placeholder content is marked in <green>.

add

URL:http://<Citrix-ADC-IP-address(NSIP)/nitro/v1/config/tmglobal_auditsyslogpolicy_binding

HTTP Method:PUT

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Content-Type:application/json

Request Payload:

```
{

"tmglobal_auditsyslogpolicy_binding":{

"policyname":<String_value>,

"priority":<Double_value>,
```

```
"gotopriorityexpression":<String_value>

}}
```

Response:

HTTP Status Code on Success: 201 Created HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error

delete

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/tmglobal_auditsyslogpolicy_binding

Query-parameters:

args

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/tmglobal_auditsyslogpolicy_binding?**args=policyname:<String_value>**

HTTP Method:DELETE

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue>

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error

get

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/tmglobal_auditsyslogpolicy_binding

Query-parameters:

filter

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/tmglobal_auditsyslogpolicy_binding?**filter=property-name1:property-value1,property-name2:property-value2**

Use this query-parameter to get the filtered set of tmglobal_auditsyslogpolicy_binding resources configured on Citrix ADC. Filtering can be done on any of the properties of the resource.

pagination

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/tmglobal_auditsyslogpolicy_binding?**pagesize=#no;pageno=#no**

Use this query-parameter to get the tmglobal_auditsyslogpolicy_binding resources in chunks.

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "tmglobal_auditsyslogpolicy_binding": [ {

"priority":<Double_value>,

"policyname":<String_value>,

"gotopriorityexpression":<String_value>,

"bindpolicytype":<Double_value>

}]}
```

get (all)

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/tmglobal_auditsyslogpolicy_binding

Query-parameters:

bulkbindings

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/tmglobal_auditsyslogpolicy_binding?**bulkbindings=yes**

NITRO allows you to fetch bindings in bulk.

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "tmglobal_auditsyslogpolicy_binding": [ {

"priority":<Double_value>,

"policyname":<String_value>,

"gotopriorityexpression":<String_value>,

"bindpolicytype":<Double_value>

}]}
```

count

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/tmglobal_auditsyslogpolicy_binding?**count=yes**

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "tmglobal_auditsyslogpolicy_binding": [ { "__count": "#no"} ] }
```

tmglobal_binding

Binding object which returns the resources bound to tmglobal.

Properties

(click to see Operations)

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
tmglobal_auditnslogpolicy_binding	<tmglobal_auditnslogpolicy_binding[]>	Read-only	auditnslogpolicy that can be bound to tmglobal.
tmglobal_tmtrafficpolicy_binding	<tmglobal_tmtrafficpolicy_binding[]>	Read-only	tmtrafficpolicy that can be bound to tmglobal.
tmglobal_auditsyslogpolicy_binding	<tmglobal_auditsyslogpolicy_binding[]>	Read-only	auditsyslogpolicy that can be bound to tmglobal.

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
tmglobal_tmsessionpolicy_binding	<tmglobal_tmsessionpolicy_binding[]>	Read-only	tmsessionpolicy that can be bound to tmglobal.

Operations

(click to see Properties)

GET| GET (ALL)

Some options that you can use for each operations:

- **Getting warnings in response:**NITRO allows you to get warnings in an operation by specifying the "warning" query parameter as "yes". For example, to get warnings while connecting to the Citrix ADC appliance, the URL is as follows:

  http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/login?warning=yes

  If any, the warnings are displayed in the response payload with the HTTP code "209 X-NITRO-WARNING".

- **Authenticated access for individual NITRO operations:**NITRO allows you to logon to the Citrix ADC appliance to perform individual operations. You can use this option instead of creating a NITRO session (using the login object) and then using that session to perform all operations,

  To do this, you must specify the username and password in the request header of the NITRO request as follows:

  X-NITRO-USER:<username>

  X-NITRO-PASS:<password>

  Note: In such cases, make sure that the request header DOES not include the following:

  Cookie:NITRO_AUTH_TOKEN=<tokenvalue>

`*Note:`*

Mandatory parameters are marked in redand placeholder content is marked in <green>.

get

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/tmglobal_binding

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "tmglobal_binding": [ {

"tmglobal_tmtrafficpolicy_binding":<tmglobal_tmtrafficpolicy_binding[]_val
ue>,

"tmglobal_auditsyslogpolicy_binding":<tmglobal_auditsyslogpolicy_binding[]
_value>,

"tmglobal_tmsessionpolicy_binding":<tmglobal_tmsessionpolicy_binding[]_val
ue>,

"tmglobal_auditnslogpolicy_binding":<tmglobal_auditnslogpolicy_binding[]_v
alue>

}]}
```

get (all)

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/tmglobal_binding

Query-parameters:

bulkbindings

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/tmglobal_binding?**bulkbindings=yes**

NITRO allows you to fetch bindings in bulk.

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "tmglobal_binding": [ {

"tmglobal_tmtrafficpolicy_binding":<tmglobal_tmtrafficpolicy_binding[]_val
ue>,

"tmglobal_auditsyslogpolicy_binding":<tmglobal_auditsyslogpolicy_binding[]
_value>,

"tmglobal_tmsessionpolicy_binding":<tmglobal_tmsessionpolicy_binding[]_val
ue>,

"tmglobal_auditnslogpolicy_binding":<tmglobal_auditnslogpolicy_binding[]_v
alue>

}]}
```

tmglobal_tmsessionpolicy_binding

Binding object showing the tmsessionpolicy that can be bound to tmglobal.

Properties

(click to see Operations)

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
priority	<Double>	Read-write	The priority of the policy.
policyname	<String>	Read-write	The name of the policy.
gotopriorityexpression	<String>	Read-write	Expression specifying the priority of the next policy which will get evaluated if the current policy rule evaluates to TRUE.
bindpolicytype	<Double>	Read-only	Bound policy type.
__count	<Double>	Read-only	count parameter

Operations

(click to see Properties)

ADD:| DELETE:| GET| GET (ALL)| COUNT

Some options that you can use for each operations:

- **Getting warnings in response:**NITRO allows you to get warnings in an operation by specifying the "warning" query parameter as "yes". For example, to get warnings while connecting to the Citrix ADC appliance, the URL is as follows:

  http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/login?warning=yes

  If any, the warnings are displayed in the response payload with the HTTP code "209 X-NITRO-WARNING".

- **Authenticated access for individual NITRO operations:**NITRO allows you to logon to the Citrix ADC appliance to perform individual operations. You can use this option instead of

creating a NITRO session (using the login object) and then using that session to perform all operations,

To do this, you must specify the username and password in the request header of the NITRO request as follows:

X-NITRO-USER:<username>

X-NITRO-PASS:<password>

Note: In such cases, make sure that the request header DOES not include the following:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue>

`*Note:`*

Mandatory parameters are marked in redand placeholder content is marked in <green>.

add

URL:http://<Citrix-ADC-IP-address(NSIP)/nitro/v1/config/tmglobal_tmsessionpolicy_binding

HTTP Method:PUT

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Content-Type:application/json

Request Payload:

```
{

"tmglobal_tmsessionpolicy_binding":{

"policyname":<String_value>,

"priority":<Double_value>,

"gotopriorityexpression":<String_value>

}}
```

Response:

HTTP Status Code on Success: 201 Created HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error

delete

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/tmglobal_tmsessionpolicy_binding

Query-parameters:

args

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/tmglobal_tmsessionpolicy_binding?**args=policyname:<String_value>**

HTTP Method:DELETE

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue>

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error

get

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/tmglobal_tmsessionpolicy_binding

Query-parameters:

filter

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/tmglobal_tmsessionpolicy_binding?**filter=property-name1:property-value1,property-name2:property-value2**

Use this query-parameter to get the filtered set of tmglobal_tmsessionpolicy_binding resources configured on Citrix ADC. Filtering can be done on any of the properties of the resource.

pagination

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/tmglobal_tmsessionpolicy_binding?**pagesize=#no;pageno=#no**

Use this query-parameter to get the tmglobal_tmsessionpolicy_binding resources in chunks.

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "tmglobal_tmsessionpolicy_binding": [ {

"priority":<Double_value>,

"policyname":<String_value>,

"gotopriorityexpression":<String_value>,

"bindpolicytype":<Double_value>

}]}
```

get (all)

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/tmglobal_tmsessionpolicy_binding

Query-parameters:

bulkbindings

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/tmglobal_tmsessionpolicy_binding?**bulkbindings=yes**

NITRO allows you to fetch bindings in bulk.

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "tmglobal_tmsessionpolicy_binding": [ {

"priority":<Double_value>,

"policyname":<String_value>,

"gotopriorityexpression":<String_value>,

"bindpolicytype":<Double_value>

}]}
```

count

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/tmglobal_tmsessionpolicy_binding?**count=yes**

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "tmglobal_tmsessionpolicy_binding": [ { "__count": "#no"} ] }
```

tmglobal_tmtrafficpolicy_binding

Binding object showing the tmtrafficpolicy that can be bound to tmglobal.

Properties

(click to see Operations)

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
priority	<Double>	Read-write	The priority of the policy.
globalbindtype	<String>	Read-write	. Default value: SYSTEM_GLOBAL Possible values = SYSTEM_GLOBAL, VPN_GLOBAL, RNAT_GLOBAL
policyname	<String>	Read-write	The name of the policy.

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
gotopriorityexpression	<String>	Read-write	Applicable only to advance tmsession policy. Expression or other value specifying the next policy to be evaluated if the current policy evaluates to TRUE. Specify one of the following values: * NEXT - Evaluate the policy with the next higher priority number. * END - End policy evaluation. * An expression that evaluates to a number. If you specify an expression, the number to which it evaluates determines the next policy to evaluate, as follows: * If the expression evaluates to a higher numbered priority, the policy with that priority is evaluated next. * If the expression evaluates to the priority of the current policy, the policy with the next higher numbered priority is evaluated next. * If the expression evaluates to a priority number

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
			that is numerically higher than the highest numbered priority, policy evaluation ends. An UNDEF event is triggered if: * The expression is invalid. * The expression evaluates to a priority number that is numerically lower than the current policy's priority. * The expression evaluates to a priority number that is between the current policy's priority number (say, 30) and the highest priority number (say, 100), but does not match any configured priority number (for example, the expression evaluates to the number 85). This example assumes that the priority number increments by 10 for every successive policy, and therefore a priority number of 85 does not exist in the policy label.
type	<String>	Read-write	Bindpoint to which the policy is bound.

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
			Possible values = REQ_OVERRIDE, REQ_DEFAULT, RES_OVERRIDE, RES_DEFAULT
bindpolicytype	<Double>	Read-only	Bound policy type.
__count	<Double>	Read-only	count parameter

Operations

(click to see Properties)

ADD:| DELETE:| GET| GET (ALL)| COUNT

Some options that you can use for each operations:

- **Getting warnings in response:**NITRO allows you to get warnings in an operation by specifying the "warning" query parameter as "yes". For example, to get warnings while connecting to the Citrix ADC appliance, the URL is as follows:

  http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/login?warning=yes

  If any, the warnings are displayed in the response payload with the HTTP code "209 X-NITRO-WARNING".
- **Authenticated access for individual NITRO operations:**NITRO allows you to logon to the Citrix ADC appliance to perform individual operations. You can use this option instead of creating a NITRO session (using the login object) and then using that session to perform all operations,

  To do this, you must specify the username and password in the request header of the NITRO request as follows:

  X-NITRO-USER:<username>

  X-NITRO-PASS:<password>

  Note: In such cases, make sure that the request header DOES not include the following:

  Cookie:NITRO_AUTH_TOKEN=<tokenvalue>

`*Note:` *

Mandatory parameters are marked in redand placeholder content is marked in <green>.

add

URL:http://<Citrix-ADC-IP-address(NSIP)/nitro/v1/config/tmglobal_tmtrafficpolicy_binding

HTTP Method:PUT

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Content-Type:application/json

Request Payload:

```
{

"tmglobal_tmtrafficpolicy_binding":{

"policyname":<String_value>,

"priority":<Double_value>,

"gotopriorityexpression":<String_value>

}}
```

Response:

HTTP Status Code on Success: 201 Created HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error

delete

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/tmglobal_tmtrafficpolicy_binding

Query-parameters:

args

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/tmglobal_tmtrafficpolicy_binding?**args=policyname:<String_value>**

HTTP Method:DELETE

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue>

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error

get

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/tmglobal_tmtrafficpolicy_binding

Query-parameters:

filter

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/tmglobal_tmtrafficpolicy_binding?**filter=property-name1:property-value1,property-name2:property-value2**

Use this query-parameter to get the filtered set of tmglobal_tmtrafficpolicy_binding resources configured on Citrix ADC. Filtering can be done on any of the properties of the resource.

pagination

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/tmglobal_tmtrafficpolicy_binding?**pagesize=#no;pageno=#no**

Use this query-parameter to get the tmglobal_tmtrafficpolicy_binding resources in chunks.

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "tmglobal_tmtrafficpolicy_binding": [ {

"priority":<Double_value>,

"globalbindtype":<String_value>,

"policyname":<String_value>,

"gotopriorityexpression":<String_value>,

"type":<String_value>,

"bindpolicytype":<Double_value>

}]}
```

get (all)

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/tmglobal_tmtrafficpolicy_binding

Query-parameters:

bulkbindings

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/tmglobal_tmtrafficpolicy_binding?**bulkbindings=yes**

NITRO allows you to fetch bindings in bulk.

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "tmglobal_tmtrafficpolicy_binding": [ {

"priority":<Double_value>,

"globalbindtype":<String_value>,

"policyname":<String_value>,

"gotopriorityexpression":<String_value>,

"type":<String_value>,

"bindpolicytype":<Double_value>

}]}
```

count

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/tmglobal_tmtrafficpolicy_binding?**count=yes**

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "tmglobal_tmtrafficpolicy_binding": [ { "__count": "#no"} ] }
```

tmsamlssoprofile

Configuration for SAML sso action resource.

Properties

(click to see Operations)

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
name	<String>	Read-write	Name for the new saml single sign-on profile. Must begin with an ASCII alphanumeric or underscore (_) character, and must contain only ASCII alphanumeric, underscore, hash (#), period (.), space, colon (:), at (@), equals (=), and hyphen (-) characters. Cannot be changed after an SSO action is created.

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
			The following requirement applies only to the Citrix ADC CLI: If the name includes one or more spaces, enclose the name in double or single quotation marks (for example, "my action" or 'my action'). Minimum length = 1
samlsigningcertname	<String>	Read-write	Name of the SSL certificate that is used to Sign Assertion. Minimum length = 1
assertionconsumerserviceurl	<String>	Read-write	URL to which the assertion is to be sent. Minimum length = 1
relaystaterule	<String>	Read-write	Expression to extract relaystate to be sent along with assertion. Evaluation of this expression should return TEXT content. This is typically a targ et url to which user is redirected after the recipient

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
			validates SAML token.
sendpassword	<String>	Read-write	Option to send password in assertion. Default value: OFF Possible values = ON, OFF
samlissuername	<String>	Read-write	The name to be used in requests sent from Citrix ADC to IdP to uniquely identify Citrix ADC. Minimum length = 1
signaturealg	<String>	Read-write	Algorithm to be used to sign/verify SAML transactions. Default value: RSA-SHA256 Possible values = RSA-SHA1, RSA-SHA256
digestmethod	<String>	Read-write	Algorithm to be used to compute/verify digest for SAML transactions. Default value: SHA256 Possible values = SHA1, SHA256
audience	<String>	Read-write	Audience for which assertion sent by IdP

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
			is applicable. This is typically entity name or url that represents ServiceProvider.
nameidformat	<String>	Read-write	Format of Name Identifier sent in Assertion. Default value: transient Possible values = Unspecified, emailAddress, X509SubjectName, WindowsDomainQualifiedName, kerberos, entity, persistent, transient
nameidexpr	<String>	Read-write	Expression that will be evaluated to obtain NameIdentifier to be sent in assertion. Maximum length = 128
attribute1	<String>	Read-write	Name of attribute1 that needs to be sent in SAML Assertion.
attribute1expr	<String>	Read-write	Expression that will be evaluated to obtain attribute1's value to be sent in Assertion. Maximum length = 128

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
attribute1friendlyname	<String>	Read-write	User-Friendly Name of attribute1 that needs to be sent in SAML Assertion.
attribute1format	<String>	Read-write	Format of Attribute1 to be sent in Assertion. Possible values = URI, Basic
attribute2	<String>	Read-write	Name of attribute2 that needs to be sent in SAML Assertion.
attribute2expr	<String>	Read-write	Expression that will be evaluated to obtain attribute2's value to be sent in Assertion. Maximum length = 128
attribute2friendlyname	<String>	Read-write	User-Friendly Name of attribute2 that needs to be sent in SAML Assertion.
attribute2format	<String>	Read-write	Format of Attribute2 to be sent in Assertion. Possible values = URI, Basic
attribute3	<String>	Read-write	Name of attribute3 that needs to be sent in SAML Assertion.
attribute3expr	<String>	Read-write	Expression that will be evaluated to

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
			obtain attribute3's value to be sent in Assertion. Maximum length = 128
attribute3friendlyname	<String>	Read-write	User-Friendly Name of attribute3 that needs to be sent in SAML Assertion.
attribute3format	<String>	Read-write	Format of Attribute3 to be sent in Assertion. Possible values = URI, Basic
attribute4	<String>	Read-write	Name of attribute4 that needs to be sent in SAML Assertion.
attribute4expr	<String>	Read-write	Expression that will be evaluated to obtain attribute4's value to be sent in Assertion. Maximum length = 128
attribute4friendlyname	<String>	Read-write	User-Friendly Name of attribute4 that needs to be sent in SAML Assertion.
attribute4format	<String>	Read-write	Format of Attribute4 to be sent in Assertion. Possible values = URI, Basic

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
attribute5	<String>	Read-write	Name of attribute5 that needs to be sent in SAML Assertion.
attribute5expr	<String>	Read-write	Expression that will be evaluated to obtain attribute5's value to be sent in Assertion. Maximum length = 128
attribute5friendlyname	<String>	Read-write	User-Friendly Name of attribute5 that needs to be sent in SAML Assertion.
attribute5format	<String>	Read-write	Format of Attribute5 to be sent in Assertion. Possible values = URI, Basic
attribute6	<String>	Read-write	Name of attribute6 that needs to be sent in SAML Assertion.
attribute6expr	<String>	Read-write	Expression that will be evaluated to obtain attribute6's value to be sent in Assertion. Maximum length = 128
attribute6friendlyname	<String>	Read-write	User-Friendly Name of attribute6 that needs to be sent in SAML Assertion.

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
attribute6format	<String>	Read-write	Format of Attribute6 to be sent in Assertion. Possible values = URI, Basic
attribute7	<String>	Read-write	Name of attribute7 that needs to be sent in SAML Assertion.
attribute7expr	<String>	Read-write	Expression that will be evaluated to obtain attribute7's value to be sent in Assertion. Maximum length = 128
attribute7friendlyname	<String>	Read-write	User-Friendly Name of attribute7 that needs to be sent in SAML Assertion.
attribute7format	<String>	Read-write	Format of Attribute7 to be sent in Assertion. Possible values = URI, Basic
attribute8	<String>	Read-write	Name of attribute8 that needs to be sent in SAML Assertion.
attribute8expr	<String>	Read-write	Expression that will be evaluated to obtain attribute8's value to be sent in Assertion. Maximum length = 128

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
attribute8friendlyname	<String>	Read-write	User-Friendly Name of attribute8 that needs to be sent in SAML Assertion.
attribute8format	<String>	Read-write	Format of Attribute8 to be sent in Assertion. Possible values = URI, Basic
attribute9	<String>	Read-write	Name of attribute9 that needs to be sent in SAML Assertion.
attribute9expr	<String>	Read-write	Expression that will be evaluated to obtain attribute9's value to be sent in Assertion. Maximum length = 128
attribute9friendlyname	<String>	Read-write	User-Friendly Name of attribute9 that needs to be sent in SAML Assertion.
attribute9format	<String>	Read-write	Format of Attribute9 to be sent in Assertion. Possible values = URI, Basic
attribute10	<String>	Read-write	Name of attribute10 that needs to be sent in SAML Assertion.
attribute10expr	<String>	Read-write	Expression that will be evaluated to

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
			obtain attribute10's value to be sent in Assertion. Maximum length = 128
attribute10friendlyname	<String>	Read-write	User-Friendly Name of attribute10 that needs to be sent in SAML Assertion.
attribute10format	<String>	Read-write	Format of Attribute10 to be sent in Assertion. Possible values = URI, Basic
attribute11	<String>	Read-write	Name of attribute11 that needs to be sent in SAML Assertion.
attribute11expr	<String>	Read-write	Expression that will be evaluated to obtain attribute11's value to be sent in Assertion. Maximum length = 128
attribute11friendlyname	<String>	Read-write	User-Friendly Name of attribute11 that needs to be sent in SAML Assertion.
attribute11format	<String>	Read-write	Format of Attribute11 to be sent in Assertion. Possible values = URI, Basic

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
attribute12	<String>	Read-write	Name of attribute12 that needs to be sent in SAML Assertion.
attribute12expr	<String>	Read-write	Expression that will be evaluated to obtain attribute12's value to be sent in Assertion. Maximum length = 128
attribute12friendlyname	<String>	Read-write	User-Friendly Name of attribute12 that needs to be sent in SAML Assertion.
attribute12format	<String>	Read-write	Format of Attribute12 to be sent in Assertion. Possible values = URI, Basic
attribute13	<String>	Read-write	Name of attribute13 that needs to be sent in SAML Assertion.
attribute13expr	<String>	Read-write	Expression that will be evaluated to obtain attribute13's value to be sent in Assertion. Maximum length = 128
attribute13friendlyname	<String>	Read-write	User-Friendly Name of attribute13 that needs to be sent in SAML Assertion.

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
attribute13format	<String>	Read-write	Format of Attribute13 to be sent in Assertion. Possible values = URI, Basic
attribute14	<String>	Read-write	Name of attribute14 that needs to be sent in SAML Assertion.
attribute14expr	<String>	Read-write	Expression that will be evaluated to obtain attribute14's value to be sent in Assertion. Maximum length = 128
attribute14friendlyname	<String>	Read-write	User-Friendly Name of attribute14 that needs to be sent in SAML Assertion.
attribute14format	<String>	Read-write	Format of Attribute14 to be sent in Assertion. Possible values = URI, Basic
attribute15	<String>	Read-write	Name of attribute15 that needs to be sent in SAML Assertion.
attribute15expr	<String>	Read-write	Expression that will be evaluated to obtain attribute15's value to be sent in Assertion. Maximum length = 128

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
attribute15friendlyname	<String>	Read-write	User-Friendly Name of attribute15 that needs to be sent in SAML Assertion.
attribute15format	<String>	Read-write	Format of Attribute15 to be sent in Assertion. Possible values = URI, Basic
attribute16	<String>	Read-write	Name of attribute16 that needs to be sent in SAML Assertion.
attribute16expr	<String>	Read-write	Expression that will be evaluated to obtain attribute16's value to be sent in Assertion. Maximum length = 128
attribute16friendlyname	<String>	Read-write	User-Friendly Name of attribute16 that needs to be sent in SAML Assertion.
attribute16format	<String>	Read-write	Format of Attribute16 to be sent in Assertion. Possible values = URI, Basic
encryptassertion	<String>	Read-write	Option to encrypt assertion when Citrix ADC sends one. Default value: OFF

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
			Possible values = ON, OFF
samlspcertname	<String>	Read-write	Name of the SSL certificate of peer/ receving party using which Assertion is encrypted. Minimum length = 1
encryptionalgorithm	<String>	Read-write	Algorithm to be used to encrypt SAML assertion. Default value: AES256 Possible values = DES3, AES128, AES192, AES256
skewtime	<Double>	Read-write	This option specifies the number of minutes on either side of current time that the assertion would be valid. For example, if skewTime is 10, then assertion would be valid from (current time - 10) min to (current time + 10) min, ie 20min in all. Default value: 5
signassertion	<String>	Read-write	Option to sign portions of assertion when Citrix ADC IDP sends one. Based on the user selection,

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
			either Assertion or Response or Both or none can be signed. Default value: ASSERTION Possible values = NONE, ASSERTION, RESPONSE, BOTH
__count	<Double>	Read-only	count parameter

Operations

(click to see Properties)

ADD| DELETE| UPDATE| UNSET| GET (ALL)| GET| COUNT

Some options that you can use for each operations:

- **Getting warnings in response:**NITRO allows you to get warnings in an operation by specifying the "warning" query parameter as "yes". For example, to get warnings while connecting to the Citrix ADC appliance, the URL is as follows:

  http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/login?warning=yes

  If any, the warnings are displayed in the response payload with the HTTP code "209 X-NITRO-WARNING".

- **Authenticated access for individual NITRO operations:**NITRO allows you to logon to the Citrix ADC appliance to perform individual operations. You can use this option instead of creating a NITRO session (using the login object) and then using that session to perform all operations,

  To do this, you must specify the username and password in the request header of the NITRO request as follows:

  X-NITRO-USER:<username>

  X-NITRO-PASS:<password>

Note: In such cases, make sure that the request header DOES not include the following:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue>

`*Note:` *

Mandatory parameters are marked in redand placeholder content is marked in <green>.

add

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/tmsamlssoprofile

HTTP Method:POST

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Content-Type:application/json

Request Payload:

```
{"tmsamlssoprofile":{

<b>"name":<String_value>,

</b>"samlsigningcertname":<String_value>,

<b>"assertionconsumerserviceurl":<String_value>,

</b>"relaystaterule":<String_value>,

"sendpassword":<String_value>,

"samlissuername":<String_value>,

"signaturealg":<String_value>,

"digestmethod":<String_value>,

"audience":<String_value>,

"nameidformat":<String_value>,

"nameidexpr":<String_value>,
```

```
"attribute1":<String_value>,

"attribute1expr":<String_value>,

"attribute1friendlyname":<String_value>,

"attribute1format":<String_value>,

"attribute2":<String_value>,

"attribute2expr":<String_value>,

"attribute2friendlyname":<String_value>,

"attribute2format":<String_value>,

"attribute3":<String_value>,

"attribute3expr":<String_value>,

"attribute3friendlyname":<String_value>,

"attribute3format":<String_value>,

"attribute4":<String_value>,

"attribute4expr":<String_value>,

"attribute4friendlyname":<String_value>,

"attribute4format":<String_value>,

"attribute5":<String_value>,

"attribute5expr":<String_value>,

"attribute5friendlyname":<String_value>,

"attribute5format":<String_value>,

"attribute6":<String_value>,

"attribute6expr":<String_value>,
```

```
"attribute6friendlyname":<String_value>,

"attribute6format":<String_value>,

"attribute7":<String_value>,

"attribute7expr":<String_value>,

"attribute7friendlyname":<String_value>,

"attribute7format":<String_value>,

"attribute8":<String_value>,

"attribute8expr":<String_value>,

"attribute8friendlyname":<String_value>,

"attribute8format":<String_value>,

"attribute9":<String_value>,

"attribute9expr":<String_value>,

"attribute9friendlyname":<String_value>,

"attribute9format":<String_value>,

"attribute10":<String_value>,

"attribute10expr":<String_value>,

"attribute10friendlyname":<String_value>,

"attribute10format":<String_value>,

"attribute11":<String_value>,

"attribute11expr":<String_value>,

"attribute11friendlyname":<String_value>,

"attribute11format":<String_value>,
```

```
"attribute12":<String_value>,

"attribute12expr":<String_value>,

"attribute12friendlyname":<String_value>,

"attribute12format":<String_value>,

"attribute13":<String_value>,

"attribute13expr":<String_value>,

"attribute13friendlyname":<String_value>,

"attribute13format":<String_value>,

"attribute14":<String_value>,

"attribute14expr":<String_value>,

"attribute14friendlyname":<String_value>,

"attribute14format":<String_value>,

"attribute15":<String_value>,

"attribute15expr":<String_value>,

"attribute15friendlyname":<String_value>,

"attribute15format":<String_value>,

"attribute16":<String_value>,

"attribute16expr":<String_value>,

"attribute16friendlyname":<String_value>,

"attribute16format":<String_value>,

"encryptassertion":<String_value>,

"samlspcertname":<String_value>,
```

```
"encryptionalgorithm":<String_value>,

"skewtime":<Double_value>,

"signassertion":<String_value>

}}
```

Response:

HTTP Status Code on Success: 201 Created HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error

delete

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/tmsamlssoprofile/name_value<String>

HTTP Method:DELETE

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue>

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error

update

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/tmsamlssoprofile

HTTP Method:PUT

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Content-Type:application/json

Request Payload:

```
{"tmsamlssoprofile":{

<b>"name":<String_value>,

</b>"samlsigningcertname":<String_value>,

"assertionconsumerserviceurl":<String_value>,

"sendpassword":<String_value>,

"samlissuername":<String_value>,

"relaystaterule":<String_value>,

"signaturealg":<String_value>,

"digestmethod":<String_value>,

"audience":<String_value>,

"nameidformat":<String_value>,

"nameidexpr":<String_value>,

"attribute1":<String_value>,

"attribute1expr":<String_value>,

"attribute1friendlyname":<String_value>,

"attribute1format":<String_value>,

"attribute2":<String_value>,

"attribute2expr":<String_value>,

"attribute2friendlyname":<String_value>,

"attribute2format":<String_value>,

"attribute3":<String_value>,
```

```
"attribute3expr":<String_value>,

"attribute3friendlyname":<String_value>,

"attribute3format":<String_value>,

"attribute4":<String_value>,

"attribute4expr":<String_value>,

"attribute4friendlyname":<String_value>,

"attribute4format":<String_value>,

"attribute5":<String_value>,

"attribute5expr":<String_value>,

"attribute5friendlyname":<String_value>,

"attribute5format":<String_value>,

"attribute6":<String_value>,

"attribute6expr":<String_value>,

"attribute6friendlyname":<String_value>,

"attribute6format":<String_value>,

"attribute7":<String_value>,

"attribute7expr":<String_value>,

"attribute7friendlyname":<String_value>,

"attribute7format":<String_value>,

"attribute8":<String_value>,

"attribute8expr":<String_value>,

"attribute8friendlyname":<String_value>,
```

```
"attribute8format":<String_value>,

"attribute9":<String_value>,

"attribute9expr":<String_value>,

"attribute9friendlyname":<String_value>,

"attribute9format":<String_value>,

"attribute10":<String_value>,

"attribute10expr":<String_value>,

"attribute10friendlyname":<String_value>,

"attribute10format":<String_value>,

"attribute11":<String_value>,

"attribute11expr":<String_value>,

"attribute11friendlyname":<String_value>,

"attribute11format":<String_value>,

"attribute12":<String_value>,

"attribute12expr":<String_value>,

"attribute12friendlyname":<String_value>,

"attribute12format":<String_value>,

"attribute13":<String_value>,

"attribute13expr":<String_value>,

"attribute13friendlyname":<String_value>,

"attribute13format":<String_value>,

"attribute14":<String_value>,
```

```
"attribute14expr":<String_value>,

"attribute14friendlyname":<String_value>,

"attribute14format":<String_value>,

"attribute15":<String_value>,

"attribute15expr":<String_value>,

"attribute15friendlyname":<String_value>,

"attribute15format":<String_value>,

"attribute16":<String_value>,

"attribute16expr":<String_value>,

"attribute16friendlyname":<String_value>,

"attribute16format":<String_value>,

"encryptassertion":<String_value>,

"samlspcertname":<String_value>,

"encryptionalgorithm":<String_value>,

"skewtime":<Double_value>,

"signassertion":<String_value>

}}
```

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error

unset

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/tmsamlssoprofile?**action=unset**

HTTP Method:POST

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Content-Type:application/json

Request Payload:

```
{"tmsamlssoprofile":{

<b>"name":<String_value>,

</b>"samlsigningcertname":true,

"sendpassword":true,

"samlissuername":true,

"relaystaterule":true,

"signaturealg":true,

"digestmethod":true,

"audience":true,

"nameidformat":true,

"nameidexpr":true,

"attribute1":true,

"attribute1friendlyname":true,

"attribute1format":true,

"attribute2":true,
```

```
"attribute2friendlyname":true,

"attribute2format":true,

"attribute3":true,

"attribute3friendlyname":true,

"attribute3format":true,

"attribute4":true,

"attribute4friendlyname":true,

"attribute4format":true,

"attribute5":true,

"attribute5friendlyname":true,

"attribute5format":true,

"attribute6":true,

"attribute6friendlyname":true,

"attribute6format":true,

"attribute7":true,

"attribute7friendlyname":true,

"attribute7format":true,

"attribute8":true,

"attribute8friendlyname":true,

"attribute8format":true,

"attribute9":true,

"attribute9friendlyname":true,
```

```
"attribute9format":true,
"attribute10":true,
"attribute10friendlyname":true,
"attribute10format":true,
"attribute11":true,
"attribute11friendlyname":true,
"attribute11format":true,
"attribute12":true,
"attribute12friendlyname":true,
"attribute12format":true,
"attribute13":true,
"attribute13friendlyname":true,
"attribute13format":true,
"attribute14":true,
"attribute14friendlyname":true,
"attribute14format":true,
"attribute15":true,
"attribute15friendlyname":true,
"attribute15format":true,
"attribute16":true,
"attribute16friendlyname":true,
"attribute16format":true,
```

```
"encryptassertion":true,

"samlspcertname":true,

"encryptionalgorithm":true,

"skewtime":true,

"signassertion":true

}}
```

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error

get (all)

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/tmsamlssoprofile

Query-parameters:

attrs

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/tmsamlssoprofile?**attrs=property-name1,property-name2**

Use this query parameter to specify the resource details that you want to retrieve.

filter

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/tmsamlssoprofile?**filter=property-name1:property-val1,property-name2:property-val2**

Use this query-parameter to get the filtered set of tmsamlssoprofile resources configured on Citrix ADC. Filtering can be done on any of the properties of the resource.

view

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/tmsamlssoprofile?**view=summary**

Note: By default, the retrieved results are displayed in detail view (?view=detail).

pagination

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/tmsamlssoprofile?**pagesize=#no;pageno=#no**

Use this query-parameter to get the tmsamlssoprofile resources in chunks.

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "tmsamlssoprofile": [ {

"name":<String_value>,

"samlsigningcertname":<String_value>,

"assertionconsumerserviceurl":<String_value>,

"sendpassword":<String_value>,

"samlissuername":<String_value>,

"relaystaterule":<String_value>,

"signaturealg":<String_value>,

"digestmethod":<String_value>,
```

```
"audience":<String_value>,

"nameidformat":<String_value>,

"nameidexpr":<String_value>,

"attribute1":<String_value>,

"attribute2":<String_value>,

"attribute3":<String_value>,

"attribute4":<String_value>,

"attribute5":<String_value>,

"attribute6":<String_value>,

"attribute7":<String_value>,

"attribute8":<String_value>,

"attribute9":<String_value>,

"attribute10":<String_value>,

"attribute11":<String_value>,

"attribute12":<String_value>,

"attribute13":<String_value>,

"attribute14":<String_value>,

"attribute15":<String_value>,

"attribute16":<String_value>,

"attribute1friendlyname":<String_value>,

"attribute2friendlyname":<String_value>,

"attribute3friendlyname":<String_value>,
```

```
"attribute4friendlyname":<String_value>,

"attribute5friendlyname":<String_value>,

"attribute6friendlyname":<String_value>,

"attribute7friendlyname":<String_value>,

"attribute8friendlyname":<String_value>,

"attribute9friendlyname":<String_value>,

"attribute10friendlyname":<String_value>,

"attribute11friendlyname":<String_value>,

"attribute12friendlyname":<String_value>,

"attribute13friendlyname":<String_value>,

"attribute14friendlyname":<String_value>,

"attribute15friendlyname":<String_value>,

"attribute16friendlyname":<String_value>,

"attribute1format":<String_value>,

"attribute2format":<String_value>,

"attribute3format":<String_value>,

"attribute4format":<String_value>,

"attribute5format":<String_value>,

"attribute6format":<String_value>,

"attribute7format":<String_value>,

"attribute8format":<String_value>,

"attribute9format":<String_value>,
```

```
"attribute10format":<String_value>,

"attribute11format":<String_value>,

"attribute12format":<String_value>,

"attribute13format":<String_value>,

"attribute14format":<String_value>,

"attribute15format":<String_value>,

"attribute16format":<String_value>,

"attribute1expr":<String_value>,

"attribute2expr":<String_value>,

"attribute3expr":<String_value>,

"attribute4expr":<String_value>,

"attribute5expr":<String_value>,

"attribute6expr":<String_value>,

"attribute7expr":<String_value>,

"attribute8expr":<String_value>,

"attribute9expr":<String_value>,

"attribute10expr":<String_value>,

"attribute11expr":<String_value>,

"attribute12expr":<String_value>,

"attribute13expr":<String_value>,

"attribute14expr":<String_value>,

"attribute15expr":<String_value>,
```

```
"attribute16expr":<String_value>,

"encryptassertion":<String_value>,

"samlspcertname":<String_value>,

"encryptionalgorithm":<String_value>,

"skewtime":<Double_value>,

"signassertion":<String_value>

}]}
```

get

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/tmsamlssoprofile/name_value<String>

Query-parameters:

attrs

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/tmsamlssoprofile/name_value<String>?**attrs=property-name1,property-name2**

Use this query parameter to specify the resource details that you want to retrieve.

view

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/tmsamlssoprofile/name_value<String>?**view=summary**

Note: By default, the retrieved results are displayed in detail view (?view=detail).

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "tmsamlssoprofile": [ {

"name":<String_value>,

"samlsigningcertname":<String_value>,

"assertionconsumerserviceurl":<String_value>,

"sendpassword":<String_value>,

"samlissuername":<String_value>,

"relaystaterule":<String_value>,

"signaturealg":<String_value>,

"digestmethod":<String_value>,

"audience":<String_value>,

"nameidformat":<String_value>,

"nameidexpr":<String_value>,

"attribute1":<String_value>,

"attribute2":<String_value>,

"attribute3":<String_value>,

"attribute4":<String_value>,

"attribute5":<String_value>,
```

```
"attribute6":<String_value>,

"attribute7":<String_value>,

"attribute8":<String_value>,

"attribute9":<String_value>,

"attribute10":<String_value>,

"attribute11":<String_value>,

"attribute12":<String_value>,

"attribute13":<String_value>,

"attribute14":<String_value>,

"attribute15":<String_value>,

"attribute16":<String_value>,

"attribute1friendlyname":<String_value>,

"attribute2friendlyname":<String_value>,

"attribute3friendlyname":<String_value>,

"attribute4friendlyname":<String_value>,

"attribute5friendlyname":<String_value>,

"attribute6friendlyname":<String_value>,

"attribute7friendlyname":<String_value>,

"attribute8friendlyname":<String_value>,

"attribute9friendlyname":<String_value>,

"attribute10friendlyname":<String_value>,

"attribute11friendlyname":<String_value>,
```

```
"attribute12friendlyname":<String_value>,

"attribute13friendlyname":<String_value>,

"attribute14friendlyname":<String_value>,

"attribute15friendlyname":<String_value>,

"attribute16friendlyname":<String_value>,

"attribute1format":<String_value>,

"attribute2format":<String_value>,

"attribute3format":<String_value>,

"attribute4format":<String_value>,

"attribute5format":<String_value>,

"attribute6format":<String_value>,

"attribute7format":<String_value>,

"attribute8format":<String_value>,

"attribute9format":<String_value>,

"attribute10format":<String_value>,

"attribute11format":<String_value>,

"attribute12format":<String_value>,

"attribute13format":<String_value>,

"attribute14format":<String_value>,

"attribute15format":<String_value>,

"attribute16format":<String_value>,

"attribute1expr":<String_value>,
```

```
"attribute2expr":<String_value>,

"attribute3expr":<String_value>,

"attribute4expr":<String_value>,

"attribute5expr":<String_value>,

"attribute6expr":<String_value>,

"attribute7expr":<String_value>,

"attribute8expr":<String_value>,

"attribute9expr":<String_value>,

"attribute10expr":<String_value>,

"attribute11expr":<String_value>,

"attribute12expr":<String_value>,

"attribute13expr":<String_value>,

"attribute14expr":<String_value>,

"attribute15expr":<String_value>,

"attribute16expr":<String_value>,

"encryptassertion":<String_value>,

"samlspcertname":<String_value>,

"encryptionalgorithm":<String_value>,

"skewtime":<Double_value>,

"signassertion":<String_value>

}]}
```

count

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/tmsamlssoprofile?**count=yes**

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "tmsamlssoprofile": [ { "__count": "#no"} ] }
```

tmsessionaction

Configuration for TM session action resource.

Properties

(click to see Operations)

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
name	<String>	Read-write	Name for the session action. Must begin with an ASCII alphanumeric or underscore (_) character, and must contain only ASCII

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
			alphanumeric, underscore, hash (#), period (.), space, colon (:), at (@), equals (=), and hyphen (-) characters. Cannot be changed after a session action is created. The following requirement applies only to the Citrix ADC CLI: If the name includes one or more spaces, enclose the name in double or single quotation marks (for example, "my action" or 'my action'). Minimum length = 1
sesstimeout	<Double>	Read-write	Session timeout, in minutes. If there is no traffic during the timeout period, the user is disconnected and must reauthenticate to access intranet resources. Minimum value = 1
defaultauthorizationaction	<String>	Read-write	Allow or deny access to content for which

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
			there is no specific authorization policy. Possible values = ALLOW, DENY
sso	<String>	Read-write	Use single sign-on (SSO) to log users on to all web applications automatically after they authenticate, or pass users to the web application logon page to authenticate to each application individually. Default value: OFF Possible values = ON, OFF
ssocredential	<String>	Read-write	Use the primary or secondary authentication credentials for single sign-on (SSO). Possible values = PRIMARY, SECONDARY
ssodomain	<String>	Read-write	Domain to use for single sign-on (SSO). Minimum length = 1 Maximum length = 32

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
httponlycookie	<String>	Read-write	Allow only an HTTP session cookie, in which case the cookie cannot be accessed by scripts. Possible values = YES, NO
kcdaccount	<String>	Read-write	Kerberos constrained delegation account name. Minimum length = 1 Maximum length = 32
persistentcookie	<String>	Read-write	Enable or disable persistent SSO cookies for the traffic management (TM) session. A persistent cookie remains on the user device and is sent with each HTTP request. The cookie becomes stale if the session ends. This setting is overwritten if a traffic action sets persistent cookie to OFF. Note: If persistent cookie is enabled, make sure you set the persistent cookie validity.

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
			Possible values = ON, OFF
persistentcookievalidity	<Double>	Read-write	Integer specifying the number of minutes for which the persistent cookie remains valid. Can be set only if the persistent cookie setting is enabled. Minimum value = 1
homepage	<String>	Read-write	Web address of the home page that a user is displayed when authentication vserver is bookmarked and used to login.
__count	<Double>	Read-only	count parameter

Operations

(click to see Properties)

ADD| DELETE| UPDATE| UNSET| GET (ALL)| GET| COUNT

Some options that you can use for each operations:

- **Getting warnings in response:**NITRO allows you to get warnings in an operation by specifying the "warning" query parameter as "yes". For example, to get warnings while connecting to the Citrix ADC appliance, the URL is as follows:

  http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/login?warning=yes

  If any, the warnings are displayed in the response payload with the HTTP code "209 X-NITRO-WARNING".

- **Authenticated access for individual NITRO operations:**NITRO allows you to logon to the Citrix ADC appliance to perform individual operations. You can use this option instead of creating a NITRO session (using the login object) and then using that session to perform all operations,

  To do this, you must specify the username and password in the request header of the NITRO request as follows:

  X-NITRO-USER:<username>

  X-NITRO-PASS:<password>

  Note: In such cases, make sure that the request header DOES not include the following:

  Cookie:NITRO_AUTH_TOKEN=<tokenvalue>

`*Note:`*

Mandatory parameters are marked in redand placeholder content is marked in <green>.

add

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/tmsessionaction

HTTP Method:POST

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Content-Type:application/json

Request Payload:

```
{"tmsessionaction":{

<b>"name":<String_value>,

</b>"sesstimeout":<Double_value>,

"defaultauthorizationaction":<String_value>,

"sso":<String_value>,

"ssocredential":<String_value>,
```

```
"ssodomain":<String_value>,

"httponlycookie":<String_value>,

"kcdaccount":<String_value>,

"persistentcookie":<String_value>,

"persistentcookievalidity":<Double_value>,

"homepage":<String_value>

}}
```

Response:

HTTP Status Code on Success: 201 Created HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error

delete

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/tmsessionaction/name_value<String>

HTTP Method:DELETE

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue>

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error

update

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/tmsessionaction

HTTP Method:PUT

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Content-Type:application/json

Request Payload:

```
{"tmsessionaction":{

<b>"name":<String_value>,

</b>"sesstimeout":<Double_value>,

"defaultauthorizationaction":<String_value>,

"sso":<String_value>,

"ssocredential":<String_value>,

"ssodomain":<String_value>,

"kcdaccount":<String_value>,

"httponlycookie":<String_value>,

"persistentcookie":<String_value>,

"persistentcookievalidity":<Double_value>,

"homepage":<String_value>

}}
```

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error

unset

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/tmsessionaction?**action=unset**

HTTP Method:POST

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Content-Type:application/json

Request Payload:

```
{"tmsessionaction":{

<b>"name":<String_value>,

</b>"sesstimeout":true,

"defaultauthorizationaction":true,

"sso":true,

"ssocredential":true,

"ssodomain":true,

"kcdaccount":true,

"httponlycookie":true,

"persistentcookie":true,

"persistentcookievalidity":true,

"homepage":true

}}
```

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error

get (all)

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/tmsessionaction

Query-parameters:

attrs

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/tmsessionaction?**attrs=property-name1,property-name2**

Use this query parameter to specify the resource details that you want to retrieve.

filter

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/tmsessionaction?**filter=property-name1:property-val1,property-name2:property-val2**

Use this query-parameter to get the filtered set of tmsessionaction resources configured on Citrix ADC. Filtering can be done on any of the properties of the resource.

view

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/tmsessionaction?**view=summary**

Note: By default, the retrieved results are displayed in detail view (?view=detail).

pagination

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/tmsessionaction?**pagesize=#no;pageno=#no**

Use this query-parameter to get the tmsessionaction resources in chunks.

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "tmsessionaction": [ {

"name":<String_value>,

"sesstimeout":<Double_value>,

"defaultauthorizationaction":<String_value>,

"sso":<String_value>,

"ssocredential":<String_value>,

"ssodomain":<String_value>,

"kcdaccount":<String_value>,

"httponlycookie":<String_value>,

"persistentcookie":<String_value>,

"persistentcookievalidity":<Double_value>,

"homepage":<String_value>

}]}
```

get

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/tmsessionaction/name_value<String>

Query-parameters:

attrs

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/tmsessionaction/name_value<String>?**attrs=property-name1,property-name2**

Use this query parameter to specify the resource details that you want to retrieve.

view

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/tmsessionaction/name_value<String>?**view=summary**

Note: By default, the retrieved results are displayed in detail view (?view=detail).

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "tmsessionaction": [ {

"name":<String_value>,

"sesstimeout":<Double_value>,

"defaultauthorizationaction":<String_value>,

"sso":<String_value>,

"ssocredential":<String_value>,

"ssodomain":<String_value>,

"kcdaccount":<String_value>,

"httponlycookie":<String_value>,

"persistentcookie":<String_value>,

"persistentcookievalidity":<Double_value>,
```

```
"homepage":<String_value>

}]}
```

count

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/tmsessionaction?**count=yes**

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "tmsessionaction": [ { "__count": "#no"} ] }
```

tmsessionparameter

Configuration for session parameter resource.

Properties

(click to see Operations)

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
sesstimeout	<Double>	Read-write	Session timeout, in minutes. If there is no traffic during the

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
			timeout period, the user is disconnected and must reauthenticate to access the intranet resources. Default value: 30 Minimum value = 1
defaultauthorizationaction	<String>	Read-write	Allow or deny access to content for which there is no specific authorization policy. Default value: ALLOW Possible values = ALLOW, DENY
sso	<String>	Read-write	Log users on to all web applications automatically after they authenticate, or pass users to the web application logon page to authenticate for each application. Default value: OFF Possible values = ON, OFF
ssocredential	<String>	Read-write	Use primary or secondary authentication credentials for single sign-on.

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
			Default value: PRIMARY Possible values = PRIMARY, SECONDARY
ssodomain	<String>	Read-write	Domain to use for single sign-on. Minimum length = 1 Maximum length = 32
kcdaccount	<String>	Read-write	Kerberos constrained delegation account name. Minimum length = 1 Maximum length = 32
httponlycookie	<String>	Read-write	Allow only an HTTP session cookie, in which case the cookie cannot be accessed by scripts. Default value: YES Possible values = YES, NO
persistentcookie	<String>	Read-write	Use persistent SSO cookies for the traffic session. A persistent cookie remains on the user device and is sent with each HTTP request. The cookie becomes

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
			stale if the session ends. Default value: OFF Possible values = ON, OFF
persistentcookievalidity	<Double>	Read-write	Integer specifying the number of minutes for which the persistent cookie remains valid. Can be set only if the persistence cookie setting is enabled. Minimum value = 1
homepage	<String>	Read-write	Web address of the home page that a user is displayed when authentication vserver is bookmarked and used to login. Default value: "None"
name	<String>	Read-only	.
tmsessionpolicybindtype	<String>	Read-only	Indicates current bind type (Classic/ Advanced) for TM session policy across all bind entities. Default value: Advanced Policy Possible values = Classic Policy, Advanced Policy

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
tmsessionpolicycount	<Double>	Read-only	Count of TM session policies across all bind entities.

Operations

(click to see Properties)

UPDATE| UNSET| GET (ALL)

Some options that you can use for each operations:

- **Getting warnings in response:**NITRO allows you to get warnings in an operation by specifying the "warning" query parameter as "yes". For example, to get warnings while connecting to the Citrix ADC appliance, the URL is as follows:

  http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/login?warning=yes

  If any, the warnings are displayed in the response payload with the HTTP code "209 X-NITRO-WARNING".

- **Authenticated access for individual NITRO operations:**NITRO allows you to logon to the Citrix ADC appliance to perform individual operations. You can use this option instead of creating a NITRO session (using the login object) and then using that session to perform all operations,

  To do this, you must specify the username and password in the request header of the NITRO request as follows:

  X-NITRO-USER:<username>

  X-NITRO-PASS:<password>

  Note: In such cases, make sure that the request header DOES not include the following:

  Cookie:NITRO_AUTH_TOKEN=<tokenvalue>

`*Note:`*

Mandatory parameters are marked in redand placeholder content is marked in <green>.

update

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/tmsessionparameter

HTTP Method:PUT

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Content-Type:application/json

Request Payload:

```
{"tmsessionparameter":{

"sesstimeout":<Double_value>,

"defaultauthorizationaction":<String_value>,

"sso":<String_value>,

"ssocredential":<String_value>,

"ssodomain":<String_value>,

"kcdaccount":<String_value>,

"httponlycookie":<String_value>,

"persistentcookie":<String_value>,

"persistentcookievalidity":<Double_value>,

"homepage":<String_value>

}}
```

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error

unset

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/tmsessionparameter?**action=unset**

HTTP Method:POST

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Content-Type:application/json

Request Payload:

```
{"tmsessionparameter":{

"sesstimeout":true,

"sso":true,

"ssodomain":true,

"kcdaccount":true,

"persistentcookie":true,

"homepage":true,

"defaultauthorizationaction":true,

"ssocredential":true,

"httponlycookie":true,

"persistentcookievalidity":true

}}
```

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error

get (all)

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/tmsessionparameter

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "tmsessionparameter": [ {

"name":<String_value>,

"sesstimeout":<Double_value>,

"defaultauthorizationaction":<String_value>,

"sso":<String_value>,

"ssocredential":<String_value>,

"ssodomain":<String_value>,

"kcdaccount":<String_value>,

"httponlycookie":<String_value>,

"homepage":<String_value>,

"persistentcookie":<String_value>,

"persistentcookievalidity":<Double_value>,
```

```
"tmsessionpolicybindtype":<String_value>,

"tmsessionpolicycount":<Double_value>

}]}
```

tmsessionpolicy

Configuration for TM session policy resource.

Properties

(click to see Operations)

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
name	<String>	Read-write	Name for the session policy. Must begin with an ASCII alphanumeric or underscore (_) character, and must contain only ASCII alphanumeric, underscore, hash (#), period (.), space, colon (:), at sign (@), equal sign (=), and hyphen (-) characters. Cannot be changed after a session policy is created. The following requirement applies only to the Citrix ADC CLI: If the name includes one or more spaces, enclose the

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
			name in double or single quotation marks (for example, "my policy" or 'my policy'). Minimum length = 1
rule	<String>	Read-write	Expression, against which traffic is evaluated. Both classic and advance expressions are supported in default partition but only advance expressions in non-default partition. The following requirements apply only to the Citrix ADC CLI: * If the expression includes one or more spaces, enclose the entire expression in double quotation marks. * If the expression itself includes double quotation marks, escape the quotations by using the \ character.

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
			* Alternatively, you can use single quotation marks to enclose the rule, in which case you do not have to escape the double quotation marks.
action	<String>	Read-write	Action to be applied to connections that match this policy. Minimum length = 1
expressiontype	<String>	Read-only	Type of policy (Classic/Advanced) . Possible values = Classic Policy, Advanced Policy
hits	<Double>	Read-only	Number of hits.
gotopriorityexpression	<String>	Read-only	Expression specifying the priority of the next policy which will get evaluated if the current policy rule evaluates to TRUE.
__count	<Double>	Read-only	count parameter

Operations

(click to see Properties)

ADD| DELETE| UPDATE| UNSET| GET (ALL)| GET| COUNT

Some options that you can use for each operations:

- **Getting warnings in response:**NITRO allows you to get warnings in an operation by specifying the "warning" query parameter as "yes". For example, to get warnings while connecting to the Citrix ADC appliance, the URL is as follows:

  http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/login?warning=yes

  If any, the warnings are displayed in the response payload with the HTTP code "209 X-NITRO-WARNING".

- **Authenticated access for individual NITRO operations:**NITRO allows you to logon to the Citrix ADC appliance to perform individual operations. You can use this option instead of creating a NITRO session (using the login object) and then using that session to perform all operations,

  To do this, you must specify the username and password in the request header of the NITRO request as follows:

  X-NITRO-USER:<username>

  X-NITRO-PASS:<password>

  Note: In such cases, make sure that the request header DOES not include the following:

  Cookie:NITRO_AUTH_TOKEN=<tokenvalue>

`*Note:`*

Mandatory parameters are marked in redand placeholder content is marked in <green>.

add

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/tmsessionpolicy

HTTP Method:POST

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Content-Type:application/json

Request Payload:

```
{"tmsessionpolicy":{
```

```
<b>"name":<String_value>,

</b><b>"rule":<String_value>,

</b><b>"action":<String_value>

</b>}}
```

Response:

HTTP Status Code on Success: 201 Created HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error

delete

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/tmsessionpolicy/name_value<String>

HTTP Method:DELETE

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue>

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error

update

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/tmsessionpolicy

HTTP Method:PUT

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Content-Type:application/json

Request Payload:

```
{"tmsessionpolicy":{

<b>"name":<String_value>,

</b>"rule":<String_value>,

"action":<String_value>

}}
```

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error

unset

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/tmsessionpolicy?**action=unset**

HTTP Method:POST

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Content-Type:application/json

Request Payload:

```
{"tmsessionpolicy":{

<b>"name":<String_value>,

</b>"rule":true,

"action":true

}}
```

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error

get (all)

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/tmsessionpolicy

Query-parameters:

attrs

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/tmsessionpolicy?**attrs=property-name1,property-name2**

Use this query parameter to specify the resource details that you want to retrieve.

filter

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/tmsessionpolicy?**filter=property-name1:property-val1,property-name2:property-val2**

Use this query-parameter to get the filtered set of tmsessionpolicy resources configured on Citrix ADC. Filtering can be done on any of the properties of the resource.

view

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/tmsessionpolicy?**view=summary**

Note: By default, the retrieved results are displayed in detail view (?view=detail).

pagination

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/tmsessionpolicy?**pagesize=#no;pageno=#no**

Use this query-parameter to get the tmsessionpolicy resources in chunks.

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "tmsessionpolicy": [ {

"name":<String_value>,

"rule":<String_value>,

"action":<String_value>,

"expressiontype":<String_value>,

"hits":<Double_value>,

"gotopriorityexpression":<String_value>

}]}
```

get

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/tmsessionpolicy/name_value<String>

Query-parameters:

attrs

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/tmsessionpolicy/name_value<String>?**attrs=property-name1,property-name2**

Use this query parameter to specify the resource details that you want to retrieve.

view

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/tmsessionpolicy/name_value<String>?**view=summary**

Note: By default, the retrieved results are displayed in detail view (?view=detail).

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "tmsessionpolicy": [ {

"name":<String_value>,

"rule":<String_value>,

"action":<String_value>,

"expressiontype":<String_value>,

"hits":<Double_value>,

"gotopriorityexpression":<String_value>

}]}
```

count

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/tmsessionpolicy?**count=yes**

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "tmsessionpolicy": [ { "__count": "#no"} ] }
```

tmsessionpolicy_aaagroup_binding

Binding object showing the aaagroup that can be bound to tmsessionpolicy.

Properties

(click to see Operations)

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
boundto	<String>	Read-write	The entity name to which policy is bound.
name	<String>	Read-write	Name of the session policy for which to display detailed information. Minimum length = 1
priority	<Double>	Read-only	.

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
activepolicy	<Double>	Read-only	.
__count	<Double>	Read-only	count parameter

Operations

(click to see Properties)

GET| GET (ALL)| COUNT

Some options that you can use for each operations:

- **Getting warnings in response:**NITRO allows you to get warnings in an operation by specifying the "warning" query parameter as "yes". For example, to get warnings while connecting to the Citrix ADC appliance, the URL is as follows:

  http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/login?warning=yes

  If any, the warnings are displayed in the response payload with the HTTP code "209 X-NITRO-WARNING".
- **Authenticated access for individual NITRO operations:**NITRO allows you to logon to the Citrix ADC appliance to perform individual operations. You can use this option instead of creating a NITRO session (using the login object) and then using that session to perform all operations,

  To do this, you must specify the username and password in the request header of the NITRO request as follows:

  X-NITRO-USER:<username>

  X-NITRO-PASS:<password>

  Note: In such cases, make sure that the request header DOES not include the following:

  Cookie:NITRO_AUTH_TOKEN=<tokenvalue>

`*Note:`*

Mandatory parameters are marked in redand placeholder content is marked in <green>.

get

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/tmsessionpolicy_aaagroup_binding/name_value<String>

Query-parameters:

filter

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/tmsessionpolicy_aaagroup_binding/name_value<String>?**filter=property-name1:property-value1,property-name2:property-value2**

Use this query-parameter to get the filtered set of tmsessionpolicy_aaagroup_binding resources configured on Citrix ADC. Filtering can be done on any of the properties of the resource.

pagination

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/tmsessionpolicy_aaagroup_binding/name_value<String>?**pagesize=#no;pageno=#no**

Use this query-parameter to get the tmsessionpolicy_aaagroup_binding resources in chunks.

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "tmsessionpolicy_aaagroup_binding": [ {

"boundto":<String_value>,

"name":<String_value>,
```

```
"priority":<Double_value>,

"activepolicy":<Double_value>

}]}
```

get (all)

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/tmsessionpolicy_aaagroup_binding

Query-parameters:

bulkbindings

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/tmsessionpolicy_aaagroup_binding?**bulkbindings=yes**

NITRO allows you to fetch bindings in bulk.

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "tmsessionpolicy_aaagroup_binding": [ {

"boundto":<String_value>,

"name":<String_value>,
```

```
"priority":<Double_value>,

"activepolicy":<Double_value>

}]}
```

count

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/tmsessionpolicy_aaagroup_binding/name_value<String>?**count=yes**

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{"tmsessionpolicy_aaagroup_binding": [ { "__count": "#no"} ] }
```

tmsessionpolicy_aaauser_binding

Binding object showing the aaauser that can be bound to tmsessionpolicy.

Properties

(click to see Operations)

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
boundto	<String>	Read-write	The entity name to which policy is bound.
name	<String>	Read-write	Name of the session policy for which to display detailed information. Minimum length = 1
priority	<Double>	Read-only	.
activepolicy	<Double>	Read-only	.
__count	<Double>	Read-only	count parameter

Operations

(click to see Properties)

GET| GET (ALL)| COUNT

Some options that you can use for each operations:

- **Getting warnings in response:**NITRO allows you to get warnings in an operation by specifying the "warning" query parameter as "yes". For example, to get warnings while connecting to the Citrix ADC appliance, the URL is as follows:

  http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/login?warning=yes

  If any, the warnings are displayed in the response payload with the HTTP code "209 X-NITRO-WARNING".

- **Authenticated access for individual NITRO operations:**NITRO allows you to logon to the Citrix ADC appliance to perform individual operations. You can use this option instead of creating a NITRO session (using the login object) and then using that session to perform all operations,

  To do this, you must specify the username and password in the request header of the NITRO request as follows:

  X-NITRO-USER:<username>

X-NITRO-PASS:<password>

Note: In such cases, make sure that the request header DOES not include the following:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue>

`*Note:`*

Mandatory parameters are marked in redand placeholder content is marked in <green>.

get

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/tmsessionpolicy_aaauser_binding/name_value<String>

Query-parameters:

filter

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/tmsessionpolicy_aaauser_binding/name_value<String>?**filter=property-name1:property-value1,property-name2:property-value2**

Use this query-parameter to get the filtered set of tmsessionpolicy_aaauser_binding resources configured on Citrix ADC. Filtering can be done on any of the properties of the resource.

pagination

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/tmsessionpolicy_aaauser_binding/name_value<String>?**pagesize=#no;pageno=#no**

Use this query-parameter to get the tmsessionpolicy_aaauser_binding resources in chunks.

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "tmsessionpolicy_aaauser_binding": [ {

"boundto":<String_value>,

"name":<String_value>,

"priority":<Double_value>,

"activepolicy":<Double_value>

}]}
```

get (all)

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/tmsessionpolicy_aaauser_binding

Query-parameters:

bulkbindings

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/tmsessionpolicy_aaauser_binding?**bulkbindings=yes**

NITRO allows you to fetch bindings in bulk.

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "tmsessionpolicy_aaauser_binding": [ {

"boundto":<String_value>,

"name":<String_value>,

"priority":<Double_value>,

"activepolicy":<Double_value>

}]}
```

count

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/tmsessionpolicy_aaauser_binding/name_value<String>?**count=yes**

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{"tmsessionpolicy_aaauser_binding": [ { "__count": "#no"} ] }
```

tmsessionpolicy_authenticationvserver_binding

Binding object showing the authenticationvserver that can be bound to tmsessionpolicy.

Properties

(click to see Operations)

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
boundto	<String>	Read-write	The entity name to which policy is bound.
name	<String>	Read-write	Name of the session policy for which to display detailed information. Minimum length = 1
priority	<Double>	Read-only	.
activepolicy	<Double>	Read-only	.
__count	<Double>	Read-only	count parameter

Operations

(click to see Properties)

GET| GET (ALL)| COUNT

Some options that you can use for each operations:

- **Getting warnings in response:**NITRO allows you to get warnings in an operation by specifying the "warning" query parameter as "yes". For example, to get warnings while connecting to the Citrix ADC appliance, the URL is as follows:

  http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/login?warning=yes

  If any, the warnings are displayed in the response payload with the HTTP code "209 X-NITRO-WARNING".

- **Authenticated access for individual NITRO operations:**NITRO allows you to logon to the Citrix ADC appliance to perform individual operations. You can use this option instead of creating a NITRO session (using the login object) and then using that session to perform all operations,

  To do this, you must specify the username and password in the request header of the NITRO request as follows:

  X-NITRO-USER:<username>

  X-NITRO-PASS:<password>

  Note: In such cases, make sure that the request header DOES not include the following:

  Cookie:NITRO_AUTH_TOKEN=<tokenvalue>

`*Note:`*

Mandatory parameters are marked in redand placeholder content is marked in <green>.

get

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/tmsessionpolicy_authenticationvserver_binding/name_value<String>

Query-parameters:

filter

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/tmsessionpolicy_authenticationvserver_binding/name_value<String>?**filter=property-name1:property-value1,property-name2:property-value2**

Use this query-parameter to get the filtered set of tmsessionpolicy_authenticationvserver_binding resources configured on Citrix ADC. Filtering can be done on any of the properties of the resource.

pagination

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/tmsessionpolicy_authenticationvserver_binding/name_value<String>?**pagesize=#no;pageno=#no**

Use this query-parameter to get the tmsessionpolicy_authenticationvserver_binding resources in chunks.

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "tmsessionpolicy_authenticationvserver_binding": [ {

"boundto":<String_value>,

"name":<String_value>,

"priority":<Double_value>,

"activepolicy":<Double_value>

}]}
```

get (all)

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/tmsessionpolicy_authenticationvserver_binding

Query-parameters:

bulkbindings

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/tmsessionpolicy_authenticationvserver_binding?**bulkbindings=yes**

NITRO allows you to fetch bindings in bulk.

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "tmsessionpolicy_authenticationvserver_binding": [ {

"boundto":<String_value>,

"name":<String_value>,

"priority":<Double_value>,

"activepolicy":<Double_value>

}]}
```

count

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/tmsessionpolicy_authenticationvserver_binding/name_value<String>?**count=yes**

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{"tmsessionpolicy_authenticationvserver_binding": [ { "__count":
"#no"} ] }
```

tmsessionpolicy_binding

Binding object which returns the resources bound to tmsessionpolicy.

Properties

(click to see Operations)

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
name	<String>	Read-write	Name of the session policy for which to display detailed information. Minimum length = 1
tmsessionpolicy_authenticationvserver_binding	<tmsessionpolicy_authenticationvserver_binding[]>	Read-only	authenticationvserver that can be bound to tmsessionpolicy.
tmsessionpolicy_aaagroup_binding	<tmsessionpolicy_aaagroup_binding[]>	Read-only	aaagroup that can be bound to tmsessionpolicy.
tmsessionpolicy_aaauser_binding	<tmsessionpolicy_aaauser_binding[]>	Read-only	aaauser that can be bound to tmsessionpolicy.

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
tmsessionpolicy_tmglobal_binding	<tmsessionpolicy_tmglobal_binding[]>	Read-only	tmglobal that can be bound to tmsessionpolicy.

Operations

(click to see Properties)

GET| GET (ALL)

Some options that you can use for each operations:

- **Getting warnings in response:**NITRO allows you to get warnings in an operation by specifying the "warning" query parameter as "yes". For example, to get warnings while connecting to the Citrix ADC appliance, the URL is as follows:

  http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/login?warning=yes

  If any, the warnings are displayed in the response payload with the HTTP code "209 X-NITRO-WARNING".

- **Authenticated access for individual NITRO operations:**NITRO allows you to logon to the Citrix ADC appliance to perform individual operations. You can use this option instead of creating a NITRO session (using the login object) and then using that session to perform all operations,

  To do this, you must specify the username and password in the request header of the NITRO request as follows:

  X-NITRO-USER:<username>

  X-NITRO-PASS:<password>

  Note: In such cases, make sure that the request header DOES not include the following:

  Cookie:NITRO_AUTH_TOKEN=<tokenvalue>

`*Note:` *

Mandatory parameters are marked in redand placeholder content is marked in <green>.

get

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/tmsessionpolicy_binding/name_value<String>

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "tmsessionpolicy_binding": [ {

"name":<String_value>,

"tmsessionpolicy_aaauser_binding":<tmsessionpolicy_aaauser_binding[]_value
>,

"tmsessionpolicy_tmglobal_binding":<tmsessionpolicy_tmglobal_binding[]_val
ue>,

"tmsessionpolicy_authenticationvserver_binding":<tmsessionpolicy_authentic
ationvserver_binding[]_value>,

"tmsessionpolicy_aaagroup_binding":<tmsessionpolicy_aaagroup_binding[]_val
ue>

}]}
```

get (all)

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/tmsessionpolicy_binding

Query-parameters:

bulkbindings

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/tmsessionpolicy_binding?**bulkbindings=yes**

NITRO allows you to fetch bindings in bulk.

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "tmsessionpolicy_binding": [ {

"name":<String_value>,

"tmsessionpolicy_aaauser_binding":<tmsessionpolicy_aaauser_binding[]_value
>,

"tmsessionpolicy_tmglobal_binding":<tmsessionpolicy_tmglobal_binding[]_val
ue>,

"tmsessionpolicy_authenticationvserver_binding":<tmsessionpolicy_authentic
ationvserver_binding[]_value>,

"tmsessionpolicy_aaagroup_binding":<tmsessionpolicy_aaagroup_binding[]_val
ue>

}]}
```

tmsessionpolicy_tmglobal_binding

Binding object showing the tmglobal that can be bound to tmsessionpolicy.

Properties

(click to see Operations)

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
boundto	<String>	Read-write	The entity name to which policy is bound.
name	<String>	Read-write	Name of the session policy for which to display detailed information. Minimum length = 1
priority	<Double>	Read-only	.
activepolicy	<Double>	Read-only	.
__count	<Double>	Read-only	count parameter

Operations

(click to see Properties)

GET| GET (ALL)| COUNT

Some options that you can use for each operations:

- **Getting warnings in response:**NITRO allows you to get warnings in an operation by specifying the "warning" query parameter as "yes". For example, to get warnings while connecting to the Citrix ADC appliance, the URL is as follows:

  http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/login?warning=yes

  If any, the warnings are displayed in the response payload with the HTTP code "209 X-NITRO-WARNING".

- **Authenticated access for individual NITRO operations:**NITRO allows you to logon to the Citrix ADC appliance to perform individual operations. You can use this option instead of creating a NITRO session (using the login object) and then using that session to perform all operations,

  To do this, you must specify the username and password in the request header of the NITRO request as follows:

  X-NITRO-USER:<username>

  X-NITRO-PASS:<password>

  Note: In such cases, make sure that the request header DOES not include the following:

  Cookie:NITRO_AUTH_TOKEN=<tokenvalue>

`*Note:`*

Mandatory parameters are marked in redand placeholder content is marked in <green>.

get

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/tmsessionpolicy_tmglobal_binding/name_value<String>

Query-parameters:

filter

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/tmsessionpolicy_tmglobal_binding/name_value<String>?**filter=property-name1:property-value1,property-name2:property-value2**

Use this query-parameter to get the filtered set of tmsessionpolicy_tmglobal_binding resources configured on Citrix ADC. Filtering can be done on any of the properties of the resource.

pagination

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/tmsessionpolicy_tmglobal_binding/name_value<String>?**pagesize=#no;pageno=#no**

Use this query-parameter to get the tmsessionpolicy_tmglobal_binding resources in chunks.

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "tmsessionpolicy_tmglobal_binding": [ {

"boundto":<String_value>,

"name":<String_value>,

"priority":<Double_value>,

"activepolicy":<Double_value>

}]}
```

get (all)

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/tmsessionpolicy_tmglobal_binding

Query-parameters:

bulkbindings

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/tmsessionpolicy_tmglobal_binding?**bulkbindings=yes**

NITRO allows you to fetch bindings in bulk.

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "tmsessionpolicy_tmglobal_binding": [ {

"boundto":<String_value>,

"name":<String_value>,

"priority":<Double_value>,

"activepolicy":<Double_value>

}]}
```

count

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/tmsessionpolicy_tmglobal_binding/name_value<String>?**count=yes**

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{"tmsessionpolicy_tmglobal_binding": [ { "__count": "#no"} ] }
```

tmtrafficaction

Configuration for TM traffic action resource.

Properties

(click to see Operations)

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
name	<String>	Read-write	Name for the traffic action. Must begin with an ASCII alphanumeric or underscore (_) character, and must contain only ASCII alphanumeric, underscore, hash (#), period (.), space, colon (:), at (@), equals (=), and hyphen (-) characters. Cannot be changed after a traffic action is created. The following requirement applies only to the Citrix ADC CLI: If the name includes one or more spaces, enclose the

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
			name in double or single quotation marks (for example, "my action" or 'my action'). Minimum length = 1
apptimeout	<Double>	Read-write	Time interval, in minutes, of user inactivity after which the connection is closed. Minimum value = 1 Maximum value = 715827
sso	<String>	Read-write	Use single sign-on for the resource that the user is accessing now. Possible values = ON, OFF
formssoaction	<String>	Read-write	Name of the configured form-based single sign-on profile.
persistentcookie	<String>	Read-write	Use persistent cookies for the traffic session. A persistent cookie remains on the user device and is sent with each HTTP request. The cookie becomes stale if the session ends.

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
			Possible values = ON, OFF
initiatelogout	<String>	Read-write	Initiate logout for the traffic management (TM) session if the policy evaluates to true. The session is then terminated after two minutes. Possible values = ON, OFF
kcdaccount	<String>	Read-write	Kerberos constrained delegation account name. Default value: "None" Minimum length = 1 Maximum length = 32
samlssoprofile	<String>	Read-write	Profile to be used for doing SAML SSO to remote relying party. Minimum length = 1
forcedtimeout	<String>	Read-write	Setting to start, stop or reset TM session force timer. Possible values = START, STOP, RESET
forcedtimeoutval	<Double>	Read-write	Time interval, in minutes, for which

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
			force timer should be set.
userexpression	<String>	Read-write	expression that will be evaluated to obtain username for SingleSignOn. Maximum length = 256
passwdexpression	<String>	Read-write	expression that will be evaluated to obtain password for SingleSignOn. Maximum length = 256
__count	<Double>	Read-only	count parameter

Operations

(click to see Properties)

ADD| DELETE| UPDATE| UNSET| GET (ALL)| GET| COUNT

Some options that you can use for each operations:

- **Getting warnings in response:**NITRO allows you to get warnings in an operation by specifying the "warning" query parameter as "yes". For example, to get warnings while connecting to the Citrix ADC appliance, the URL is as follows:

  http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/login?warning=yes

  If any, the warnings are displayed in the response payload with the HTTP code "209 X-NITRO-WARNING".
- **Authenticated access for individual NITRO operations:**NITRO allows you to logon to the Citrix ADC appliance to perform individual operations. You can use this option instead of creating a NITRO session (using the login object) and then using that session to perform all operations,

To do this, you must specify the username and password in the request header of the NITRO request as follows:

X-NITRO-USER:<username>

X-NITRO-PASS:<password>

Note: In such cases, make sure that the request header DOES not include the following:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue>

`*Note:`*

Mandatory parameters are marked in redand placeholder content is marked in <green>.

add

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/tmtrafficaction

HTTP Method:POST

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Content-Type:application/json

Request Payload:

```
{"tmtrafficaction":{

<b>"name":<String_value>,

</b>"apptimeout":<Double_value>,

"sso":<String_value>,

"formssoaction":<String_value>,

"persistentcookie":<String_value>,

"initiatelogout":<String_value>,

"kcdaccount":<String_value>,
```

```
"samlssoprofile":<String_value>,

"forcedtimeout":<String_value>,

"forcedtimeoutval":<Double_value>,

"userexpression":<String_value>,

"passwdexpression":<String_value>

}}
```

Response:

HTTP Status Code on Success: 201 Created HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error

delete

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/tmtrafficaction/name_value<String>

HTTP Method:DELETE

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue>

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error

update

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/tmtrafficaction

HTTP Method:PUT

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Content-Type:application/json

Request Payload:

```
{"tmtrafficaction":{

<b>"name":<String_value>,

</b>"apptimeout":<Double_value>,

"sso":<String_value>,

"formssoaction":<String_value>,

"persistentcookie":<String_value>,

"initiatelogout":<String_value>,

"kcdaccount":<String_value>,

"samlssoprofile":<String_value>,

"forcedtimeout":<String_value>,

"forcedtimeoutval":<Double_value>,

"userexpression":<String_value>,

"passwdexpression":<String_value>

}}
```

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error

unset

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/tmtrafficaction?**action=unset**

HTTP Method:POST

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Content-Type:application/json

Request Payload:

```
{"tmtrafficaction":{

<b>"name":<String_value>,

</b>"persistentcookie":true,

"kcdaccount":true,

"forcedtimeout":true,

"userexpression":true,

"passwdexpression":true

}}
```

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error

get (all)

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/tmtrafficaction

Query-parameters:

attrs

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/tmtrafficaction?**attrs=property-name1,property-name2**

Use this query parameter to specify the resource details that you want to retrieve.

filter

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/tmtrafficaction?**filter=property-name1:property-val1,property-name2:property-val2**

Use this query-parameter to get the filtered set of tmtrafficaction resources configured on Citrix ADC. Filtering can be done on any of the properties of the resource.

view

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/tmtrafficaction?**view=summary**

Note: By default, the retrieved results are displayed in detail view (?view=detail).

pagination

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/tmtrafficaction?**pagesize=#no;pageno=#no**

Use this query-parameter to get the tmtrafficaction resources in chunks.

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "tmtrafficaction": [ {

"name":<String_value>,

"apptimeout":<Double_value>,
```

```
"sso":<String_value>,

"formssoaction":<String_value>,

"persistentcookie":<String_value>,

"initiatelogout":<String_value>,

"kcdaccount":<String_value>,

"samlssoprofile":<String_value>,

"forcedtimeout":<String_value>,

"forcedtimeoutval":<Double_value>,

"userexpression":<String_value>,

"passwdexpression":<String_value>

}]}
```

get

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/tmtrafficaction/name_value<String>

Query-parameters:

attrs

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/tmtrafficaction/name_value<String>?**attrs=property-name1,property-name2**

Use this query parameter to specify the resource details that you want to retrieve.

view

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/tmtrafficaction/name_value<String>?**view=summary**

Note: By default, the retrieved results are displayed in detail view (?view=detail).

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "tmtrafficaction": [ {

"name":<String_value>,

"apptimeout":<Double_value>,

"sso":<String_value>,

"formssoaction":<String_value>,

"persistentcookie":<String_value>,

"initiatelogout":<String_value>,

"kcdaccount":<String_value>,

"samlssoprofile":<String_value>,

"forcedtimeout":<String_value>,

"forcedtimeoutval":<Double_value>,

"userexpression":<String_value>,

"passwdexpression":<String_value>

}]}
```

count

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/tmtrafficaction?**count=yes**

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "tmtrafficaction": [ { "__count": "#no"} ] }
```

tmtrafficpolicy

Configuration for TM traffic policy resource.

Properties

(click to see Operations)

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
name	<String>	Read-write	Name for the traffic policy. Must begin with an ASCII alphanumeric or underscore (_) character, and must contain only ASCII

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
			alphanumeric, underscore, hash (#), period (.), space, colon (:), at (@), equals (=), and hyphen (-) characters. Cannot be changed after the policy is created. The following requirement applies only to the Citrix ADC CLI: If the name includes one or more spaces, enclose the name in double or single quotation marks (for example, "my policy" or 'my policy'). Minimum length = 1
rule	<String>	Read-write	Name of the Citrix ADC named expression, or an expression, that the policy uses to determine whether to apply certain action on the current traffic.
action	<String>	Read-write	Name of the action to apply to requests or connections that match this policy. Minimum length = 1

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
hits	<Double>	Read-only	Number of hits.
__count	<Double>	Read-only	count parameter

Operations

(click to see Properties)

ADD| DELETE| UPDATE| UNSET| GET (ALL)| GET| COUNT

Some options that you can use for each operations:

- **Getting warnings in response:**NITRO allows you to get warnings in an operation by specifying the "warning" query parameter as "yes". For example, to get warnings while connecting to the Citrix ADC appliance, the URL is as follows:

  http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/login?warning=yes

  If any, the warnings are displayed in the response payload with the HTTP code "209 X-NITRO-WARNING".
- **Authenticated access for individual NITRO operations:**NITRO allows you to logon to the Citrix ADC appliance to perform individual operations. You can use this option instead of creating a NITRO session (using the login object) and then using that session to perform all operations,

  To do this, you must specify the username and password in the request header of the NITRO request as follows:

  X-NITRO-USER:<username>

  X-NITRO-PASS:<password>

  Note: In such cases, make sure that the request header DOES not include the following:

  Cookie:NITRO_AUTH_TOKEN=<tokenvalue>

`*Note:` *

Mandatory parameters are marked in redand placeholder content is marked in <green>.

add

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/tmtrafficpolicy

HTTP Method:POST

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Content-Type:application/json

Request Payload:

```
{"tmtrafficpolicy":{

<b>"name":<String_value>,

</b><b>"rule":<String_value>,

</b><b>"action":<String_value>

</b>}}
```

Response:

HTTP Status Code on Success: 201 Created HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error

delete

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/tmtrafficpolicy/name_value<String>

HTTP Method:DELETE

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue>

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error

update

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/tmtrafficpolicy

HTTP Method:PUT

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Content-Type:application/json

Request Payload:

```
{"tmtrafficpolicy":{

<b>"name":<String_value>,

</b>"rule":<String_value>,

"action":<String_value>

}}
```

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error

unset

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/tmtrafficpolicy?**action=unset**

HTTP Method:POST

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Content-Type:application/json

Request Payload:

```
{"tmtrafficpolicy":{

<b>"name":<String_value>,

</b>"rule":true,

"action":true

}}
```

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error

get (all)

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/tmtrafficpolicy

Query-parameters:

attrs

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/tmtrafficpolicy?**attrs=property-name1,property-name2**

Use this query parameter to specify the resource details that you want to retrieve.

filter

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/tmtrafficpolicy?**filter=property-name1:property-val1,property-name2:property-val2**

Use this query-parameter to get the filtered set of tmtrafficpolicy resources configured on Citrix ADC. Filtering can be done on any of the properties of the resource.

view

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/tmtrafficpolicy?**view=summary**

Note: By default, the retrieved results are displayed in detail view (?view=detail).

pagination

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/tmtrafficpolicy?**pagesize=#no;pageno=#no**

Use this query-parameter to get the tmtrafficpolicy resources in chunks.

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "tmtrafficpolicy": [ {

"name":<String_value>,

"rule":<String_value>,

"action":<String_value>,

"hits":<Double_value>

}]}
```

get

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/tmtrafficpolicy/name_value<String>

Query-parameters:

attrs

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/tmtrafficpolicy/name_value<String>?**attrs=property-name1,property-name2**

Use this query parameter to specify the resource details that you want to retrieve.

view

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/tmtrafficpolicy/name_value<String>?**view=summary**

Note: By default, the retrieved results are displayed in detail view (?view=detail).

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "tmtrafficpolicy": [ {

"name":<String_value>,

"rule":<String_value>,

"action":<String_value>,

"hits":<Double_value>

}]}
```

count

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/tmtrafficpolicy?**count=yes**

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "tmtrafficpolicy": [ { "__count": "#no"} ] }
```

tmtrafficpolicy_binding

Binding object which returns the resources bound to tmtrafficpolicy.

Properties

(click to see Operations)

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
name	<String>	Read-write	Name of the traffic policy for which to display detailed information. Minimum length = 1

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
tmtrafficpolicy_csvserver_binding	<tmtrafficpolicy_csvserver_binding[]>	Read-only	csvserver that can be bound to tmtrafficpolicy.
tmtrafficpolicy_tmglobal_binding	<tmtrafficpolicy_tmglobal_binding[]>	Read-only	tmglobal that can be bound to tmtrafficpolicy.
tmtrafficpolicy_lbvserver_binding	<tmtrafficpolicy_lbvserver_binding[]>	Read-only	lbvserver that can be bound to tmtrafficpolicy.

Operations

(click to see Properties)

GET| GET (ALL)

Some options that you can use for each operations:

- **Getting warnings in response:**NITRO allows you to get warnings in an operation by specifying the "warning" query parameter as "yes". For example, to get warnings while connecting to the Citrix ADC appliance, the URL is as follows:

  http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/login?warning=yes

  If any, the warnings are displayed in the response payload with the HTTP code "209 X-NITRO-WARNING".

- **Authenticated access for individual NITRO operations:**NITRO allows you to logon to the Citrix ADC appliance to perform individual operations. You can use this option instead of creating a NITRO session (using the login object) and then using that session to perform all operations,

  To do this, you must specify the username and password in the request header of the NITRO request as follows:

  X-NITRO-USER:<username>

  X-NITRO-PASS:<password>

  Note: In such cases, make sure that the request header DOES not include the following:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue>

`*Note:` *

Mandatory parameters are marked in redand placeholder content is marked in <green>.

get

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/tmtrafficpolicy_binding/name_value<String>

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "tmtrafficpolicy_binding": [ {

"name":<String_value>,

"tmtrafficpolicy_lbvserver_binding":<tmtrafficpolicy_lbvserver_binding[]_value>,

"tmtrafficpolicy_csvserver_binding":<tmtrafficpolicy_csvserver_binding[]_value>,

"tmtrafficpolicy_tmglobal_binding":<tmtrafficpolicy_tmglobal_binding[]_value>

}]}
```

get (all)

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/tmtrafficpolicy_binding

Query-parameters:

bulkbindings

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/tmtrafficpolicy_binding?**bulkbindings=yes**

NITRO allows you to fetch bindings in bulk.

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "tmtrafficpolicy_binding": [ {

"name":<String_value>,

"tmtrafficpolicy_lbvserver_binding":<tmtrafficpolicy_lbvserver_binding[]_v
alue>,

"tmtrafficpolicy_csvserver_binding":<tmtrafficpolicy_csvserver_binding[]_v
alue>,

"tmtrafficpolicy_tmglobal_binding":<tmtrafficpolicy_tmglobal_binding[]_val
ue>

}]}
```

tmtrafficpolicy_csvserver_binding

Binding object showing the csvserver that can be bound to tmtrafficpolicy.

Properties

(click to see Operations)

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
boundto	<String>	Read-write	The entity name to which policy is bound.
name	<String>	Read-write	Name of the traffic policy for which to display detailed information. Minimum length = 1
priority	<Double>	Read-only	.
activepolicy	<Double>	Read-only	.
__count	<Double>	Read-only	count parameter

Operations

(click to see Properties)

GET| GET (ALL)| COUNT

Some options that you can use for each operations:

- **Getting warnings in response:**NITRO allows you to get warnings in an operation by specifying the "warning" query parameter as "yes". For example, to get warnings while connecting to the Citrix ADC appliance, the URL is as follows:

  http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/login?warning=yes

  If any, the warnings are displayed in the response payload with the HTTP code "209 X-NITRO-WARNING".

- **Authenticated access for individual NITRO operations:**NITRO allows you to logon to the Citrix ADC appliance to perform individual operations. You can use this option instead of creating a NITRO session (using the login object) and then using that session to perform all operations,

  To do this, you must specify the username and password in the request header of the NITRO request as follows:

  X-NITRO-USER:<username>

  X-NITRO-PASS:<password>

  Note: In such cases, make sure that the request header DOES not include the following:

  Cookie:NITRO_AUTH_TOKEN=<tokenvalue>

`*Note:`*

Mandatory parameters are marked in redand placeholder content is marked in <green>.

get

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/tmtrafficpolicy_csvserver_binding/name_value<String>

Query-parameters:

filter

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/tmtrafficpolicy_csvserver_binding/name_value<String>?**filter=property-name1:property-value1,property-name2:property-value2**

Use this query-parameter to get the filtered set of tmtrafficpolicy_csvserver_binding resources configured on Citrix ADC. Filtering can be done on any of the properties of the resource.

pagination

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/tmtrafficpolicy_csvserver_binding/name_value<String>?**pagesize=#no;pageno=#no**

Use this query-parameter to get the tmtrafficpolicy_csvserver_binding resources in chunks.

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "tmtrafficpolicy_csvserver_binding": [ {

"boundto":<String_value>,

"name":<String_value>,

"priority":<Double_value>,

"activepolicy":<Double_value>

}]}
```

get (all)

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/tmtrafficpolicy_csvserver_binding

Query-parameters:

bulkbindings

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/tmtrafficpolicy_csvserver_binding?**bulkbindings=yes**

NITRO allows you to fetch bindings in bulk.

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "tmtrafficpolicy_csvserver_binding": [ {

"boundto":<String_value>,

"name":<String_value>,

"priority":<Double_value>,

"activepolicy":<Double_value>

}]}
```

count

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/tmtrafficpolicy_csvserver_binding/name_value<String>?**count=yes**

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{"tmtrafficpolicy_csvserver_binding": [ { "__count": "#no"} ] }
```

tmtrafficpolicy_lbvserver_binding

Binding object showing the lbvserver that can be bound to tmtrafficpolicy.

Properties

(click to see Operations)

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
boundto	<String>	Read-write	The entity name to which policy is bound.
name	<String>	Read-write	Name of the traffic policy for which to display detailed information. Minimum length = 1
priority	<Double>	Read-only	.
activepolicy	<Double>	Read-only	.
__count	<Double>	Read-only	count parameter

Operations

(click to see Properties)

GET| GET (ALL)| COUNT

Some options that you can use for each operations:

- **Getting warnings in response:**NITRO allows you to get warnings in an operation by specifying the "warning" query parameter as "yes". For example, to get warnings while connecting to the Citrix ADC appliance, the URL is as follows:

  http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/login?warning=yes

  If any, the warnings are displayed in the response payload with the HTTP code "209 X-NITRO-WARNING".

- **Authenticated access for individual NITRO operations:**NITRO allows you to logon to the Citrix ADC appliance to perform individual operations. You can use this option instead of creating a NITRO session (using the login object) and then using that session to perform all operations,

  To do this, you must specify the username and password in the request header of the NITRO request as follows:

  X-NITRO-USER:<username>

  X-NITRO-PASS:<password>

  Note: In such cases, make sure that the request header DOES not include the following:

  Cookie:NITRO_AUTH_TOKEN=<tokenvalue>

`*Note:`*

Mandatory parameters are marked in redand placeholder content is marked in <green>.

get

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/tmtrafficpolicy_lbvserver_binding/name_value<String>

Query-parameters:

filter

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/tmtrafficpolicy_lbvserver_binding/name_value<String>?**filter=property-name1:property-value1,property-name2:property-value2**

Use this query-parameter to get the filtered set of tmtrafficpolicy_lbvserver_binding resources configured on Citrix ADC. Filtering can be done on any of the properties of the resource.

pagination

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/tmtrafficpolicy_lbvserver_binding/name_value<String>?**pagesize=#no;pageno=#no**

Use this query-parameter to get the tmtrafficpolicy_lbvserver_binding resources in chunks.

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "tmtrafficpolicy_lbvserver_binding": [ {

"boundto":<String_value>,

"name":<String_value>,

"priority":<Double_value>,

"activepolicy":<Double_value>

}]}
```

get (all)

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/tmtrafficpolicy_lbvserver_binding

Query-parameters:

bulkbindings

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/tmtrafficpolicy_lbvserver_binding?**bulkbindings=yes**

NITRO allows you to fetch bindings in bulk.

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "tmtrafficpolicy_lbvserver_binding": [ {

"boundto":<String_value>,

"name":<String_value>,

"priority":<Double_value>,

"activepolicy":<Double_value>

}]}
```

count

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/tmtrafficpolicy_lbvserver_binding/name_value<String>?**count=yes**

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{"tmtrafficpolicy_lbvserver_binding": [ { "__count": "#no"} ] }
```

tmtrafficpolicy_tmglobal_binding

Binding object showing the tmglobal that can be bound to tmtrafficpolicy.

Properties

(click to see Operations)

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
boundto	<String>	Read-write	The entity name to which policy is bound.
name	<String>	Read-write	Name of the traffic policy for which to display detailed information. Minimum length = 1
priority	<Double>	Read-only	.
activepolicy	<Double>	Read-only	.
__count	<Double>	Read-only	count parameter

Operations

(click to see Properties)

GET| GET (ALL)| COUNT

Some options that you can use for each operations:

- **Getting warnings in response:**NITRO allows you to get warnings in an operation by specifying the "warning" query parameter as "yes". For example, to get warnings while connecting to the Citrix ADC appliance, the URL is as follows:

  http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/login?warning=yes

  If any, the warnings are displayed in the response payload with the HTTP code "209 X-NITRO-WARNING".

- **Authenticated access for individual NITRO operations:**NITRO allows you to logon to the Citrix ADC appliance to perform individual operations. You can use this option instead of creating a NITRO session (using the login object) and then using that session to perform all operations,

  To do this, you must specify the username and password in the request header of the NITRO request as follows:

  X-NITRO-USER:<username>

  X-NITRO-PASS:<password>

  Note: In such cases, make sure that the request header DOES not include the following:

  Cookie:NITRO_AUTH_TOKEN=<tokenvalue>

`*Note:`*

Mandatory parameters are marked in redand placeholder content is marked in <green>.

get

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/tmtrafficpolicy_tmglobal_binding/name_value<String>

Query-parameters:

filter

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/tmtrafficpolicy_tmglobal_binding/name_value<String>?**filter=property-name1:property-value1,property-name2:property-value2**

Use this query-parameter to get the filtered set of tmtrafficpolicy_tmglobal_binding resources configured on Citrix ADC. Filtering can be done on any of the properties of the resource.

pagination

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/tmtrafficpolicy_tmglobal_binding/name_value<String>?**pagesize=#no;pageno=#no**

Use this query-parameter to get the tmtrafficpolicy_tmglobal_binding resources in chunks.

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "tmtrafficpolicy_tmglobal_binding": [ {

"boundto":<String_value>,

"name":<String_value>,

"priority":<Double_value>,

"activepolicy":<Double_value>

}]}
```

get (all)

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/tmtrafficpolicy_tmglobal_binding

Query-parameters:

bulkbindings

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/tmtrafficpolicy_tmglobal_binding?**bulkbindings=yes**

NITRO allows you to fetch bindings in bulk.

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "tmtrafficpolicy_tmglobal_binding": [ {

"boundto":<String_value>,

"name":<String_value>,

"priority":<Double_value>,

"activepolicy":<Double_value>

}]}
```

count

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/tmtrafficpolicy_tmglobal_binding/name_value<String>?**count=yes**

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{"tmtrafficpolicy_tmglobal_binding": [ { "__count": "#no"} ] }
```

Transform

Transform

URL Transformation configuration.

NAME	DESCRIPTION
transformaction	Configuration for transform action
transformglobal_binding	Binding object showing the resources that can be bound to transformglobal
transformglobal_transformpolicy_binding	Binding object showing the transformpolicy that can be bound to transformglobal
transformpolicy	Configuration for URL Transformation policy

NAME	DESCRIPTION
transformpolicy_binding	Binding object showing the resources that can be bound to transformpolicy
transformpolicy_csvserver_binding	Binding object showing the csvserver that can be bound to transformpolicy
transformpolicy_lbvserver_binding	Binding object showing the lbvserver that can be bound to transformpolicy
transformpolicy_transformglobal_binding	Binding object showing the transformglobal that can be bound to transformpolicy
transformpolicy_transformpolicylabel_binding	Binding object showing the transformpolicylabel that can be bound to transformpolicy
transformpolicylabel	Configuration for transform policy label
transformpolicylabel_binding	Binding object showing the resources that can be bound to transformpolicylabel
transformpolicylabel_policybinding_binding	Binding object showing the policybinding that can be bound to transformpolicylabel
transformpolicylabel_transformpolicy_binding	Binding object showing the transformpolicy that can be bound to transformpolicylabel
transformprofile	Configuration for URL Transformation profile
transformprofile_binding	Binding object showing the resources that can be bound to transformprofile
transformprofile_transformaction_binding	Binding object showing the transformaction that can be bound to transformprofile

transformaction

Configuration for transform action resource.

Properties

(click to see Operations)

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
name	<String>	Read-write	Name for the URL transformation action. Must begin with a letter, number, or the underscore character (_), and must contain only letters, numbers, and the hyphen (-), period (.) pound (#), space ( ), at (@), equals (=), colon (:), and underscore characters. Cannot be changed after the URL Transformation action is added. The following requirement applies only to the Citrix ADC CLI: If the name includes one or more spaces, enclose the name in double or single quotation marks (for example, ?my transform action? or ?my transform action). Minimum length = 1

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
profilename	<String>	Read-write	Name of the URL Transformation profile with which to associate this action. Minimum length = 1
priority	<Double>	Read-write	Positive integer specifying the priority of the action within the profile. A lower number specifies a higher priority. Must be unique within the list of actions bound to the profile. Policies are evaluated in the order of their priority numbers, and the first policy that matches is applied. Minimum value = 1 Maximum value = 2147483647
state	<String>	Read-write	Enable or disable this action. Default value: ENABLED Possible values = ENABLED, DISABLED
requrlfrom	<String>	Read-write	PCRE-format regular expression that describes the request URL pattern to be transformed.

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
			Minimum length = 1
requrlinto	<String>	Read-write	PCRE-format regular expression that describes the transformation to be performed on URLs that match the reqUrlFrom pattern. Minimum length = 1
resurlfrom	<String>	Read-write	PCRE-format regular expression that describes the response URL pattern to be transformed. Minimum length = 1
resurlinto	<String>	Read-write	PCRE-format regular expression that describes the transformation to be performed on URLs that match the resUrlFrom pattern. Minimum length = 1
cookiedomainfrom	<String>	Read-write	Pattern that matches the domain to be transformed in Set-Cookie headers. Minimum length = 1
cookiedomaininto	<String>	Read-write	PCRE-format regular expression that describes the transformation to be performed on cookie domains that match the

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
			cookieDomainFrom pattern. NOTE: The cookie domain to be transformed is extracted from the request. Minimum length = 1
comment	<String>	Read-write	Any comments to preserve information about this URL Transformation action.
continuematching	<String>	Read-only	Continue transforming using the next rule in the list. Possible values = ON, OFF
__count	<Double>	Read-only	count parameter

Operations

(click to see Properties)

ADD| DELETE| UPDATE| UNSET| GET (ALL)| GET| COUNT

Some options that you can use for each operations:

- **Getting warnings in response:**NITRO allows you to get warnings in an operation by specifying the "warning" query parameter as "yes". For example, to get warnings while connecting to the Citrix ADC appliance, the URL is as follows:

  http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/login?warning=yes

  If any, the warnings are displayed in the response payload with the HTTP code "209 X-NITRO-WARNING".

- **Authenticated access for individual NITRO operations:**NITRO allows you to logon to the Citrix ADC appliance to perform individual operations. You can use this option instead of creating a NITRO session (using the login object) and then using that session to perform all operations,

  To do this, you must specify the username and password in the request header of the NITRO request as follows:

  X-NITRO-USER:<username>

  X-NITRO-PASS:<password>

  Note: In such cases, make sure that the request header DOES not include the following:

  Cookie:NITRO_AUTH_TOKEN=<tokenvalue>

`*Note:`*

Mandatory parameters are marked in redand placeholder content is marked in <green>.

add

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/transformaction

HTTP Method:POST

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Content-Type:application/json

Request Payload:

```
{"transformaction":{

<b>"name":<String_value>,

</b><b>"profilename":<String_value>,

</b><b>"priority":<Double_value>,

</b>"state":<String_value>

}}
```

Response:

HTTP Status Code on Success: 201 Created HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error

delete

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/transformaction/name_value<String>

HTTP Method:DELETE

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue>

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error

update

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/transformaction

HTTP Method:PUT

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Content-Type:application/json

Request Payload:

```
{"transformaction":{

<b>"name":<String_value>,

</b>"priority":<Double_value>,

"requrlfrom":<String_value>,
```

```
"requrlinto":<String_value>,

"resurlfrom":<String_value>,

"resurlinto":<String_value>,

"cookiedomainfrom":<String_value>,

"cookiedomaininto":<String_value>,

"state":<String_value>,

"comment":<String_value>

}}
```

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error

unset

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/transformaction?**action=unset**

HTTP Method:POST

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Content-Type:application/json

Request Payload:

```
{"transformaction":{

<b>"name":<String_value>,

</b>"requrlfrom":true,
```

```
"requrlinto":true,

"resurlfrom":true,

"resurlinto":true,

"cookiedomainfrom":true,

"cookiedomaininto":true,

"state":true,

"comment":true

}}
```

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error

get (all)

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/transformaction

Query-parameters:

attrs

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/transformaction?**attrs=property-name1,property-name2**

Use this query parameter to specify the resource details that you want to retrieve.

filter

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/transformaction?**filter=property-name1:property-val1,property-name2:property-val2**

Use this query-parameter to get the filtered set of transformaction resources configured on Citrix ADC. Filtering can be done on any of the properties of the resource.

view

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/transformaction?**view=summary**

Note: By default, the retrieved results are displayed in detail view (?view=detail).

pagination

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/transformaction?**pagesize=#no;pageno=#no**

Use this query-parameter to get the transformaction resources in chunks.

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "transformaction": [ {

"name":<String_value>,

"profilename":<String_value>,

"priority":<Double_value>,

"requrlfrom":<String_value>,

"requrlinto":<String_value>,

"resurlfrom":<String_value>,

"resurlinto":<String_value>,
```

```
"cookiedomainfrom":<String_value>,

"cookiedomaininto":<String_value>,

"continuematching":<String_value>,

"state":<String_value>,

"comment":<String_value>

}]}
```

get

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/transformaction/name_value<String>

Query-parameters:

attrs

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/transformaction/name_value<String>?**attrs=property-name1,property-name2**

Use this query parameter to specify the resource details that you want to retrieve.

view

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/transformaction/name_value<String>?**view=summary**

Note: By default, the retrieved results are displayed in detail view (?view=detail).

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "transformaction": [ {

"name":<String_value>,

"profilename":<String_value>,

"priority":<Double_value>,

"requrlfrom":<String_value>,

"requrlinto":<String_value>,

"resurlfrom":<String_value>,

"resurlinto":<String_value>,

"cookiedomainfrom":<String_value>,

"cookiedomaininto":<String_value>,

"continuematching":<String_value>,

"state":<String_value>,

"comment":<String_value>

}]}
```

count

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/transformaction?**count=yes**

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "transformaction": [ { "__count": "#no"} ] }
```

transformglobal_binding

Binding object which returns the resources bound to transformglobal.

Properties

(click to see Operations)

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
transformglobal_transformpolicy_binding	<transformglobal_transformpolicy_binding []>	Read-only	transformpolicy that can be bound to transformglobal.

Operations

(click to see Properties)

GET| GET (ALL)

Some options that you can use for each operations:

- **Getting warnings in response:**NITRO allows you to get warnings in an operation by specifying the "warning" query parameter as "yes". For example, to get warnings while connecting to the Citrix ADC appliance, the URL is as follows:

  http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/login?warning=yes

If any, the warnings are displayed in the response payload with the HTTP code "209 X-NITRO-WARNING".

- **Authenticated access for individual NITRO operations:**NITRO allows you to logon to the Citrix ADC appliance to perform individual operations. You can use this option instead of creating a NITRO session (using the login object) and then using that session to perform all operations,

  To do this, you must specify the username and password in the request header of the NITRO request as follows:

  X-NITRO-USER:<username>

  X-NITRO-PASS:<password>

  Note: In such cases, make sure that the request header DOES not include the following:

  Cookie:NITRO_AUTH_TOKEN=<tokenvalue>

`*Note:`*

Mandatory parameters are marked in redand placeholder content is marked in <green>.

get

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/transformglobal_binding

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "transformglobal_binding": [ {

"transformglobal_transformpolicy_binding":<transformglobal_transformpolicy
_binding[]_value>

}]}
```

get (all)

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/transformglobal_binding

Query-parameters:

bulkbindings

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/transformglobal_binding?**bulkbindings=yes**

NITRO allows you to fetch bindings in bulk.

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "transformglobal_binding": [ {

"transformglobal_transformpolicy_binding":<transformglobal_transformpolicy
_binding[]_value>
```

```
}]}
```

transformglobal_transformpolicy_binding

Binding object showing the transformpolicy that can be bound to transformglobal.

Properties

(click to see Operations)

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
priority	<Double>	Read-write	Specifies the priority of the policy.
globalbindtype	<String>	Read-write	. Default value: SYSTEM_GLOBAL Possible values = SYSTEM_GLOBAL, VPN_GLOBAL, RNAT_GLOBAL
policyname	<String>	Read-write	Name of the transform policy.
labelname	<String>	Read-write	Name of the policy label to invoke if the current policy evaluates to TRUE, the invoke parameter is set, and the label type is Policy Label.
gotopriorityexpression	<String>	Read-write	Expression specifying the priority of the next policy which will get evaluated if the

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
			current policy rule evaluates to TRUE.
invoke	<Boolean>	Read-write	If the current policy evaluates to TRUE, terminate evaluation of policies bound to the current policy label, and then forwards the request or response to the specified virtual server or evaluates the specified policy label.
type	<String>	Read-write	Specifies the bind point to which to bind the policy. Available settings function as follows: * REQ_OVERRIDE. Request override. Binds the policy to the priority request queue. * REQ_DEFAULT. Binds the policy to the default request queue. Possible values = REQ_OVERRIDE, REQ_DEFAULT
labeltype	<String>	Read-write	Type of invocation. Available settings function as follows: * reqvserver - Send the request to the specified request virtual server. *

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
			resvserver - Send the response to the specified response virtual server. * policylabel - Invoke the specified policy label. Possible values = reqvserver, resvserver, policylabel
numpol	<Double>	Read-only	The number of policies bound to the bindpoint.
flowtype	<Double>	Read-only	flowtype of the bound transform policy.
__count	<Double>	Read-only	count parameter

Operations

(click to see Properties)

ADD:| DELETE:| GET| GET (ALL)| COUNT

Some options that you can use for each operations:

- **Getting warnings in response:**NITRO allows you to get warnings in an operation by specifying the "warning" query parameter as "yes". For example, to get warnings while connecting to the Citrix ADC appliance, the URL is as follows:

  http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/login?warning=yes

  If any, the warnings are displayed in the response payload with the HTTP code "209 X-NITRO-WARNING".
- **Authenticated access for individual NITRO operations:**NITRO allows you to logon to the Citrix ADC appliance to perform individual operations. You can use this option instead of

creating a NITRO session (using the login object) and then using that session to perform all operations,

To do this, you must specify the username and password in the request header of the NITRO request as follows:

X-NITRO-USER:<username>

X-NITRO-PASS:<password>

Note: In such cases, make sure that the request header DOES not include the following:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue>

`*Note:`*

Mandatory parameters are marked in redand placeholder content is marked in <green>.

add

URL:http://<Citrix-ADC-IP-address(NSIP)/nitro/v1/config/transformglobal_transformpolicy_binding

HTTP Method:PUT

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Content-Type:application/json

Request Payload:

```
{

"transformglobal_transformpolicy_binding":{

<b>"policyname":<String_value>,

</b><b>"priority":<Double_value>,

</b>"gotopriorityexpression":<String_value>,

"type":<String_value>,

"invoke":<Boolean_value>,
```

```
"labeltype":<String_value>,

"labelname":<String_value>

}}
```

Response:

HTTP Status Code on Success: 201 Created HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error

delete

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/transformglobal_transformpolicy_binding

Query-parameters:

args

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/transformglobal_transformpolicy_binding?**args=policyname:<String_value>,type:<String_value>,priority:<Double_value>**

HTTP Method:DELETE

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue>

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error

get

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/transformglobal_transformpolicy_binding

Query-parameters:

filter

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/transformglobal_transformpolicy_binding?**filter=property-name1:property-value1,property-name2:property-value2**

Use this query-parameter to get the filtered set of transformglobal_transformpolicy_binding resources configured on Citrix ADC. Filtering can be done on any of the properties of the resource.

pagination

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/transformglobal_transformpolicy_binding?**pagesize=#no;pageno=#no**

Use this query-parameter to get the transformglobal_transformpolicy_binding resources in chunks.

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "transformglobal_transformpolicy_binding": [ {

"priority":<Double_value>,

"globalbindtype":<String_value>,

"policyname":<String_value>,

"labelname":<String_value>,

"gotopriorityexpression":<String_value>,

"invoke":<Boolean_value>,
```

```
"type":<String_value>,

"labeltype":<String_value>,

"numpol":<Double_value>,

"flowtype":<Double_value>

}]}
```

get (all)

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/transformglobal_transformpolicy_binding

Query-parameters:

bulkbindings

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/transformglobal_transformpolicy_binding?**bulkbindings=yes**

NITRO allows you to fetch bindings in bulk.

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "transformglobal_transformpolicy_binding": [ {
```

```
"priority":<Double_value>,

"globalbindtype":<String_value>,

"policyname":<String_value>,

"labelname":<String_value>,

"gotopriorityexpression":<String_value>,

"invoke":<Boolean_value>,

"type":<String_value>,

"labeltype":<String_value>,

"numpol":<Double_value>,

"flowtype":<Double_value>

}]}
```

count

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/transformglobal_transformpolicy_binding?**count=yes**

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "transformglobal_transformpolicy_binding": [ { "__count": "#no"} ] }
```

transformpolicy

Configuration for URL Transformation policy resource.

Properties

(click to see Operations)

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
name	<String>	Read-write	Name for the URL Transformation policy. Must begin with a letter, number, or the underscore character (_), and must contain only letters, numbers, and the hyphen (-), period (.) pound (#), space ( ), at (@), equals (=), colon (:), and underscore characters. Can be changed after the URL Transformation policy is added. The following requirement applies only to the Citrix ADC CLI:

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
			If the name includes one or more spaces, enclose the name in double or single quotation marks (for example, ?my transform policy? or ?my transform policy). Minimum length = 1
rule	<String>	Read-write	Expression, or name of a named expression, against which to evaluate traffic. The following requirements apply only to the Citrix ADC CLI: * If the expression includes blank spaces, the entire expression must be enclosed in double quotation marks. * If the expression itself includes double quotation marks, you must escape the quotations by using the \ character.

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
			* Alternatively, you can use single quotation marks to enclose the rule, in which case you do not have to escape the double quotation marks.
profilename	<String>	Read-write	Name of the URL Transformation profile to use to transform requests and responses that match the policy. Minimum length = 1
comment	<String>	Read-write	Any comments to preserve information about this URL Transformation policy.
logaction	<String>	Read-write	Log server to use to log connections that match this policy.
newname	<String>	Read-write	New name for the policy. Must begin with a letter, number, or the underscore character (_), and must contain only letters, numbers, and the hyphen (-), period (.) pound (#), space ( ), at (@), equals (=), colon (:), and underscore characters.

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
			The following requirement applies only to the Citrix ADC CLI: If the name includes one or more spaces, enclose the name in double or single quotation marks (for example, ?my transform policy? or ?my transform policy). Minimum length = 1
hits	<Double>	Read-only	Number of hits.
isdefault	<Boolean>	Read-only	A value of true is returned if it is a default transform policy.
__count	<Double>	Read-only	count parameter

Operations

(click to see Properties)

ADD| DELETE| UPDATE| UNSET| RENAME| GET (ALL)| GET| COUNT

Some options that you can use for each operations:

- **Getting warnings in response:**NITRO allows you to get warnings in an operation by specifying the "warning" query parameter as "yes". For example, to get warnings while connecting to the Citrix ADC appliance, the URL is as follows:

  http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/login?warning=yes

If any, the warnings are displayed in the response payload with the HTTP code "209 X-NITRO-WARNING".

- **Authenticated access for individual NITRO operations:**NITRO allows you to logon to the Citrix ADC appliance to perform individual operations. You can use this option instead of creating a NITRO session (using the login object) and then using that session to perform all operations,

  To do this, you must specify the username and password in the request header of the NITRO request as follows:

  X-NITRO-USER:<username>

  X-NITRO-PASS:<password>

  Note: In such cases, make sure that the request header DOES not include the following:

  Cookie:NITRO_AUTH_TOKEN=<tokenvalue>

`*Note:`*

Mandatory parameters are marked in redand placeholder content is marked in <green>.

add

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/transformpolicy

HTTP Method:POST

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Content-Type:application/json

Request Payload:

```
{"transformpolicy":{

<b>"name":<String_value>,

</b><b>"rule":<String_value>,

</b><b>"profilename":<String_value>,
```

```
</b>"comment":<String_value>,

"logaction":<String_value>

}}
```

Response:

HTTP Status Code on Success: 201 Created HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error

delete

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/transformpolicy/name_value<String>

HTTP Method:DELETE

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue>

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error

update

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/transformpolicy

HTTP Method:PUT

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Content-Type:application/json

Request Payload:

```
{"transformpolicy":{
```

```
<b>"name":<String_value>,

</b>"rule":<String_value>,

"profilename":<String_value>,

"comment":<String_value>,

"logaction":<String_value>

}}
```

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error

unset

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/transformpolicy?**action=unset**

HTTP Method:POST

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Content-Type:application/json

Request Payload:

```
{"transformpolicy":{

<b>"name":<String_value>,

</b>"comment":true,

"logaction":true

}}
```

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error

rename

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/transformpolicy?**action=rename**

HTTP Method:POST

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Content-Type:application/json

Request Payload:

```
{"transformpolicy":{

<b>"name":<String_value>,

</b><b>"newname":<String_value>

</b>}}
```

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error

get (all)

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/transformpolicy

Query-parameters:

attrs

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/transformpolicy?**attrs=property-name1,property-name2**

Use this query parameter to specify the resource details that you want to retrieve.

filter

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/transformpolicy?**filter=property-name1:property-val1,property-name2:property-val2**

Use this query-parameter to get the filtered set of transformpolicy resources configured on Citrix ADC. Filtering can be done on any of the properties of the resource.

view

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/transformpolicy?**view=summary**

Note: By default, the retrieved results are displayed in detail view (?view=detail).

pagination

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/transformpolicy?**pagesize=#no;pageno=#no**

Use this query-parameter to get the transformpolicy resources in chunks.

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "transformpolicy": [ {
```

```
"name":<String_value>,

"rule":<String_value>,

"profilename":<String_value>,

"hits":<Double_value>,

"comment":<String_value>,

"logaction":<String_value>,

"isdefault":<Boolean_value>

}]}
```

get

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/transformpolicy/name_value<String>

Query-parameters:

attrs

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/transformpolicy/name_value<String>?**attrs=property-name1,property-name2**

Use this query parameter to specify the resource details that you want to retrieve.

view

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/transformpolicy/name_value<String>?**view=summary**

Note: By default, the retrieved results are displayed in detail view (?view=detail).

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "transformpolicy": [ {

"name":<String_value>,

"rule":<String_value>,

"profilename":<String_value>,

"hits":<Double_value>,

"comment":<String_value>,

"logaction":<String_value>,

"isdefault":<Boolean_value>

}]}
```

count

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/transformpolicy?**count=yes**

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "transformpolicy": [ { "__count": "#no"} ] }
```

transformpolicylabel

Configuration for transform policy label resource.

Properties

(click to see Operations)

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
labelname	<String>	Read-write	Name for the policy label. Must begin with a letter, number, or the underscore character (_), and must contain only letters, numbers, and the hyphen (-), period (.) pound (#), space ( ), at (@), equals (=), colon (:), and underscore characters. Can be changed after the URL Transformation policy label is added. The following requirement applies only to the Citrix ADC CLI: If the name includes one or more

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
			spaces, enclose the name in double or single quotation marks (for example, ?my transform policylabel? or ?my transform policylabel).
policylabeltype	<String>	Read-write	Types of transformations allowed by the policies bound to the label. For URL transformation, always http_req (HTTP Request). Possible values = http_req
newname	<String>	Read-write	New name for the policy label. Must begin with a letter, number, or the underscore character (_), and must contain only letters, numbers, and the hyphen (-), period (.) pound (#), space ( ), at (@), equals (=), colon (:), and underscore characters.

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
			The following requirement applies only to the Citrix ADC CLI: If the name includes one or more spaces, enclose the name in double or single quotation marks (for example, ?my transform policylabel? or ?my transform policylabel). Minimum length = 1
numpol	<Double>	Read-only	Number of polices bound to label.
hits	<Double>	Read-only	Number of times policy label was invoked.
priority	<Double>	Read-only	Specifies the priority of the policy.
gotopriorityexpression	<String>	Read-only	Expression specifying the priority of the next policy which will get evaluated if the current policy rule evaluates to TRUE.

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
labeltype	<String>	Read-only	Type of invocation. Available settings function as follows: * reqvserver - Forward the request to the specified request virtual server. * policylabel - Invoke the specified policy label. Possible values = reqvserver, policylabel
invoke_labelname	<String>	Read-only	Name of the policy label.
description	<String>	Read-only	Description of the policylabel.
__count	<Double>	Read-only	count parameter

Operations

(click to see Properties)

ADD| DELETE| RENAME| GET (ALL)| GET| COUNT

Some options that you can use for each operations:

- **Getting warnings in response:**NITRO allows you to get warnings in an operation by specifying the "warning" query parameter as "yes". For example, to get warnings while connecting to the Citrix ADC appliance, the URL is as follows:

  http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/login?warning=yes

  If any, the warnings are displayed in the response payload with the HTTP code "209 X-NITRO-WARNING".

- **Authenticated access for individual NITRO operations:**NITRO allows you to logon to the Citrix ADC appliance to perform individual operations. You can use this option instead of creating a NITRO session (using the login object) and then using that session to perform all operations,

  To do this, you must specify the username and password in the request header of the NITRO request as follows:

  X-NITRO-USER:<username>

  X-NITRO-PASS:<password>

  Note: In such cases, make sure that the request header DOES not include the following:

  Cookie:NITRO_AUTH_TOKEN=<tokenvalue>

`*Note:`*

Mandatory parameters are marked in redand placeholder content is marked in <green>.

add

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/transformpolicylabel

HTTP Method:POST

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Content-Type:application/json

Request Payload:

```
{"transformpolicylabel":{

<b>"labelname":<String_value>,

</b><b>"policylabeltype":<String_value>

</b>}}
```

Response:

HTTP Status Code on Success: 201 Created HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error

delete

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/transformpolicylabel/labelname_value<String>

HTTP Method:DELETE

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue>

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error

rename

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/transformpolicylabel?**action=rename**

HTTP Method:POST

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Content-Type:application/json

Request Payload:

```
{"transformpolicylabel":{

<b>"labelname":<String_value>,

</b><b>"newname":<String_value>

</b>}}
```

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error

get (all)

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/transformpolicylabel

Query-parameters:

attrs

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/transformpolicylabel?**attrs=property-name1,property-name2**

Use this query parameter to specify the resource details that you want to retrieve.

filter

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/transformpolicylabel?**filter=property-name1:property-val1,property-name2:property-val2**

Use this query-parameter to get the filtered set of transformpolicylabel resources configured on Citrix ADC. Filtering can be done on any of the properties of the resource.

view

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/transformpolicylabel?**view=summary**

Note: By default, the retrieved results are displayed in detail view (?view=detail).

pagination

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/transformpolicylabel?**pagesize=#no;pageno=#no**

Use this query-parameter to get the transformpolicylabel resources in chunks.

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "transformpolicylabel": [ {

"labelname":<String_value>,

"policylabeltype":<String_value>,

"numpol":<Double_value>,

"hits":<Double_value>,

"priority":<Double_value>,

"gotopriorityexpression":<String_value>,

"labeltype":<String_value>,

"invoke_labelname":<String_value>,

"description":<String_value>

}]}
```

get

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/transformpolicylabel/labelname_value<String>

Query-parameters:

attrs

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/transformpolicylabel/labelname_value<String>?**attrs=property-name1,property-name2**

Use this query parameter to specify the resource details that you want to retrieve.

view

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/transformpolicylabel/labelname_value<String>?**view=summary**

Note: By default, the retrieved results are displayed in detail view (?view=detail).

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "transformpolicylabel": [ {

"labelname":<String_value>,

"policylabeltype":<String_value>,

"numpol":<Double_value>,

"hits":<Double_value>,

"priority":<Double_value>,

"gotopriorityexpression":<String_value>,
```

```
"labeltype":<String_value>,

"invoke_labelname":<String_value>,

"description":<String_value>

}]}
```

count

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/transformpolicylabel?**count=yes**

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "transformpolicylabel": [ { "__count": "#no"} ] }
```

transformpolicylabel_binding

Binding object which returns the resources bound to transformpolicylabel.

Properties

(click to see Operations)

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
labelname	<String>	Read-write	Name for the policy label. Must begin with a letter, number, or the underscore character (_), and must contain only letters, numbers, and the hyphen (-), period (.) pound (#), space ( ), at (@), equals (=), colon (:), and underscore characters. Can be changed after the URL Transformation policy label is added. The following requirement applies only to the Citrix ADC CLI: If the name includes one or more spaces, enclose the name in double or single quotation marks (for example, ?my transform policylabel? or ?my transform policylabel).
transformpolicylabel_policybinding_binding	<transformpolicylabel_policybinding_binding[]>	Read-only	policybinding that can be bound to transformpolicylabel.

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
transformpolicylabel_transformpolicy_binding	<transformpolicylabel_transformpolicy_binding[]>	Read-only	transformpolicy that can be bound to transformpolicylabel.

Operations

(click to see Properties)

GET| GET (ALL)

Some options that you can use for each operations:

- **Getting warnings in response:**NITRO allows you to get warnings in an operation by specifying the "warning" query parameter as "yes". For example, to get warnings while connecting to the Citrix ADC appliance, the URL is as follows:

  http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/login?warning=yes

  If any, the warnings are displayed in the response payload with the HTTP code "209 X-NITRO-WARNING".

- **Authenticated access for individual NITRO operations:**NITRO allows you to logon to the Citrix ADC appliance to perform individual operations. You can use this option instead of creating a NITRO session (using the login object) and then using that session to perform all operations,

  To do this, you must specify the username and password in the request header of the NITRO request as follows:

  X-NITRO-USER:<username>

  X-NITRO-PASS:<password>

  Note: In such cases, make sure that the request header DOES not include the following:

  Cookie:NITRO_AUTH_TOKEN=<tokenvalue>

`*Note:` *

Mandatory parameters are marked in redand placeholder content is marked in <green>.

get

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/transformpolicylabel_binding/labelname_value<String>

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "transformpolicylabel_binding": [ {

"labelname":<String_value>,

"transformpolicylabel_policybinding_binding":<transformpolicylabel_policybinding_binding[]_value>,

"transformpolicylabel_transformpolicy_binding":<transformpolicylabel_transformpolicy_binding[]_value>

}]}
```

get (all)

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/transformpolicylabel_binding

Query-parameters:

bulkbindings

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/transformpolicylabel_binding?**bulkbindings=yes**

NITRO allows you to fetch bindings in bulk.

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "transformpolicylabel_binding": [ {

"labelname":<String_value>,

"transformpolicylabel_policybinding_binding":<transformpolicylabel_policybinding_binding[]_value>,

"transformpolicylabel_transformpolicy_binding":<transformpolicylabel_transformpolicy_binding[]_value>

}]}
```

transformpolicylabel_policybinding_binding

Binding object showing the policybinding that can be bound to transformpolicylabel.

Properties

(click to see Operations)

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
priority	<Double>	Read-write	Specifies the priority of the policy.
policyname	<String>	Read-write	Name of the URL Transformation policy to bind to the policy label.
labelname	<String>	Read-write	Name of the URL Transformation policy label to which to bind the policy.
invoke_labelname	<String>	Read-write	Name of the policy label.
gotopriorityexpression	<String>	Read-write	Expression specifying the priority of the next policy which will get evaluated if the current policy rule evaluates to TRUE.
invoke	<Boolean>	Read-write	If the current policy evaluates to TRUE, terminate evaluation of policies bound to the current policy label, and then forward the request to the specified virtual server or evaluate the specified policy label.
labeltype	<String>	Read-write	Type of invocation. Available settings function as follows: * reqvserver - Forward the request to the

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
			specified request virtual server. * policylabel - Invoke the specified policy label. Possible values = reqvserver, policylabel
__count	<Double>	Read-write	count parameter

Operations

(click to see Properties)

GET| GET (ALL)| COUNT

Some options that you can use for each operations:

- **Getting warnings in response:**NITRO allows you to get warnings in an operation by specifying the "warning" query parameter as "yes". For example, to get warnings while connecting to the Citrix ADC appliance, the URL is as follows:

  http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/login?warning=yes

  If any, the warnings are displayed in the response payload with the HTTP code "209 X-NITRO-WARNING".

- **Authenticated access for individual NITRO operations:**NITRO allows you to logon to the Citrix ADC appliance to perform individual operations. You can use this option instead of creating a NITRO session (using the login object) and then using that session to perform all operations,

  To do this, you must specify the username and password in the request header of the NITRO request as follows:

  X-NITRO-USER:<username>

  X-NITRO-PASS:<password>

  Note: In such cases, make sure that the request header DOES not include the following:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue>

`*Note:`*

Mandatory parameters are marked in redand placeholder content is marked in <green>.

get

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/transformpolicylabel_policybinding_binding/labelname_value<String>

Query-parameters:

filter

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/transformpolicylabel_policybinding_binding/labelname_value<String>?**filter=property-name1:property-value1,property-name2:property-value2**

Use this query-parameter to get the filtered set of transformpolicylabel_policybinding_binding resources configured on Citrix ADC. Filtering can be done on any of the properties of the resource.

pagination

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/transformpolicylabel_policybinding_binding/labelname_value<String>?**pagesize=#no;pageno=#no**

Use this query-parameter to get the transformpolicylabel_policybinding_binding resources in chunks.

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "transformpolicylabel_policybinding_binding": [ {

"priority":<Double_value>,

"policyname":<String_value>,

"labelname":<String_value>,

"invoke_labelname":<String_value>,

"gotopriorityexpression":<String_value>,

"invoke":<Boolean_value>,

"labeltype":<String_value>

}]}
```

get (all)

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/transformpolicylabel_policybinding_binding

Query-parameters:

bulkbindings

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/transformpolicylabel_policybinding_binding?**bulkbindings=yes**

NITRO allows you to fetch bindings in bulk.

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "transformpolicylabel_policybinding_binding": [ {

"priority":<Double_value>,

"policyname":<String_value>,

"labelname":<String_value>,

"invoke_labelname":<String_value>,

"gotopriorityexpression":<String_value>,

"invoke":<Boolean_value>,

"labeltype":<String_value>

}]}
```

count

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/transformpolicylabel_policybinding_binding/labelname_value<String>?**count=yes**

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{"transformpolicylabel_policybinding_binding": [ { "__count": "#no"} ] }
```

transformpolicylabel_transformpolicy_binding

Binding object showing the transformpolicy that can be bound to transformpolicylabel.

Properties

(click to see Operations)

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
priority	<Double>	Read-write	Specifies the priority of the policy.
policyname	<String>	Read-write	Name of the URL Transformation policy to bind to the policy label.
labelname	<String>	Read-write	Name of the URL Transformation policy label to which to bind the policy.
invoke_labelname	<String>	Read-write	Name of the policy label.
gotopriorityexpression	<String>	Read-write	Expression specifying the priority of the next policy which will get

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
			evaluated if the current policy rule evaluates to TRUE.
invoke	<Boolean>	Read-write	If the current policy evaluates to TRUE, terminate evaluation of policies bound to the current policy label, and then forward the request to the specified virtual server or evaluate the specified policy label.
labeltype	<String>	Read-write	Type of invocation. Available settings function as follows: * reqvserver - Forward the request to the specified request virtual server. * policylabel - Invoke the specified policy label. Possible values = reqvserver, policylabel
__count	<Double>	Read-write	count parameter

Operations

(click to see Properties)

ADD:| DELETE:| GET| GET (ALL)| COUNT

Some options that you can use for each operations:

- **Getting warnings in response:**NITRO allows you to get warnings in an operation by specifying the "warning" query parameter as "yes". For example, to get warnings while connecting to the Citrix ADC appliance, the URL is as follows:

  http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/login?warning=yes

  If any, the warnings are displayed in the response payload with the HTTP code "209 X-NITRO-WARNING".

- **Authenticated access for individual NITRO operations:**NITRO allows you to logon to the Citrix ADC appliance to perform individual operations. You can use this option instead of creating a NITRO session (using the login object) and then using that session to perform all operations,

  To do this, you must specify the username and password in the request header of the NITRO request as follows:

  X-NITRO-USER:<username>

  X-NITRO-PASS:<password>

  Note: In such cases, make sure that the request header DOES not include the following:

  Cookie:NITRO_AUTH_TOKEN=<tokenvalue>

`*Note:`*

Mandatory parameters are marked in redand placeholder content is marked in <green>.

add

URL:http://<Citrix-ADC-IP-address(NSIP)/nitro/v1/config/transformpolicylabel_transformpolicy_binding

HTTP Method:PUT

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Content-Type:application/json

Request Payload:

```
{
```

```
"transformpolicylabel_transformpolicy_binding":{

<b>"labelname":<String_value>,

</b><b>"policyname":<String_value>,

</b><b>"priority":<Double_value>,

</b>"gotopriorityexpression":<String_value>,

"invoke":<Boolean_value>,

"labeltype":<String_value>,

"invoke_labelname":<String_value>

}}
```

Response:

HTTP Status Code on Success: 201 Created HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error

delete

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/transformpolicylabel_transformpolicy_binding/labelname_value<String>

Query-parameters:

args

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/transformpolicylabel_transformpolicy_binding/labelname_value<String>?**args=policyname:<String_value>,priority:<Double_value>**

HTTP Method:DELETE

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue>

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error

get

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/transformpolicylabel_transformpolicy_binding/labelname_value<String>

Query-parameters:

filter

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/transformpolicylabel_transformpolicy_binding/labelname_value<String>?**filter=property-name1:property-value1,property-name2:property-value2**

Use this query-parameter to get the filtered set of transformpolicylabel_transformpolicy_binding resources configured on Citrix ADC. Filtering can be done on any of the properties of the resource.

pagination

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/transformpolicylabel_transformpolicy_binding/labelname_value<String>?**pagesize=#no;pageno=#no**

Use this query-parameter to get the transformpolicylabel_transformpolicy_binding resources in chunks.

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "transformpolicylabel_transformpolicy_binding": [ {

"priority":<Double_value>,

"policyname":<String_value>,

"labelname":<String_value>,

"invoke_labelname":<String_value>,

"gotopriorityexpression":<String_value>,

"invoke":<Boolean_value>,

"labeltype":<String_value>

}]}
```

get (all)

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/transformpolicylabel_transformpolicy_binding

Query-parameters:

bulkbindings

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/transformpolicylabel_transformpolicy_binding?**bulkbindings=yes**

NITRO allows you to fetch bindings in bulk.

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "transformpolicylabel_transformpolicy_binding": [ {

"priority":<Double_value>,

"policyname":<String_value>,

"labelname":<String_value>,

"invoke_labelname":<String_value>,

"gotopriorityexpression":<String_value>,

"invoke":<Boolean_value>,

"labeltype":<String_value>

}]}
```

count

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/transformpolicylabel_transformpolicy_binding/labelname_value<String>?**count=yes**

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{"transformpolicylabel_transformpolicy_binding": [ { "__count": "#no"} ] }
```

transformpolicy_binding

Binding object which returns the resources bound to transformpolicy.

Properties

(click to see Operations)

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
name	<String>	Read-write	Name of the URL Transformation policy. Minimum length = 1
transformpolicy_lbvserver_binding	<transformpolicy_lbvserver_binding[]>	Read-only	lbvserver that can be bound to transformpolicy.
transformpolicy_transformglobal_binding	<transformpolicy_transformglobal_binding[]>	Read-only	transformglobal that can be bound to transformpolicy.
transformpolicy_csvserver_binding	<transformpolicy_csvserver_binding[]>	Read-only	csvserver that can be bound to transformpolicy.
transformpolicy_transformpolicylabel_binding	<transformpolicy_transformpolicylabel_binding[]>	Read-only	transformpolicylabel that can be bound to transformpolicy.

Operations

(click to see Properties)

GET| GET (ALL)

Some options that you can use for each operations:

- **Getting warnings in response:**NITRO allows you to get warnings in an operation by specifying the "warning" query parameter as "yes". For example, to get warnings while connecting to the Citrix ADC appliance, the URL is as follows:

  http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/login?warning=yes

  If any, the warnings are displayed in the response payload with the HTTP code "209 X-NITRO-WARNING".

- **Authenticated access for individual NITRO operations:**NITRO allows you to logon to the Citrix ADC appliance to perform individual operations. You can use this option instead of creating a NITRO session (using the login object) and then using that session to perform all operations,

  To do this, you must specify the username and password in the request header of the NITRO request as follows:

  X-NITRO-USER:<username>

  X-NITRO-PASS:<password>

  Note: In such cases, make sure that the request header DOES not include the following:

  Cookie:NITRO_AUTH_TOKEN=<tokenvalue>

`*Note:`*

Mandatory parameters are marked in redand placeholder content is marked in <green>.

get

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/transformpolicy_binding/name_value<String>

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "transformpolicy_binding": [ {

"name":<String_value>,

"transformpolicy_transformglobal_binding":<transformpolicy_transformglobal
_binding[]_value>,

"transformpolicy_lbvserver_binding":<transformpolicy_lbvserver_binding[]_v
alue>,

"transformpolicy_csvserver_binding":<transformpolicy_csvserver_binding[]_v
alue>,

"transformpolicy_transformpolicylabel_binding":<transformpolicy_transformp
olicylabel_binding[]_value>

}]}
```

get (all)

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/transformpolicy_binding

Query-parameters:

bulkbindings

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/transformpolicy_binding?**bulkbindings=yes**

NITRO allows you to fetch bindings in bulk.

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "transformpolicy_binding": [ {

"name":<String_value>,

"transformpolicy_transformglobal_binding":<transformpolicy_transformglobal
_binding[]_value>,

"transformpolicy_lbvserver_binding":<transformpolicy_lbvserver_binding[]_v
alue>,

"transformpolicy_csvserver_binding":<transformpolicy_csvserver_binding[]_v
alue>,

"transformpolicy_transformpolicylabel_binding":<transformpolicy_transformp
olicylabel_binding[]_value>

}]}
```

transformpolicy_csvserver_binding

Binding object showing the csvserver that can be bound to transformpolicy.

Properties

(click to see Operations)

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
name	<String>	Read-write	Name of the URL Transformation policy. Minimum length = 1
boundto	<String>	Read-write	Location where policy is bound.
priority	<Double>	Read-only	Specifies the priority of the policy.
gotopriorityexpression	<String>	Read-only	Expression specifying the priority of the next policy which will get evaluated if the current policy rule evaluates to TRUE.
labeltype	<String>	Read-only	Type of policy label invocation. Possible values = reqvserver, policylabel
activepolicy	<Integer>	Read-only	Indicates whether policy is bound or not.
labelname	<String>	Read-only	Name of the label to invoke if the current policy rule evaluates to TRUE.
__count	<Double>	Read-only	count parameter

Operations

(click to see Properties)

GET| GET (ALL)| COUNT

Some options that you can use for each operations:

- **Getting warnings in response:**NITRO allows you to get warnings in an operation by specifying the "warning" query parameter as "yes". For example, to get warnings while connecting to the Citrix ADC appliance, the URL is as follows:

  http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/login?warning=yes

  If any, the warnings are displayed in the response payload with the HTTP code "209 X-NITRO-WARNING".

- **Authenticated access for individual NITRO operations:**NITRO allows you to logon to the Citrix ADC appliance to perform individual operations. You can use this option instead of creating a NITRO session (using the login object) and then using that session to perform all operations,

  To do this, you must specify the username and password in the request header of the NITRO request as follows:

  X-NITRO-USER:<username>

  X-NITRO-PASS:<password>

  Note: In such cases, make sure that the request header DOES not include the following:

  Cookie:NITRO_AUTH_TOKEN=<tokenvalue>

`*Note:`*

Mandatory parameters are marked in redand placeholder content is marked in <green>.

get

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/transformpolicy_csvserver_binding/name_value<String>

Query-parameters:

filter

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/transformpolicy_csvserver_binding/name_value<String>?**filter=property-name1:property-value1,property-name2:property-value2**

Use this query-parameter to get the filtered set of transformpolicy_csvserver_binding resources configured on Citrix ADC. Filtering can be done on any of the properties of the resource.

pagination

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/transformpolicy_csvserver_binding/name_value<String>?**pagesize=#no;pageno=#no**

Use this query-parameter to get the transformpolicy_csvserver_binding resources in chunks.

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "transformpolicy_csvserver_binding": [ {

"name":<String_value>,

"boundto":<String_value>,

"priority":<Double_value>,

"gotopriorityexpression":<String_value>,

"labeltype":<String_value>,

"activepolicy":<Integer_value>,

"labelname":<String_value>

}]}
```

get (all)

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/transformpolicy_csvserver_binding

Query-parameters:

bulkbindings

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/transformpolicy_csvserver_binding?**bulkbindings=yes**

NITRO allows you to fetch bindings in bulk.

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "transformpolicy_csvserver_binding": [ {

"name":<String_value>,

"boundto":<String_value>,

"priority":<Double_value>,

"gotopriorityexpression":<String_value>,

"labeltype":<String_value>,

"activepolicy":<Integer_value>,
```

```
"labelname":<String_value>

}]}
```

count

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/transformpolicy_csvserver_binding/name_value<String>?**count=yes**

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{"transformpolicy_csvserver_binding": [ { "__count": "#no"} ] }
```

transformpolicy_lbvserver_binding

Binding object showing the lbvserver that can be bound to transformpolicy.

Properties

(click to see Operations)

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
name	<String>	Read-write	Name of the URL Transformation policy. Minimum length = 1
boundto	<String>	Read-write	Location where policy is bound.
priority	<Double>	Read-only	Specifies the priority of the policy.
gotopriorityexpression	<String>	Read-only	Expression specifying the priority of the next policy which will get evaluated if the current policy rule evaluates to TRUE.
labeltype	<String>	Read-only	Type of policy label invocation. Possible values = reqvserver, policylabel
activepolicy	<Integer>	Read-only	Indicates whether policy is bound or not.
labelname	<String>	Read-only	Name of the label to invoke if the current policy rule evaluates to TRUE.
__count	<Double>	Read-only	count parameter

Operations

(click to see Properties)

GET| GET (ALL)| COUNT

Some options that you can use for each operations:

- **Getting warnings in response:**NITRO allows you to get warnings in an operation by specifying the "warning" query parameter as "yes". For example, to get warnings while connecting to the Citrix ADC appliance, the URL is as follows:

  http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/login?warning=yes

  If any, the warnings are displayed in the response payload with the HTTP code "209 X-NITRO-WARNING".

- **Authenticated access for individual NITRO operations:**NITRO allows you to logon to the Citrix ADC appliance to perform individual operations. You can use this option instead of creating a NITRO session (using the login object) and then using that session to perform all operations,

  To do this, you must specify the username and password in the request header of the NITRO request as follows:

  X-NITRO-USER:<username>

  X-NITRO-PASS:<password>

  Note: In such cases, make sure that the request header DOES not include the following:

  Cookie:NITRO_AUTH_TOKEN=<tokenvalue>

`*Note:`*

Mandatory parameters are marked in redand placeholder content is marked in <green>.

get

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/transformpolicy_lbvserver_binding/name_value<String>

Query-parameters:

filter

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/transformpolicy_lbvserver_binding/name_value<String>?**filter=property-name1:property-value1,property-name2:property-value2**

Use this query-parameter to get the filtered set of transformpolicy_lbvserver_binding resources configured on Citrix ADC. Filtering can be done on any of the properties of the resource.

pagination

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/transformpolicy_lbvserver_binding/name_value<String>?**pagesize=#no;pageno=#no**

Use this query-parameter to get the transformpolicy_lbvserver_binding resources in chunks.

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "transformpolicy_lbvserver_binding": [ {

"name":<String_value>,

"boundto":<String_value>,

"priority":<Double_value>,

"gotopriorityexpression":<String_value>,

"labeltype":<String_value>,

"activepolicy":<Integer_value>,

"labelname":<String_value>

}]}
```

get (all)

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/transformpolicy_lbvserver_binding

Query-parameters:

bulkbindings

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/transformpolicy_lbvserver_binding?**bulkbindings=yes**

NITRO allows you to fetch bindings in bulk.

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "transformpolicy_lbvserver_binding": [ {

"name":<String_value>,

"boundto":<String_value>,

"priority":<Double_value>,

"gotopriorityexpression":<String_value>,

"labeltype":<String_value>,

"activepolicy":<Integer_value>,
```

```
"labelname":<String_value>

}]}
```

count

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/transformpolicy_lbvserver_binding/name_value<String>?**count=yes**

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{"transformpolicy_lbvserver_binding": [ { "__count": "#no"} ] }
```

transformpolicy_transformglobal_binding

Binding object showing the transformglobal that can be bound to transformpolicy.

Properties

(click to see Operations)

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
name	<String>	Read-write	Name of the URL Transformation policy. Minimum length = 1
boundto	<String>	Read-write	Location where policy is bound.
priority	<Double>	Read-only	Specifies the priority of the policy.
gotopriorityexpression	<String>	Read-only	Expression specifying the priority of the next policy which will get evaluated if the current policy rule evaluates to TRUE.
labeltype	<String>	Read-only	Type of policy label invocation. Possible values = reqvserver, policylabel
activepolicy	<Integer>	Read-only	Indicates whether policy is bound or not.
labelname	<String>	Read-only	Name of the label to invoke if the current policy rule evaluates to TRUE.
__count	<Double>	Read-only	count parameter

Operations

(click to see Properties)

GET| GET (ALL)| COUNT

Some options that you can use for each operations:

- **Getting warnings in response:**NITRO allows you to get warnings in an operation by specifying the "warning" query parameter as "yes". For example, to get warnings while connecting to the Citrix ADC appliance, the URL is as follows:

  http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/login?warning=yes

  If any, the warnings are displayed in the response payload with the HTTP code "209 X-NITRO-WARNING".

- **Authenticated access for individual NITRO operations:**NITRO allows you to logon to the Citrix ADC appliance to perform individual operations. You can use this option instead of creating a NITRO session (using the login object) and then using that session to perform all operations,

  To do this, you must specify the username and password in the request header of the NITRO request as follows:

  X-NITRO-USER:<username>

  X-NITRO-PASS:<password>

  Note: In such cases, make sure that the request header DOES not include the following:

  Cookie:NITRO_AUTH_TOKEN=<tokenvalue>

`*Note:`*

Mandatory parameters are marked in redand placeholder content is marked in <green>.

get

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/transformpolicy_transformglobal_binding/name_value<String>

Query-parameters:

filter

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/transformpolicy_transformglobal_binding/name_value<String>?**filter=property-name1:property-value1,property-name2:property-value2**

Use this query-parameter to get the filtered set of transformpolicy_transformglobal_binding resources configured on Citrix ADC. Filtering can be done on any of the properties of the resource.

pagination

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/transformpolicy_transformglobal_binding/name_value<String>?**pagesize=#no;pageno=#no**

Use this query-parameter to get the transformpolicy_transformglobal_binding resources in chunks.

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "transformpolicy_transformglobal_binding": [ {

"name":<String_value>,

"boundto":<String_value>,

"priority":<Double_value>,

"gotopriorityexpression":<String_value>,

"labeltype":<String_value>,

"activepolicy":<Integer_value>,

"labelname":<String_value>

}]}
```

get (all)

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/transformpolicy_transformglobal_binding

Query-parameters:

bulkbindings

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/transformpolicy_transformglobal_binding?**bulkbindings=yes**

NITRO allows you to fetch bindings in bulk.

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "transformpolicy_transformglobal_binding": [ {

"name":<String_value>,

"boundto":<String_value>,

"priority":<Double_value>,

"gotopriorityexpression":<String_value>,

"labeltype":<String_value>,

"activepolicy":<Integer_value>,
```

```
"labelname":<String_value>

}]}
```

count

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/transformpolicy_transformglobal_binding/name_value<String>?**count=yes**

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{"transformpolicy_transformglobal_binding": [ { "__count": "#no"} ] }
```

transformpolicy_transformpolicylabel_binding

Binding object showing the transformpolicylabel that can be bound to transformpolicy.

Properties

(click to see Operations)

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
name	<String>	Read-write	Name of the URL Transformation policy. Minimum length = 1
boundto	<String>	Read-write	Location where policy is bound.
priority	<Double>	Read-only	Specifies the priority of the policy.
gotopriorityexpression	<String>	Read-only	Expression specifying the priority of the next policy which will get evaluated if the current policy rule evaluates to TRUE.
labeltype	<String>	Read-only	Type of policy label invocation. Possible values = reqvserver, policylabel
activepolicy	<Integer>	Read-only	Indicates whether policy is bound or not.
labelname	<String>	Read-only	Name of the label to invoke if the current policy rule evaluates to TRUE.
__count	<Double>	Read-only	count parameter

Operations

(click to see Properties)

GET| GET (ALL)| COUNT

Some options that you can use for each operations:

- **Getting warnings in response:**NITRO allows you to get warnings in an operation by specifying the "warning" query parameter as "yes". For example, to get warnings while connecting to the Citrix ADC appliance, the URL is as follows:

  http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/login?warning=yes

  If any, the warnings are displayed in the response payload with the HTTP code "209 X-NITRO-WARNING".

- **Authenticated access for individual NITRO operations:**NITRO allows you to logon to the Citrix ADC appliance to perform individual operations. You can use this option instead of creating a NITRO session (using the login object) and then using that session to perform all operations,

  To do this, you must specify the username and password in the request header of the NITRO request as follows:

  X-NITRO-USER:<username>

  X-NITRO-PASS:<password>

  Note: In such cases, make sure that the request header DOES not include the following:

  Cookie:NITRO_AUTH_TOKEN=<tokenvalue>

`*Note:`*

Mandatory parameters are marked in redand placeholder content is marked in <green>.

get

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/transformpolicy_transformpolicylabel_binding/name_value<String>

Query-parameters:

filter

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/transformpolicy_transformpolicylabel_binding/name_value<String>?**filter=property-name1:property-value1,property-name2:property-value2**

Use this query-parameter to get the filtered set of transformpolicy_transformpolicylabel_binding resources configured on Citrix ADC. Filtering can be done on any of the properties of the resource.

pagination

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/transformpolicy_transformpolicylabel_binding/name_value<String>?**pagesize=#no;pageno=#no**

Use this query-parameter to get the transformpolicy_transformpolicylabel_binding resources in chunks.

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "transformpolicy_transformpolicylabel_binding": [ {

"name":<String_value>,

"boundto":<String_value>,

"priority":<Double_value>,

"gotopriorityexpression":<String_value>,

"labeltype":<String_value>,

"activepolicy":<Integer_value>,

"labelname":<String_value>

}]}
```

get (all)

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/transformpolicy_transformpolicylabel_binding

Query-parameters:

bulkbindings

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/transformpolicy_transformpolicylabel_binding?**bulkbindings=yes**

NITRO allows you to fetch bindings in bulk.

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "transformpolicy_transformpolicylabel_binding": [ {

"name":<String_value>,

"boundto":<String_value>,

"priority":<Double_value>,

"gotopriorityexpression":<String_value>,

"labeltype":<String_value>,
```

```
"activepolicy":<Integer_value>,

"labelname":<String_value>

}]}
```

count

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/transformpolicy_transformpolicylabel_binding/name_value<String>?**count=yes**

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{"transformpolicy_transformpolicylabel_binding": [ { "__count": "#no"} ] }
```

transformprofile

Configuration for URL Transformation profile resource.

Properties

(click to see Operations)

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
name	<String>	Read-write	Name for the URL transformation profile. Must begin with a letter, number, or the underscore character (_), and must contain only letters, numbers, and the hyphen (-), period (.) pound (#), space ( ), at (@), equals (=), colon (:), and underscore characters. Cannot be changed after the URL transformation profile is added. The following requirement applies only to the Citrix ADC CLI: If the name includes one or more spaces, enclose the name in double or single quotation marks (for example, ?my transform profile? or ?my transform profile?). Minimum length = 1
type	<String>	Read-write	Type of transformation. Always URL for URL

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
			Transformation profiles. Possible values = URL
onlytransformabsurlinbody	<String>	Read-write	In the HTTP body, transform only absolute URLs. Relative URLs are ignored. Possible values = ON, OFF
comment	<String>	Read-write	Any comments to preserve information about this URL Transformation profile.
regexforfindingurlinjavascript	<String>	Read-only	Patclass having regexes to find the URLs in JavaScript.
regexforfindingurlincss	<String>	Read-only	Patclass having regexes to find the URLs in CSS.
regexforfindingurlinxcomponent	<String>	Read-only	Patclass having regexes to find the URLs in X-Component.
regexforfindingurlinxml	<String>	Read-only	Patclass having regexes to find the URLs in XML.
additionalreqheaderslist	<String>	Read-only	Patclass having a list of additional request header names that should transformed.

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
additionalrespheaderslist	<String>	Read-only	Patclass having a list of additional response header names that should transformed.
__count	<Double>	Read-only	count parameter

Operations

(click to see Properties)

ADD| DELETE| UPDATE| UNSET| GET (ALL)| GET| COUNT

Some options that you can use for each operations:

- **Getting warnings in response:**NITRO allows you to get warnings in an operation by specifying the "warning" query parameter as "yes". For example, to get warnings while connecting to the Citrix ADC appliance, the URL is as follows:

  http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/login?warning=yes

  If any, the warnings are displayed in the response payload with the HTTP code "209 X-NITRO-WARNING".

- **Authenticated access for individual NITRO operations:**NITRO allows you to logon to the Citrix ADC appliance to perform individual operations. You can use this option instead of creating a NITRO session (using the login object) and then using that session to perform all operations,

  To do this, you must specify the username and password in the request header of the NITRO request as follows:

  X-NITRO-USER:<username>

  X-NITRO-PASS:<password>

  Note: In such cases, make sure that the request header DOES not include the following:

  Cookie:NITRO_AUTH_TOKEN=<tokenvalue>

`*Note:`*

Mandatory parameters are marked in redand placeholder content is marked in <green>.

add

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/transformprofile

HTTP Method:POST

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Content-Type:application/json

Request Payload:

```
{"transformprofile":{

<b>"name":<String_value>,

</b>"type":<String_value>

}}
```

Response:

HTTP Status Code on Success: 201 Created HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error

delete

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/transformprofile/name_value<String>

HTTP Method:DELETE

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue>

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error

update

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/transformprofile

HTTP Method:PUT

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Content-Type:application/json

Request Payload:

```
{"transformprofile":{

<b>"name":<String_value>,

</b>"type":<String_value>,

"onlytransformabsurlinbody":<String_value>,

"comment":<String_value>

}}
```

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error

unset

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/transformprofile?**action=unset**

HTTP Method:POST

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Content-Type:application/json

Request Payload:

```
{"transformprofile":{

<b>"name":<String_value>,

</b>"type":true,

"onlytransformabsurlinbody":true,

"comment":true

}}
```

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error

get (all)

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/transformprofile

Query-parameters:

attrs

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/transformprofile?**attrs=property-name1,property-name2**

Use this query parameter to specify the resource details that you want to retrieve.

filter

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/transformprofile?**filter=property-name1:property-val1,property-name2:property-val2**

Use this query-parameter to get the filtered set of transformprofile resources configured on Citrix ADC. Filtering can be done on any of the properties of the resource.

view

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/transformprofile?**view=summary**

Note: By default, the retrieved results are displayed in detail view (?view=detail).

pagination

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/transformprofile?**pagesize=#no;pageno=#no**

Use this query-parameter to get the transformprofile resources in chunks.

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "transformprofile": [ {

"name":<String_value>,

"type":<String_value>,

"regexforfindingurlinjavascript":<String_value>,

"regexforfindingurlincss":<String_value>,

"regexforfindingurlinxcomponent":<String_value>,
```

```
"regexforfindingurlinxml":<String_value>,

"additionalreqheaderslist":<String_value>,

"additionalrespheaderslist":<String_value>,

"onlytransformabsurlinbody":<String_value>,

"comment":<String_value>

}]}
```

get

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/transformprofile/name_value<String>

Query-parameters:

attrs

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/transformprofile/name_value<String>?**attrs=property-name1,property-name2**

Use this query parameter to specify the resource details that you want to retrieve.

view

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/transformprofile/name_value<String>?**view=summary**

Note: By default, the retrieved results are displayed in detail view (?view=detail).

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "transformprofile": [ {

"name":<String_value>,

"type":<String_value>,

"regexforfindingurlinjavascript":<String_value>,

"regexforfindingurlincss":<String_value>,

"regexforfindingurlinxcomponent":<String_value>,

"regexforfindingurlinxml":<String_value>,

"additionalreqheaderslist":<String_value>,

"additionalrespheaderslist":<String_value>,

"onlytransformabsurlinbody":<String_value>,

"comment":<String_value>

}]}
```

count

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/transformprofile?**count=yes**

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "transformprofile": [ { "__count": "#no"} ] }
```

transformprofile_binding

Binding object which returns the resources bound to transformprofile.

Properties

(click to see Operations)

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
name	<String>	Read-write	Name of the profile. Minimum length = 1
transformprofile_transformaction_binding	<transformprofile_transformaction_binding[]>	Read-only	transformaction that can be bound to transformprofile.

Operations

(click to see Properties)

GET| GET (ALL)

Some options that you can use for each operations:

- **Getting warnings in response:**NITRO allows you to get warnings in an operation by specifying the "warning" query parameter as "yes". For example, to get warnings while connecting to the Citrix ADC appliance, the URL is as follows:

  http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/login?warning=yes

If any, the warnings are displayed in the response payload with the HTTP code "209 X-NITRO-WARNING".

- **Authenticated access for individual NITRO operations:**NITRO allows you to logon to the Citrix ADC appliance to perform individual operations. You can use this option instead of creating a NITRO session (using the login object) and then using that session to perform all operations,

  To do this, you must specify the username and password in the request header of the NITRO request as follows:

  X-NITRO-USER:<username>

  X-NITRO-PASS:<password>

  Note: In such cases, make sure that the request header DOES not include the following:

  Cookie:NITRO_AUTH_TOKEN=<tokenvalue>

`*Note:`*

Mandatory parameters are marked in redand placeholder content is marked in <green>.

get

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/transformprofile_binding/name_value<String>

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "transformprofile_binding": [ {

"name":<String_value>,

"transformprofile_transformaction_binding":<transformprofile_transformacti
on_binding[]_value>

}]}
```

get (all)

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/transformprofile_binding

Query-parameters:

bulkbindings

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/transformprofile_binding?**bulkbindings=yes**

NITRO allows you to fetch bindings in bulk.

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "transformprofile_binding": [ {
```

```
"name":<String_value>,

"transformprofile_transformaction_binding":<transformprofile_transformaction_binding[]_value>

}]}
```

transformprofile_transformaction_binding

Binding object showing the transformaction that can be bound to transformprofile.

Properties

(click to see Operations)

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
name	<String>	Read-write	Name of the profile. Minimum length = 1
actionname	<String>	Read-write	URL Transformation action name.
priority	<Double>	Read-only	Priority of the Action within the Profile.
resurlfrom	<String>	Read-only	Pattern of original response URLs. It corresponds to the way external users view the server, and acts as a source for response transformations.
cookiedomainfrom	<String>	Read-only	Pattern of the original domain in Set-Cookie headers.
profilename	<String>	Read-only	URL Transformation profile name.

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
state	<String>	Read-only	Enabled flag. Possible values = ENABLED, DISABLED
cookiedomaininto	<String>	Read-only	Pattern of the transformed domain in Set-Cookie headers.
actioncomment	<String>	Read-only	Comments.
resurlinto	<String>	Read-only	Pattern of transformed response URLs. It corresponds to internal addresses and indicates how they are created.
requrlinto	<String>	Read-only	Pattern of transformed request URLs. It corresponds to internal addresses and indicates how they are created.
requrlfrom	<String>	Read-only	Pattern of original request URLs. It corresponds to the way external users view the server, and acts as a source for request transformations.
__count	<Double>	Read-only	count parameter

Operations

(click to see Properties)

GET| GET (ALL)| COUNT

Some options that you can use for each operations:

- **Getting warnings in response:**NITRO allows you to get warnings in an operation by specifying the "warning" query parameter as "yes". For example, to get warnings while connecting to the Citrix ADC appliance, the URL is as follows:

  http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/login?warning=yes

  If any, the warnings are displayed in the response payload with the HTTP code "209 X-NITRO-WARNING".

- **Authenticated access for individual NITRO operations:**NITRO allows you to logon to the Citrix ADC appliance to perform individual operations. You can use this option instead of creating a NITRO session (using the login object) and then using that session to perform all operations,

  To do this, you must specify the username and password in the request header of the NITRO request as follows:

  X-NITRO-USER:<username>

  X-NITRO-PASS:<password>

  Note: In such cases, make sure that the request header DOES not include the following:

  Cookie:NITRO_AUTH_TOKEN=<tokenvalue>

`*Note:`*

Mandatory parameters are marked in redand placeholder content is marked in <green>.

get

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/transformprofile_transformaction_binding/name_value<String>

Query-parameters:

filter

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/transformprofile_transformaction_binding/name_value<String>?**filter=property-name1:property-value1,property-name2:property-value2**

Use this query-parameter to get the filtered set of transformprofile_transformaction_binding resources configured on Citrix ADC. Filtering can be done on any of the properties of the resource.

pagination

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/transformprofile_transformaction_binding/name_value<String>?**pagesize=#no;pageno=#no**

Use this query-parameter to get the transformprofile_transformaction_binding resources in chunks.

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "transformprofile_transformaction_binding": [ {

"name":<String_value>,

"actionname":<String_value>,

"priority":<Double_value>,

"resurlfrom":<String_value>,

"cookiedomainfrom":<String_value>,

"profilename":<String_value>,

"state":<String_value>,
```

```
"cookiedomaininto":<String_value>,

"actioncomment":<String_value>,

"resurlinto":<String_value>,

"requrlinto":<String_value>,

"requrlfrom":<String_value>

}]}
```

get (all)

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/transformprofile_transformaction_binding

Query-parameters:

bulkbindings

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/transformprofile_transformaction_binding?**bulkbindings=yes**

NITRO allows you to fetch bindings in bulk.

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "transformprofile_transformaction_binding": [ {

"name":<String_value>,

"actionname":<String_value>,

"priority":<Double_value>,

"resurlfrom":<String_value>,

"cookiedomainfrom":<String_value>,

"profilename":<String_value>,

"state":<String_value>,

"cookiedomaininto":<String_value>,

"actioncomment":<String_value>,

"resurlinto":<String_value>,

"requrlinto":<String_value>,

"requrlfrom":<String_value>

}]}
```

count

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/transformprofile_transformaction_binding/name_value<String>?**count=yes**

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{"transformprofile_transformaction_binding": [ { "__count": "#no"} ] }
```

Tunnel

Tunnel

SSL VPN Tunnel Configuration.

NAME	DESCRIPTION
tunnelglobal_binding	Binding object showing the resources that can be bound to tunnelglobal
tunnelglobal_tunneltrafficpolicy_binding	Binding object showing the tunneltrafficpolicy that can be bound to tunnelglobal
tunneltrafficpolicy	Configuration for tunnel policy
tunneltrafficpolicy_binding	Binding object showing the resources that can be bound to tunneltrafficpolicy
tunneltrafficpolicy_tunnelglobal_binding	Binding object showing the tunnelglobal that can be bound to tunneltrafficpolicy

tunnelglobal_binding

Binding object which returns the resources bound to tunnelglobal.

Properties

(click to see Operations)

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
tunnelglobal_tunneltrafficpolicy_binding	<tunnelglobal_tunneltrafficpolicy_binding[]>	Read-only	tunneltrafficpolicy that can be bound to tunnelglobal.

Operations

(click to see Properties)

GET| GET (ALL)

Some options that you can use for each operations:

- **Getting warnings in response:**NITRO allows you to get warnings in an operation by specifying the "warning" query parameter as "yes". For example, to get warnings while connecting to the Citrix ADC appliance, the URL is as follows:

  http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/login?warning=yes

  If any, the warnings are displayed in the response payload with the HTTP code "209 X-NITRO-WARNING".

- **Authenticated access for individual NITRO operations:**NITRO allows you to logon to the Citrix ADC appliance to perform individual operations. You can use this option instead of creating a NITRO session (using the login object) and then using that session to perform all operations,

  To do this, you must specify the username and password in the request header of the NITRO request as follows:

  X-NITRO-USER:<username>

  X-NITRO-PASS:<password>

  Note: In such cases, make sure that the request header DOES not include the following:

  Cookie:NITRO_AUTH_TOKEN=<tokenvalue>

`*Note:`*

Mandatory parameters are marked in redand placeholder content is marked in <green>.

get

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/tunnelglobal_binding

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "tunnelglobal_binding": [ {

"tunnelglobal_tunneltrafficpolicy_binding":<tunnelglobal_tunneltrafficpoli
cy_binding[]_value>

}]}
```

get (all)

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/tunnelglobal_binding

Query-parameters:

bulkbindings

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/tunnelglobal_binding?**bulkbindings=yes**

NITRO allows you to fetch bindings in bulk.

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "tunnelglobal_binding": [ {

"tunnelglobal_tunneltrafficpolicy_binding":<tunnelglobal_tunneltrafficpoli
cy_binding[]_value>

}]}
```

tunnelglobal_tunneltrafficpolicy_binding

Binding object showing the tunneltrafficpolicy that can be bound to tunnelglobal.

Properties

(click to see Operations)

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
priority	<Double>	Read-write	Priority.
globalbindtype	<String>	Read-write	. Default value: SYSTEM_GLOBAL Possible values = SYSTEM_GLOBAL, VPN_GLOBAL, RNAT_GLOBAL

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
policyname	<String>	Read-write	Policy name.
gotopriorityexpression	<String>	Read-write	Expression specifying the priority of the next policy which will get evaluated if the current policy rule evaluates to TRUE.
type	<String>	Read-write	Bind point to which the policy is bound. Possible values = REQ_OVERRIDE, REQ_DEFAULT, RES_OVERRIDE, RES_DEFAULT
state	<String>	Read-write	Current state of the binding. If the binding is enabled, the policy is active. Possible values = ENABLED, DISABLED
numpol	<Double>	Read-only	The number of policies bound to the bindpoint.
policytype	<String>	Read-only	Policy type (Classic/ Advanced) to be bound.Used for display. Possible values = Classic Policy, Advanced Policy
__count	<Double>	Read-only	count parameter

Operations

(click to see Properties)

ADD:| DELETE:| GET| GET (ALL)| COUNT

Some options that you can use for each operations:

- **Getting warnings in response:**NITRO allows you to get warnings in an operation by specifying the "warning" query parameter as "yes". For example, to get warnings while connecting to the Citrix ADC appliance, the URL is as follows:

  http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/login?warning=yes

  If any, the warnings are displayed in the response payload with the HTTP code "209 X-NITRO-WARNING".

- **Authenticated access for individual NITRO operations:**NITRO allows you to logon to the Citrix ADC appliance to perform individual operations. You can use this option instead of creating a NITRO session (using the login object) and then using that session to perform all operations,

  To do this, you must specify the username and password in the request header of the NITRO request as follows:

  X-NITRO-USER:<username>

  X-NITRO-PASS:<password>

  Note: In such cases, make sure that the request header DOES not include the following:

  Cookie:NITRO_AUTH_TOKEN=<tokenvalue>

`*Note:`*

Mandatory parameters are marked in redand placeholder content is marked in <green>.

add

URL:http://<Citrix-ADC-IP-address(NSIP)/nitro/v1/config/tunnelglobal_tunneltrafficpolicy_binding

HTTP Method:PUT

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Content-Type:application/json

Request Payload:

```
{

"tunnelglobal_tunneltrafficpolicy_binding":{

<b>"policyname":<String_value>,

</b>"priority":<Double_value>,

"gotopriorityexpression":<String_value>,

"state":<String_value>,

"type":<String_value>

}}
```

Response:

HTTP Status Code on Success: 201 Created HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error

delete

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/tunnelglobal_tunneltrafficpolicy_binding

Query-parameters:

args

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/tunnelglobal_tunneltrafficpolicy_binding?**args=policyname:<String_value>,type:<String_value>,priority:<Double_value>**

HTTP Method:DELETE

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue>

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error

get

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/tunnelglobal_tunneltrafficpolicy_binding

Query-parameters:

filter

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/tunnelglobal_tunneltrafficpolicy_binding?**filter=property-name1:property-value1,property-name2:property-value2**

Use this query-parameter to get the filtered set of tunnelglobal_tunneltrafficpolicy_binding resources configured on Citrix ADC. Filtering can be done on any of the properties of the resource.

pagination

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/tunnelglobal_tunneltrafficpolicy_binding?**pagesize=#no;pageno=#no**

Use this query-parameter to get the tunnelglobal_tunneltrafficpolicy_binding resources in chunks.

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "tunnelglobal_tunneltrafficpolicy_binding": [ {

"priority":<Double_value>,

"globalbindtype":<String_value>,

"policyname":<String_value>,

"gotopriorityexpression":<String_value>,

"type":<String_value>,

"state":<String_value>,

"numpol":<Double_value>,

"policytype":<String_value>

}]}
```

get (all)

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/tunnelglobal_tunneltrafficpolicy_binding

Query-parameters:

bulkbindings

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/tunnelglobal_tunneltrafficpolicy_binding?**bulkbindings=yes**

NITRO allows you to fetch bindings in bulk.

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "tunnelglobal_tunneltrafficpolicy_binding": [ {

"priority":<Double_value>,

"globalbindtype":<String_value>,

"policyname":<String_value>,

"gotopriorityexpression":<String_value>,

"type":<String_value>,

"state":<String_value>,

"numpol":<Double_value>,

"policytype":<String_value>

}]}
```

count

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/tunnelglobal_tunneltrafficpolicy_binding?**count=yes**

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "tunnelglobal_tunneltrafficpolicy_binding": [ { "__count": "#no"} ] }
```

tunneltrafficpolicy

Configuration for tunnel policy resource.

Properties

(click to see Operations)

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
name	<String>	Read-write	Name for the tunnel traffic policy. Must begin with an ASCII alphanumeric or underscore (_) character, and must contain only ASCII alphanumeric, underscore, hash (#), period (.), space, colon (:), at (@), equals (=), and hyphen (-) characters. Cannot be changed after

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
			the policy is created. The following requirement applies only to the Citrix ADC CLI: If the name includes one or more spaces, enclose the name in double or single quotation marks (for example, "my policy" or 'my policy)'. Minimum length = 1
rule	<String>	Read-write	Expression, against which traffic is evaluated. The following requirements apply only to the Citrix ADC CLI: * If the expression includes blank spaces, the entire expression must be enclosed in double quotation marks. * If the expression itself includes double quotation marks, you must

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
			escape the quotations by using the \ character. * Alternatively, you can use single quotation marks to enclose the rule, in which case you do not have to escape the double quotation marks.
action	<String>	Read-write	Name of the built-in compression action to associate with the policy. Minimum length = 1
comment	<String>	Read-write	Any comments to preserve information about this policy.
logaction	<String>	Read-write	Name of the messagelog action to use for requests that match this policy.
newname	<String>	Read-write	New name for the tunnel traffic policy. Must begin with an ASCII alphabetic or underscore (_) character, and must contain only ASCII alphanumeric, underscore, hash (#), period (.), space, colon (:), at (@), e

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
			quals (=), and hyphen (-) characters. Choose a name that reflects the function that the policy performs. The following requirement applies only to the Citrix ADC CLI: If the name includes one or more spaces, enclose the name in double or single quotation marks (for example, "my tunnel policy" or 'my tunnel policy'). Minimum length = 1
expressiontype	<String>	Read-only	Type of policy (Classic/Advanced). Possible values = Classic Policy, Advanced Policy
hits	<Double>	Read-only	No of hits.
undefhits	<Double>	Read-only	Number of policy UNDEF hits.
txbytes	<Double>	Read-only	Number of bytes transmitted.

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
rxbytes	<Double>	Read-only	Number of bytes received.
clientttlb	<Double>	Read-only	Total client TTLB value.
clienttransactions	<Double>	Read-only	Number of client transactions.
serverttlb	<Double>	Read-only	Total server TTLB value.
servertransactions	<Double>	Read-only	Number of server transactions.
isdefault	<Boolean>	Read-only	A value of true is returned if it is a default tunnelpolicy.
__count	<Double>	Read-only	count parameter

Operations

(click to see Properties)

ADD| DELETE| UPDATE| UNSET| RENAME| GET (ALL)| GET| COUNT

Some options that you can use for each operations:

- **Getting warnings in response:**NITRO allows you to get warnings in an operation by specifying the "warning" query parameter as "yes". For example, to get warnings while connecting to the Citrix ADC appliance, the URL is as follows:

  http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/login?warning=yes

  If any, the warnings are displayed in the response payload with the HTTP code "209 X-NITRO-WARNING".

- **Authenticated access for individual NITRO operations:**NITRO allows you to logon to the Citrix ADC appliance to perform individual operations. You can use this option instead of

creating a NITRO session (using the login object) and then using that session to perform all operations,

To do this, you must specify the username and password in the request header of the NITRO request as follows:

X-NITRO-USER:<username>

X-NITRO-PASS:<password>

Note: In such cases, make sure that the request header DOES not include the following:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue>

`*Note:`*

Mandatory parameters are marked in redand placeholder content is marked in <green>.

add

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/tunneltrafficpolicy

HTTP Method:POST

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Content-Type:application/json

Request Payload:

```
{"tunneltrafficpolicy":{

<b>"name":<String_value>,

</b><b>"rule":<String_value>,

</b><b>"action":<String_value>,

</b>"comment":<String_value>,

"logaction":<String_value>

}}
```

Response:

HTTP Status Code on Success: 201 Created HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error

delete

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/tunneltrafficpolicy/name_value<String>

HTTP Method:DELETE

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue>

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error

update

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/tunneltrafficpolicy

HTTP Method:PUT

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Content-Type:application/json

Request Payload:

```
{"tunneltrafficpolicy":{

<b>"name":<String_value>,

</b>"rule":<String_value>,

"action":<String_value>,
```

```
"comment":<String_value>,

"logaction":<String_value>

}}
```

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error

unset

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/tunneltrafficpolicy?**action=unset**

HTTP Method:POST

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Content-Type:application/json

Request Payload:

```
{"tunneltrafficpolicy":{

<b>"name":<String_value>,

</b>"comment":true,

"logaction":true

}}
```

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error

rename

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/tunneltrafficpolicy?**action=rename**

HTTP Method:POST

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Content-Type:application/json

Request Payload:

```
{"tunneltrafficpolicy":{

<b>"name":<String_value>,

</b><b>"newname":<String_value>

</b>}}
```

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error

get (all)

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/tunneltrafficpolicy

Query-parameters:

attrs

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/tunneltrafficpolicy?**attrs=property-name1,property-name2**

Use this query parameter to specify the resource details that you want to retrieve.

filter

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/tunneltrafficpolicy?**filter=property-name1:property-val1,property-name2:property-val2**

Use this query-parameter to get the filtered set of tunneltrafficpolicy resources configured on Citrix ADC. Filtering can be done on any of the properties of the resource.

view

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/tunneltrafficpolicy?**view=summary**

Note: By default, the retrieved results are displayed in detail view (?view=detail).

pagination

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/tunneltrafficpolicy?**pagesize=#no;pageno=#no**

Use this query-parameter to get the tunneltrafficpolicy resources in chunks.

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "tunneltrafficpolicy": [ {

"name":<String_value>,

"expressiontype":<String_value>,

"rule":<String_value>,
```

```
"action":<String_value>,

"hits":<Double_value>,

"undefhits":<Double_value>,

"txbytes":<Double_value>,

"rxbytes":<Double_value>,

"clientttlb":<Double_value>,

"clienttransactions":<Double_value>,

"serverttlb":<Double_value>,

"servertransactions":<Double_value>,

"comment":<String_value>,

"logaction":<String_value>,

"isdefault":<Boolean_value>

}]}
```

get

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/tunneltrafficpolicy/name_value<String>

Query-parameters:

attrs

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/tunneltrafficpolicy/name_value<String>?**attrs=property-name1,property-name2**

Use this query parameter to specify the resource details that you want to retrieve.

view

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/tunneltrafficpolicy/name_value<String>?**view=summary**

Note: By default, the retrieved results are displayed in detail view (?view=detail).

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "tunneltrafficpolicy": [ {

"name":<String_value>,

"expressiontype":<String_value>,

"rule":<String_value>,

"action":<String_value>,

"hits":<Double_value>,

"undefhits":<Double_value>,

"txbytes":<Double_value>,

"rxbytes":<Double_value>,

"clientttlb":<Double_value>,

"clienttransactions":<Double_value>,
```

```
"serverttlb":<Double_value>,

"servertransactions":<Double_value>,

"comment":<String_value>,

"logaction":<String_value>,

"isdefault":<Boolean_value>

}]}
```

count

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/tunneltrafficpolicy?**count=yes**

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "tunneltrafficpolicy": [ { "__count": "#no"} ] }
```

tunneltrafficpolicy_binding

Binding object which returns the resources bound to tunneltrafficpolicy.

Properties

(click to see Operations)

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
name	<String>	Read-write	Name of the tunnel traffic policy for which to show detailed information. Minimum length = 1
tunneltrafficpolicy_tunnelglobal_binding	<tunneltrafficpolicy_tunnelglobal_binding[]>	Read-only	tunnelglobal that can be bound to tunneltrafficpolicy.

Operations

(click to see Properties)

GET| GET (ALL)

Some options that you can use for each operations:

- **Getting warnings in response:**NITRO allows you to get warnings in an operation by specifying the "warning" query parameter as "yes". For example, to get warnings while connecting to the Citrix ADC appliance, the URL is as follows:

  http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/login?warning=yes

  If any, the warnings are displayed in the response payload with the HTTP code "209 X-NITRO-WARNING".

- **Authenticated access for individual NITRO operations:**NITRO allows you to logon to the Citrix ADC appliance to perform individual operations. You can use this option instead of creating a NITRO session (using the login object) and then using that session to perform all operations,

  To do this, you must specify the username and password in the request header of the NITRO request as follows:

  X-NITRO-USER:<username>

  X-NITRO-PASS:<password>

Note: In such cases, make sure that the request header DOES not include the following:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue>

`*Note:`*

Mandatory parameters are marked in redand placeholder content is marked in <green>.

get

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/tunneltrafficpolicy_binding/name_value<String>

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "tunneltrafficpolicy_binding": [ {

"name":<String_value>,

"tunneltrafficpolicy_tunnelglobal_binding":<tunneltrafficpolicy_tunnelglob
al_binding[]_value>

}]}
```

get (all)

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/tunneltrafficpolicy_binding

Query-parameters:

bulkbindings

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/tunneltrafficpolicy_binding?**bulkbindings=yes**

NITRO allows you to fetch bindings in bulk.

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "tunneltrafficpolicy_binding": [ {

"name":<String_value>,

"tunneltrafficpolicy_tunnelglobal_binding":<tunneltrafficpolicy_tunnelglob
al_binding[]_value>

}]}
```

tunneltrafficpolicy_tunnelglobal_binding

Binding object showing the tunnelglobal that can be bound to tunneltrafficpolicy.

Properties

(click to see Operations)

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
boundto	<String>	Read-write	The entity name to which policy is bound.
name	<String>	Read-write	Name of the tunnel traffic policy for which to show detailed information. Minimum length = 1
priority	<Double>	Read-only	.
gotopriorityexpression	<String>	Read-only	Expression specifying the priority of the next policy which will get evaluated if the current policy rule evaluates to TRUE.
activepolicy	<Double>	Read-only	.
__count	<Double>	Read-only	count parameter

Operations

(click to see Properties)

GET| GET (ALL)| COUNT

Some options that you can use for each operations:

- **Getting warnings in response:**NITRO allows you to get warnings in an operation by specifying the "warning" query parameter as "yes". For example, to get warnings while connecting to the Citrix ADC appliance, the URL is as follows:

  http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/login?warning=yes

  If any, the warnings are displayed in the response payload with the HTTP code "209 X-NITRO-WARNING".

- **Authenticated access for individual NITRO operations:**NITRO allows you to logon to the Citrix ADC appliance to perform individual operations. You can use this option instead of creating a NITRO session (using the login object) and then using that session to perform all operations,

  To do this, you must specify the username and password in the request header of the NITRO request as follows:

  X-NITRO-USER:<username>

  X-NITRO-PASS:<password>

  Note: In such cases, make sure that the request header DOES not include the following:

  Cookie:NITRO_AUTH_TOKEN=<tokenvalue>

`*Note:`*

Mandatory parameters are marked in redand placeholder content is marked in <green>.

get

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/tunneltrafficpolicy_tunnelglobal_binding/name_value<String>

Query-parameters:

filter

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/tunneltrafficpolicy_tunnelglobal_binding/name_value<String>?**filter=property-name1:property-value1,property-name2:property-value2**

Use this query-parameter to get the filtered set of tunneltrafficpolicy_tunnelglobal_binding resources configured on Citrix ADC. Filtering can be done on any of the properties of the resource.

pagination

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/tunneltrafficpolicy_tunnelglobal_binding/name_value<String>?**pagesize=#no;pageno=#no**

Use this query-parameter to get the tunneltrafficpolicy_tunnelglobal_binding resources in chunks.

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "tunneltrafficpolicy_tunnelglobal_binding": [ {

"boundto":<String_value>,

"name":<String_value>,

"priority":<Double_value>,

"gotopriorityexpression":<String_value>,

"activepolicy":<Double_value>

}]}
```

get (all)

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/tunneltrafficpolicy_tunnelglobal_binding

Query-parameters:

bulkbindings

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/tunneltrafficpolicy_tunnelglobal_binding?**bulkbindings=yes**

NITRO allows you to fetch bindings in bulk.

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "tunneltrafficpolicy_tunnelglobal_binding": [ {

"boundto":<String_value>,

"name":<String_value>,

"priority":<Double_value>,

"gotopriorityexpression":<String_value>,

"activepolicy":<Double_value>

}]}
```

count

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/tunneltrafficpolicy_tunnelglobal_binding/name_value<String>?**count=yes**

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{"tunneltrafficpolicy_tunnelglobal_binding": [ { "__count": "#no"} ] }
```

ULFD

Ulfd

ULFD server configuration.

NAME	DESCRIPTION
ulfdserver	Configuration for ulfd server

ulfdserver

Configuration for ulfd server resource.

Properties

(click to see Operations)

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
loggerip	<String>	Read-write	IP address of the server where ulfd service is running. Minimum length = 1
state	<String>	Read-only	ULFD server state. Possible values = UP, DOWN, UNKNOWN, BUSY,

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
			OUT OF SERVICE, GOING OUT OF SERVICE, DOWN WHEN GOING OUT OF SERVICE, NS_EMPTY_STR, Unknown, DISABLED
__count	<Double>	Read-only	count parameter

Operations

(click to see Properties)

ADD| DELETE| GET (ALL)| GET| COUNT

Some options that you can use for each operations:

- **Getting warnings in response:**NITRO allows you to get warnings in an operation by specifying the "warning" query parameter as "yes". For example, to get warnings while connecting to the Citrix ADC appliance, the URL is as follows:

  http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/login?warning=yes

  If any, the warnings are displayed in the response payload with the HTTP code "209 X-NITRO-WARNING".

- **Authenticated access for individual NITRO operations:**NITRO allows you to logon to the Citrix ADC appliance to perform individual operations. You can use this option instead of creating a NITRO session (using the login object) and then using that session to perform all operations,

  To do this, you must specify the username and password in the request header of the NITRO request as follows:

  X-NITRO-USER:<username>

  X-NITRO-PASS:<password>

  Note: In such cases, make sure that the request header DOES not include the following:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue>

`*Note:` *

Mandatory parameters are marked in redand placeholder content is marked in <green>.

add

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/ulfdserver

HTTP Method:POST

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Content-Type:application/json

Request Payload:

```
{"ulfdserver":{

<b>"loggerip":<String_value>

</b>}}
```

Response:

HTTP Status Code on Success: 201 Created HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error

delete

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/ulfdserver/loggerip_value<String>

HTTP Method:DELETE

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue>

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error

get (all)

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/ulfdserver

Query-parameters:

attrs

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/ulfdserver?**attrs=property-name1,property-name2**

Use this query parameter to specify the resource details that you want to retrieve.

filter

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/ulfdserver?**filter=property-name1:property-val1,property-name2:property-val2**

Use this query-parameter to get the filtered set of ulfdserver resources configured on Citrix ADC. Filtering can be done on any of the properties of the resource.

view

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/ulfdserver?**view=summary**

Note: By default, the retrieved results are displayed in detail view (?view=detail).

pagination

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/ulfdserver?**pagesize=#no;pageno=#no**

Use this query-parameter to get the ulfdserver resources in chunks.

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "ulfdserver": [ {

"loggerip":<String_value>,

"state":<String_value>

}]}
```

get

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/ulfdserver/loggerip_value<String>

Query-parameters:

attrs

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/ulfdserver/loggerip_value<String>?**attrs=property-name1,property-name2**

Use this query parameter to specify the resource details that you want to retrieve.

view

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/ulfdserver/loggerip_value<String>?**view=summary**

Note: By default, the retrieved results are displayed in detail view (?view=detail).

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "ulfdserver": [ {

"loggerip":<String_value>,

"state":<String_value>

}]}
```

count

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/ulfdserver?**count=yes**

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "ulfdserver": [ { "__count": "#no"} ] }
```

URL Filtering

Urlfiltering

URL Filtering feature is used control access to webpages based on category.

NAME	DESCRIPTION
urlfilteringcategories	Configuration for Categories
urlfilteringcategorization	Configuration for Categorization
urlfilteringcategorygroups	Configuration for Category Groups
urlfilteringparameter	Configuration for URLFILTERING paramter

urlfilteringcategories

Configuration for Categories resource.

Properties

(click to see Operations)

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
group	<String>	Read-write	URL Filtering SDK Category per Group. Minimum length = 1
categories	<String>	Read-only	URL Filtering SDK Categories. Minimum length = 1

Operations

(click to see Properties)

GET (ALL)

Some options that you can use for each operations:

- **Getting warnings in response:**NITRO allows you to get warnings in an operation by specifying the "warning" query parameter as "yes". For example, to get warnings while connecting to the Citrix ADC appliance, the URL is as follows:

  http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/login?warning=yes

  If any, the warnings are displayed in the response payload with the HTTP code "209 X-NITRO-WARNING".

- **Authenticated access for individual NITRO operations:**NITRO allows you to logon to the Citrix ADC appliance to perform individual operations. You can use this option instead of creating a NITRO session (using the login object) and then using that session to perform all operations,

  To do this, you must specify the username and password in the request header of the NITRO request as follows:

  X-NITRO-USER:<username>

  X-NITRO-PASS:<password>

  Note: In such cases, make sure that the request header DOES not include the following:

  Cookie:NITRO_AUTH_TOKEN=<tokenvalue>

`*Note:`*

Mandatory parameters are marked in redand placeholder content is marked in <green>.

get (all)

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/urlfilteringcategories

Query-parameters:

args

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/urlfilteringcategories?
args=group:<String_value>

Use this query-parameter to get urlfilteringcategories resources based on additional properties.

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "urlfilteringcategories": [ {

group:<String_value>"categories":<String_value>

}]}
```

urlfilteringcategorygroups

Configuration for Category Groups resource.

Properties

(click to see Operations)

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
categorygroups	<String>	Read-only	URL Filtering SDK Category Groups. Minimum length = 1

Operations

(click to see Properties)

GET (ALL)

Some options that you can use for each operations:

- **Getting warnings in response:**NITRO allows you to get warnings in an operation by specifying the "warning" query parameter as "yes". For example, to get warnings while connecting to the Citrix ADC appliance, the URL is as follows:

  http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/login?warning=yes

  If any, the warnings are displayed in the response payload with the HTTP code "209 X-NITRO-WARNING".

- **Authenticated access for individual NITRO operations:**NITRO allows you to logon to the Citrix ADC appliance to perform individual operations. You can use this option instead of creating a NITRO session (using the login object) and then using that session to perform all operations,

  To do this, you must specify the username and password in the request header of the NITRO request as follows:

  X-NITRO-USER:<username>

  X-NITRO-PASS:<password>

  Note: In such cases, make sure that the request header DOES not include the following:

  Cookie:NITRO_AUTH_TOKEN=<tokenvalue>

`*Note:`*

Mandatory parameters are marked in redand placeholder content is marked in <green>.

get (all)

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/urlfilteringcategorygroups

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "urlfilteringcategorygroups": [ {

"categorygroups":<String_value>

}]}
```

urlfilteringparameter

Configuration for URLFILTERING paramter resource.

Properties

(click to see Operations)

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
hoursbetweendbupdates	<Double>	Read-write	URL Filtering hours between DB updates. Minimum value = 0 Maximum value = 720
timeofdaytoupdatedb	<String>	Read-write	URL Filtering time of day to update DB.
localdatabasethreads	<Double>	Read-write	URL Filtering Local DB number of threads. Minimum value = 1 Maximum value = 4
cloudhost	<String>	Read-write	URL Filtering Cloud host.

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
seeddbpath	<String>	Read-write	URL Filtering Seed DB path.
maxnumberofcloudthreads	<Double>	Read-only	URL Filtering hours between DB updates. Minimum value = 1 Maximum value = 128
cloudkeepalivetimeout	<Double>	Read-only	URL Filtering Cloud keep alive timeout in msec. Minimum value = 1000 Maximum value = 600000
cloudserverconnecttimeout	<Double>	Read-only	URL Filtering Cloud server connect timeout in msec. Minimum value = 1000 Maximum value = 600000
clouddblookuptimeout	<Double>	Read-only	URL Filtering CloudDB send/receive timeout in msec. Minimum value = 1000 Maximum value = 600000

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
proxyhostip	<String>	Read-only	URL Filtering Cloud Proxy HostIp. Minimum length = 1
proxyport	<Double>	Read-only	URL Filtering Cloud Proxy Port.
proxyusername	<String>	Read-only	URL Filtering Cloud Proxy Username. Minimum length = 1
proxypassword	<String>	Read-only	URL Filtering Cloud Proxy Password. Minimum length = 1
seeddbsizelevel	<Double>	Read-only	URL Filtering Seed DB Size Level to get downloaded. Minimum value = 1 Maximum value = 5

Operations

(click to see Properties)

UPDATE| UNSET| GET (ALL)

Some options that you can use for each operations:

- **Getting warnings in response:**NITRO allows you to get warnings in an operation by specifying the "warning" query parameter as "yes". For example, to get warnings while connecting to the Citrix ADC appliance, the URL is as follows:

  http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/login?warning=yes

  If any, the warnings are displayed in the response payload with the HTTP code "209 X-NITRO-WARNING".

- **Authenticated access for individual NITRO operations:**NITRO allows you to logon to the Citrix ADC appliance to perform individual operations. You can use this option instead of

creating a NITRO session (using the login object) and then using that session to perform all operations,

To do this, you must specify the username and password in the request header of the NITRO request as follows:

X-NITRO-USER:<username>

X-NITRO-PASS:<password>

Note: In such cases, make sure that the request header DOES not include the following:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue>

`*Note:`*

Mandatory parameters are marked in redand placeholder content is marked in <green>.

update

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/urlfilteringparameter

HTTP Method:PUT

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Content-Type:application/json

Request Payload:

```
{"urlfilteringparameter":{

"hoursbetweendbupdates":<Double_value>,

"timeofdaytoupdatedb":<String_value>,

"localdatabasethreads":<Double_value>,

"cloudhost":<String_value>,

"seeddbpath":<String_value>

}}
```

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error

unset

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/urlfilteringparameter?**action=unset**

HTTP Method:POST

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Content-Type:application/json

Request Payload:

```
{"urlfilteringparameter":{

"hoursbetweendbupdates":true,

"timeofdaytoupdatedb":true,

"localdatabasethreads":true,

"cloudhost":true,

"seeddbpath":true

}}
```

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error

get (all)

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/urlfilteringparameter

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "urlfilteringparameter": [ {

"hoursbetweendbupdates":<Double_value>,

"timeofdaytoupdatedb":<String_value>,

"maxnumberofcloudthreads":<Double_value>,

"cloudkeepalivetimeout":<Double_value>,

"cloudserverconnecttimeout":<Double_value>,

"clouddblookuptimeout":<Double_value>,

"proxyhostip":<String_value>,

"proxyport":<Double_value>,

"proxyusername":<String_value>,

"proxypassword":<String_value>,

"seeddbsizelevel":<Double_value>,
```

```
"localdatabasethreads":<Double_value>,

"cloudhost":<String_value>,

"seeddbpath":<String_value>

}]}
```

User

User

User protocol configuration.

NAME	DESCRIPTION
userprotocol	Configuration for user protocol
uservserver	Configuration for virtual server

userprotocol

Configuration for user protocol resource.

Properties

(click to see Operations)

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
name	<String>	Read-write	Unique name for the user protocol. Not case sensitive. Must begin with an ASCII letter or underscore (_) character, and must consist only of ASCII alphanumeric or underscore characters.

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
			Minimum length = 1
transport	<String>	Read-write	Transport layer's protocol. Possible values = TCP, SSL
extension	<String>	Read-write	Name of the extension to add parsing and runtime handling of the protocol packets.
comment	<String>	Read-write	Any comments associated with the protocol.
__count	<Double>	Read-only	count parameter

Operations

(click to see Properties)

ADD| DELETE| UPDATE| UNSET| GET (ALL)| GET| COUNT

Some options that you can use for each operations:

- **Getting warnings in response:**NITRO allows you to get warnings in an operation by specifying the "warning" query parameter as "yes". For example, to get warnings while connecting to the Citrix ADC appliance, the URL is as follows:

  http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/login?warning=yes

  If any, the warnings are displayed in the response payload with the HTTP code "209 X-NITRO-WARNING".
- **Authenticated access for individual NITRO operations:**NITRO allows you to logon to the Citrix ADC appliance to perform individual operations. You can use this option instead of creating a NITRO session (using the login object) and then using that session to perform all operations,

To do this, you must specify the username and password in the request header of the NITRO request as follows:

X-NITRO-USER:<username>

X-NITRO-PASS:<password>

Note: In such cases, make sure that the request header DOES not include the following:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue>

`*Note:`*

Mandatory parameters are marked in redand placeholder content is marked in <green>.

add

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/userprotocol

HTTP Method:POST

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Content-Type:application/json

Request Payload:

```
{"userprotocol":{

<b>"name":<String_value>,

</b><b>"transport":<String_value>,

</b><b>"extension":<String_value>,

</b>"comment":<String_value>

}}
```

Response:

HTTP Status Code on Success: 201 Created HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error

delete

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/userprotocol/name_value<String>

HTTP Method:DELETE

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue>

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error

update

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/userprotocol

HTTP Method:PUT

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Content-Type:application/json

Request Payload:

```
{"userprotocol":{

<b>"name":<String_value>,

</b>"comment":<String_value>

}}
```

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error

unset

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/userprotocol?**action=unset**

HTTP Method:POST

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Content-Type:application/json

Request Payload:

```
{"userprotocol":{

<b>"name":<String_value>,

</b>"comment":true

}}
```

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error

get (all)

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/userprotocol

Query-parameters:

attrs

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/userprotocol?**attrs=property-name1,property-name2**

Use this query parameter to specify the resource details that you want to retrieve.

filter

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/userprotocol?**filter=property-name1:property-val1,property-name2:property-val2**

Use this query-parameter to get the filtered set of userprotocol resources configured on Citrix ADC. Filtering can be done on any of the properties of the resource.

view

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/userprotocol?**view=summary**

Note: By default, the retrieved results are displayed in detail view (?view=detail).

pagination

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/userprotocol?**pagesize=#no;pageno=#no**

Use this query-parameter to get the userprotocol resources in chunks.

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "userprotocol": [ {

"name":<String_value>,

"transport":<String_value>,

"extension":<String_value>,
```

```
"comment":<String_value>

}]}
```

get

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/userprotocol/name_value<String>

Query-parameters:

attrs

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/userprotocol/name_value<String>?**attrs=property-name1,property-name2**

Use this query parameter to specify the resource details that you want to retrieve.

view

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/userprotocol/name_value<String>?**view=summary**

Note: By default, the retrieved results are displayed in detail view (?view=detail).

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "userprotocol": [ {
```

```
"name":<String_value>,

"transport":<String_value>,

"extension":<String_value>,

"comment":<String_value>

}]}
```

count

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/userprotocol?**count=yes**

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "userprotocol": [ { "__count": "#no"} ] }
```

uservserver

Configuration for virtual server resource.

Properties

(click to see Operations)

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
name	<String>	Read-write	Name for the virtual server. Must begin with an ASCII alphanumeric or underscore (_) character, and must contain only ASCII alphanumeric, underscore, hash (#), period (.), space, colon (:), at sign (@), equal sign (=), and hyphen (-) characters. CLI Users: If the name includes one or more spaces, enclose the name in double or single quotation marks (for example, "my vserver" or 'my vserver'). . Minimum length = 1
userprotocol	<String>	Read-write	User protocol uesd by the service.
ipaddress	<String>	Read-write	IPv4 or IPv6 address to assign to the virtual server.
port	<Integer>	Read-write	Port number for the virtual server. Range 1 - 65535 * in CLI is represented as

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
			65535 in NITRO API
defaultlb	<String>	Read-write	Name of the default Load Balancing virtual server used for load balancing of services. The protocol type of default Load Balancing virtual server should be a user type.
Params	<String>	Read-write	Any comments associated with the protocol.
comment	<String>	Read-write	Any comments that you might want to associate with the virtual server.
state	<String>	Read-write	Initial state of the user vserver. Default value: ENABLED Possible values = ENABLED, DISABLED
curstate	<String>	Read-only	Current user vserver state. Possible values = UP, DOWN, UNKNOWN, BUSY, OUT OF SERVICE, GOING OUT OF SERVICE, DOWN

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
			WHEN GOING OUT OF SERVICE, NS_EMPTY_STR, Unknown, DISABLED
value	<String>	Read-only	SSL status. Possible values = Certkey not bound, SSL feature disabled
statechangetimesec	<String>	Read-only	Time when last state change happened. Seconds part.
statechangetimemsec	<Double>	Read-only	Time at which last state change happened. Milliseconds part.
tickssincelaststatechange	<Double>	Read-only	Time in 10 millisecond ticks since the last state change.
nodefaultbindings	<String>	Read-only	to determine if the configuration will have default ssl CIPHER and ECC curve bindings. Default value: NO Possible values = YES, NO
__count	<Double>	Read-only	count parameter

Operations

(click to see Properties)

ADD| DELETE| UPDATE| UNSET| ENABLE| DISABLE| GET (ALL)| GET| COUNT

Some options that you can use for each operations:

- **Getting warnings in response:**NITRO allows you to get warnings in an operation by specifying the "warning" query parameter as "yes". For example, to get warnings while connecting to the Citrix ADC appliance, the URL is as follows:

  http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/login?warning=yes

  If any, the warnings are displayed in the response payload with the HTTP code "209 X-NITRO-WARNING".

- **Authenticated access for individual NITRO operations:**NITRO allows you to logon to the Citrix ADC appliance to perform individual operations. You can use this option instead of creating a NITRO session (using the login object) and then using that session to perform all operations,

  To do this, you must specify the username and password in the request header of the NITRO request as follows:

  X-NITRO-USER:<username>

  X-NITRO-PASS:<password>

  Note: In such cases, make sure that the request header DOES not include the following:

  Cookie:NITRO_AUTH_TOKEN=<tokenvalue>

`*Note:`*

Mandatory parameters are marked in redand placeholder content is marked in <green>.

add

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/uservserver

HTTP Method:POST

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Content-Type:application/json

Request Payload:

```
{"uservserver":{

<b>"name":<String_value>,

</b><b>"userprotocol":<String_value>,

</b><b>"ipaddress":<String_value>,

</b><b>"port":<Integer_value>,

</b><b>"defaultlb":<String_value>,

</b>"Params":<String_value>,

"comment":<String_value>,

"state":<String_value>

}}
```

Response:

HTTP Status Code on Success: 201 Created HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error

delete

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/uservserver/name_value<String>

HTTP Method:DELETE

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue>

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error

update

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/uservserver

HTTP Method:PUT

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Content-Type:application/json

Request Payload:

```
{"uservserver":{

<b>"name":<String_value>,

</b>"ipaddress":<String_value>,

"defaultlb":<String_value>,

"Params":<String_value>,

"comment":<String_value>

}}
```

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error

unset

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/uservserver?**action=unset**

HTTP Method:POST

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Content-Type:application/json

Request Payload:

```
{"uservserver":{

<b>"name":<String_value>,

</b>"Params":true,

"comment":true

}}
```

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error

enable

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/uservserver?**action=enable**

HTTP Method:POST

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Content-Type:application/json

Request Payload:

```
{"uservserver":{

<b>"name":<String_value>

</b>}}
```

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error

disable

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/uservserver?**action=disable**

HTTP Method:POST

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Content-Type:application/json

Request Payload:

```
{"uservserver":{

<b>"name":<String_value>

</b>}}
```

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error

get (all)

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/uservserver

Query-parameters:

attrs

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/uservserver?**attrs=property-name1,property-name2**

Use this query parameter to specify the resource details that you want to retrieve.

filter

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/uservserver?**filter=property-name1:property-val1,property-name2:property-val2**

Use this query-parameter to get the filtered set of uservserver resources configured on Citrix ADC. Filtering can be done on any of the properties of the resource.

view

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/uservserver?**view=summary**

Note: By default, the retrieved results are displayed in detail view (?view=detail).

pagination

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/uservserver?**pagesize=#no;pageno=#no**

Use this query-parameter to get the uservserver resources in chunks.

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "uservserver": [ {

"name":<String_value>,

"userprotocol":<String_value>,
```

```
"ipaddress":<String_value>,

"port":<Integer_value>,

"defaultlb":<String_value>,

"Params":<String_value>,

"comment":<String_value>,

"curstate":<String_value>,

"value":<String_value>,

"statechangetimesec":<String_value>,

"statechangetimemsec":<Double_value>,

"tickssincelaststatechange":<Double_value>,

"state":<String_value>,

"nodefaultbindings":<String_value>

}]}
```

get

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/uservserver/name_value<String>

Query-parameters:

attrs

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/uservserver/name_value<String>?**attrs=property-name1,property-name2**

Use this query parameter to specify the resource details that you want to retrieve.

view

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/uservserver/name_value<String>?**view=summary**

Note: By default, the retrieved results are displayed in detail view (?view=detail).

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "uservserver": [ {

"name":<String_value>,

"userprotocol":<String_value>,

"ipaddress":<String_value>,

"port":<Integer_value>,

"defaultlb":<String_value>,

"Params":<String_value>,

"comment":<String_value>,

"curstate":<String_value>,

"value":<String_value>,

"statechangetimesec":<String_value>,
```

```
"statechangetimemsec":<Double_value>,

"tickssincelaststatechange":<Double_value>,

"state":<String_value>,

"nodefaultbindings":<String_value>

}]}
```

count

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/uservserver?**count=yes**

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "uservserver": [ { "__count": "#no"} ] }
```

Utility

Utility

Utilities.

NAME	DESCRIPTION
callhome	Configuration for callhome
install	
ping	
ping6	
raid	Configuration for raid support
techsupport	Configuration for tech support
traceroute	
traceroute6	

callhome

Configuration for callhome resource.

Properties

(click to see Operations)

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
nodeid	<Double>	Read-write	Unique number that identifies the cluster node. Minimum value = 0 Maximum value = 31
mode	<String>	Read-write	CallHome mode of operation. Default value: Default Possible values = Default, CSP

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
emailaddress	<String>	Read-write	Email address of the contact administrator. Maximum length = 78
hbcustominterval	<Double>	Read-write	Interval (in days) between CallHome heartbeats. Default value: 7 Minimum value = 1 Maximum value = 30
proxymode	<String>	Read-write	Enables or disables the proxy mode. The proxy server can be set by either specifying the IP address of the server or the name of the service representing the proxy server. Default value: NO Possible values = YES, NO
ipaddress	<String>	Read-write	IP address of the proxy server. Minimum length = 1
proxyauthservice	<String>	Read-write	Name of the service that represents the proxy server. Maximum length = 128

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
port	<Integer>	Read-write	HTTP port on the Proxy server. This is a mandatory parameter for both IP address and service name based configuration. Minimum value = 1 Range 1 - 65535 * in CLI is represented as 65535 in NITRO API
sslcardfirstfailure	<String>	Read-only	First occurrence SSL card failure.
sslcardlatestfailure	<String>	Read-only	Latest occurrence SSL card failure.
powfirstfail	<String>	Read-only	First occurrence power supply unit failure.
powlatestfailure	<String>	Read-only	Latest occurrence power supply unit failure.
hddfirstfail	<String>	Read-only	First occurrence hard disk drive failure.
hddlatestfailure	<String>	Read-only	Latest occurrence hard disk drive failure.
flashfirstfail	<String>	Read-only	First occurrence compact flash failure.

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
flashlatestfailure	<String>	Read-only	Latest occurrence compact flush failure.
rlfirsthighdrop	<String>	Read-only	First occurence of high rate limit drops.
rllatesthighdrop	<String>	Read-only	Latest occurence of high rate limit drops.
restartlatestfail	<String>	Read-only	Latest occurrence warm restart failure.
memthrefirstanomaly	<String>	Read-only	First occurrence of memory anomaly.
memthrelatestanomaly	<String>	Read-only	Latest occurrence of memory anomaly.
callhomestatus	<String[]>	Read-only	Callhome feature enabled/disable, register with upload server successful/ failed. Possible values = ENABLED, DISABLED, SUCCESSFUL, FAILED, INVALID HOSTNAME, NO DNSSERVER, HOSTNAME RESOLUTION FAILED

Operations

(click to see Properties)

UPDATE| UNSET| GET (ALL)

Some options that you can use for each operations:

- **Getting warnings in response:**NITRO allows you to get warnings in an operation by specifying the "warning" query parameter as "yes". For example, to get warnings while connecting to the Citrix ADC appliance, the URL is as follows:

  http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/login?warning=yes

  If any, the warnings are displayed in the response payload with the HTTP code "209 X-NITRO-WARNING".

- **Authenticated access for individual NITRO operations:**NITRO allows you to logon to the Citrix ADC appliance to perform individual operations. You can use this option instead of creating a NITRO session (using the login object) and then using that session to perform all operations,

  To do this, you must specify the username and password in the request header of the NITRO request as follows:

  X-NITRO-USER:<username>

  X-NITRO-PASS:<password>

  Note: In such cases, make sure that the request header DOES not include the following:

  Cookie:NITRO_AUTH_TOKEN=<tokenvalue>

`*Note:`*

Mandatory parameters are marked in redand placeholder content is marked in <green>.

update

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/callhome

HTTP Method:PUT

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Content-Type:application/json

Request Payload:

```
{"callhome":{

"mode":<String_value>,

"emailaddress":<String_value>,

"hbcustominterval":<Double_value>,

"proxymode":<String_value>,

"ipaddress":<String_value>,

"proxyauthservice":<String_value>,

"port":<Integer_value>

}}
```

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error

unset

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/callhome?**action=unset**

HTTP Method:POST

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Content-Type:application/json

Request Payload:

```
{"callhome":{
```

```
"mode":true,

"emailaddress":true,

"hbcustominterval":true,

"proxymode":true,

"ipaddress":true,

"proxyauthservice":true,

"port":true

}}
```

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error

get (all)

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/callhome

Query-parameters:

args

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/callhome?**args=nodeid:<Double_value>**

Use this query-parameter to get callhome resources based on additional properties.

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "callhome": [ {

nodeid:<Double_value>"mode":<String_value>,

"emailaddress":<String_value>,

"hbcustominterval":<Double_value>,

"proxymode":<String_value>,

"ipaddress":<String_value>,

"port":<Integer_value>,

"proxyauthservice":<String_value>,

"sslcardfirstfailure":<String_value>,

"sslcardlatestfailure":<String_value>,

"powfirstfail":<String_value>,

"powlatestfailure":<String_value>,

"hddfirstfail":<String_value>,

"hddlatestfailure":<String_value>,

"flashfirstfail":<String_value>,

"flashlatestfailure":<String_value>,

"rlfirsthighdrop":<String_value>,
```

```
"rllatesthighdrop":<String_value>,

"restartlatestfail":<String_value>,

"memthrefirstanomaly":<String_value>,

"memthrelatestanomaly":<String_value>,

"callhomestatus":<String[]_value>

}]}
```

install

Configuration for 0 resource.

Properties

(click to see Operations)

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
url	<String>	Read-write	Url for the build file. Must be in the following formats: http://[user]:[password]@host/path/to/file https://[user]:[password]@host/path/to/file sftp://[user]:[password]@host/path/to/file

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
			scp://[user]:[password]@host/path/to/file ftp://[user]:[password]@host/path/to/file file://path/to/file.
y	<Boolean>	Read-write	Do not prompt for yes/no before rebooting.
l	<Boolean>	Read-write	Use this flag to enable callhome.
enhancedupgrade	<Boolean>	Read-write	Use this flag for upgrading from/to enhancement mode.
resizeswapvar	<Boolean>	Read-write	Use this flag to change swap size on ONLY 64bit nCore/MCNS/VMPE systems NON-VPX systems.

Operations

(click to see Properties)

INSTALL

Some options that you can use for each operations:

- **Getting warnings in response:**NITRO allows you to get warnings in an operation by specifying the "warning" query parameter as "yes". For example, to get warnings while connecting to the Citrix ADC appliance, the URL is as follows:

  http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/login?warning=yes

If any, the warnings are displayed in the response payload with the HTTP code "209 X-NITRO-WARNING".

- **Authenticated access for individual NITRO operations:**NITRO allows you to logon to the Citrix ADC appliance to perform individual operations. You can use this option instead of creating a NITRO session (using the login object) and then using that session to perform all operations,

  To do this, you must specify the username and password in the request header of the NITRO request as follows:

  X-NITRO-USER:<username>

  X-NITRO-PASS:<password>

  Note: In such cases, make sure that the request header DOES not include the following:

  Cookie:NITRO_AUTH_TOKEN=<tokenvalue>

`*Note:`*

Mandatory parameters are marked in redand placeholder content is marked in <green>.

Install

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/install

HTTP Method:POST

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Content-Type:application/json

Request Payload:

```
{"install":{

<b>"url":<String_value>,

</b>"y":<Boolean_value>,

"l":<Boolean_value>,
```

```
"enhancedupgrade":<Boolean_value>,

"resizeswapvar":<Boolean_value>

}}
```

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error

ping

Configuration for 0 resource.

Properties

(click to see Operations)

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
c	<Double>	Read-write	Number of packets to send. The default value is infinite. For Nitro API, defalut value is taken as 5. Minimum value = 1 Maximum value = 65535
i	<Double>	Read-write	Waiting time, in seconds. The default value is 1 second. Minimum value = 0 Maximum value = 65535

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
I	<String>	Read-write	Network interface on which to ping, if you have multiple interfaces. Minimum length = 1 Maximum length = 15
n	<Boolean>	Read-write	Numeric output only. No name resolution.
p	<String>	Read-write	Pattern to fill in packets. Can be up to 16 bytes, useful for diagnosing data-dependent problems. Minimum length = 1 Maximum length = 32
q	<Boolean>	Read-write	Quiet output. Only the summary is printed. For Nitro API, this flag is set by default.
s	<Double>	Read-write	Data size, in bytes. The default value is 56. Minimum value = 0 Maximum value = 65507
S	<String>	Read-write	Source IP address to be used in the outgoing query packets. If the IP addrESS does not

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
			belongs to this appliance, an error is returned and nothing is sent. Minimum length = 1
T	<Double>	Read-write	Traffic Domain Id. Minimum value = 1 Maximum value = 4094
t	<Double>	Read-write	Time-out, in seconds, before ping exits. Minimum value = 1 Maximum value = 3600
hostName	<String>	Read-write	Address of host to ping. Minimum length = 1 Maximum length = 256
response	<String>	Read-only	.

Operations

(click to see Properties)

PING

Some options that you can use for each operations:

- **Getting warnings in response:**NITRO allows you to get warnings in an operation by specifying the "warning" query parameter as "yes". For example, to get warnings while connecting to the Citrix ADC appliance, the URL is as follows:

  http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/login?warning=yes

If any, the warnings are displayed in the response payload with the HTTP code "209 X-NITRO-WARNING".

- **Authenticated access for individual NITRO operations:**NITRO allows you to logon to the Citrix ADC appliance to perform individual operations. You can use this option instead of creating a NITRO session (using the login object) and then using that session to perform all operations,

  To do this, you must specify the username and password in the request header of the NITRO request as follows:

  X-NITRO-USER:<username>

  X-NITRO-PASS:<password>

  Note: In such cases, make sure that the request header DOES not include the following:

  Cookie:NITRO_AUTH_TOKEN=<tokenvalue>

`*Note:`*

Mandatory parameters are marked in redand placeholder content is marked in <green>.

Ping

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/ping

HTTP Method:POST

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Content-Type:application/json

Request Payload:

```
{"ping":{

"c":<Double_value>,

"i":<Double_value>,

"I":<String_value>,
```

```
"n":<Boolean_value>,

"p":<String_value>,

"q":<Boolean_value>,

"s":<Double_value>,

"S":<String_value>,

"T":<Double_value>,

"t":<Double_value>,

<b>"hostName":<String_value>

</b>}}
```

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Payload:**

```
{ "ping": [ {

"response":<String_value>



}]}
```

ping6

Configuration for 0 resource.

Properties

(click to see Operations)

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
b	<Double>	Read-write	Set socket buffer size. If used, should be used with roughly +100 then the datalen (-s option). The default value is 8192. Minimum value = 132 Maximum value = 131071
c	<Double>	Read-write	Number of packets to send. The default value is infinite. For Nitro API, defalut value is taken as 5. Minimum value = 1 Maximum value = 65535
i	<Double>	Read-write	Waiting time, in seconds. The default value is 1 second. Minimum value = 0 Maximum value = 65535
I	<String>	Read-write	Network interface on which to ping, if you have multiple interfaces. Minimum length = 1

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
			Maximum length = 15
m	<Boolean>	Read-write	By default, ping6 asks the kernel to fragment packets to fit into the minimum IPv6 MTU.The -m option will suppress the behavior for unicast packets.
n	<Boolean>	Read-write	Numeric output only. No name resolution.
p	<String>	Read-write	Pattern to fill in packets. Can be up to 16 bytes, useful for diagnosing data-dependent problems. Minimum length = 1 Maximum length = 32
q	<Boolean>	Read-write	Quiet output. Only summary is printed. For Nitro API, this flag is set by default.
s	<Double>	Read-write	Data size, in bytes. The default value is 32. Minimum value = 0 Maximum value = 65527
V	<Double>	Read-write	VLAN ID for link local address.

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
			Minimum value = 1 Maximum value = 4094
S	<String>	Read-write	Source IP address to be used in the outgoing query packets. Minimum length = 1
T	<Double>	Read-write	Traffic Domain Id. Minimum value = 1 Maximum value = 4094
t	<Double>	Read-write	Timeout in seconds before ping6 exits.
hostName	<String>	Read-write	Address of host to ping. Minimum length = 1 Maximum length = 256
response	<String>	Read-only	.

Operations

(click to see Properties)

PING6

Some options that you can use for each operations:

- **Getting warnings in response:**NITRO allows you to get warnings in an operation by specifying the "warning" query parameter as "yes". For example, to get warnings while connecting to the Citrix ADC appliance, the URL is as follows:

  http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/login?warning=yes

If any, the warnings are displayed in the response payload with the HTTP code "209 X-NITRO-WARNING".

- **Authenticated access for individual NITRO operations:**NITRO allows you to logon to the Citrix ADC appliance to perform individual operations. You can use this option instead of creating a NITRO session (using the login object) and then using that session to perform all operations,

  To do this, you must specify the username and password in the request header of the NITRO request as follows:

  X-NITRO-USER:<username>

  X-NITRO-PASS:<password>

  Note: In such cases, make sure that the request header DOES not include the following:

  Cookie:NITRO_AUTH_TOKEN=<tokenvalue>

`*Note:`*

Mandatory parameters are marked in redand placeholder content is marked in <green>.

Ping6

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/ping6

HTTP Method:POST

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Content-Type:application/json

Request Payload:

```
{"ping6":{

"b":<Double_value>,

"c":<Double_value>,

"i":<Double_value>,
```

```
"I":<String_value>,

"m":<Boolean_value>,

"n":<Boolean_value>,

"p":<String_value>,

"q":<Boolean_value>,

"s":<Double_value>,

"V":<Double_value>,

"S":<String_value>,

"T":<Double_value>,

"t":<Double_value>,

<b>"hostName":<String_value>

</b>}}
```

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Payload:**

```
{ "ping6": [ {

"response":<String_value>



}]}
```

raid

Configuration for raid support resource.

Operations

(click to see Properties)

GET (ALL)

Some options that you can use for each operations:

- **Getting warnings in response:**NITRO allows you to get warnings in an operation by specifying the "warning" query parameter as "yes". For example, to get warnings while connecting to the Citrix ADC appliance, the URL is as follows:

  http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/login?warning=yes

  If any, the warnings are displayed in the response payload with the HTTP code "209 X-NITRO-WARNING".

- **Authenticated access for individual NITRO operations:**NITRO allows you to logon to the Citrix ADC appliance to perform individual operations. You can use this option instead of creating a NITRO session (using the login object) and then using that session to perform all operations,

  To do this, you must specify the username and password in the request header of the NITRO request as follows:

  X-NITRO-USER:<username>

  X-NITRO-PASS:<password>

  Note: In such cases, make sure that the request header DOES not include the following:

  Cookie:NITRO_AUTH_TOKEN=<tokenvalue>

`*Note:`*

Mandatory parameters are marked in redand placeholder content is marked in <green>.

get (all)

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/raid

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "raid": [ {


}]}
```

techsupport

Configuration for tech support resource.

Properties

(click to see Operations)

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
scope	<String>	Read-write	Use this option to gather data on the present node, all cluster nodes, or for the specified partitions. The CLUSTER scope generates smaller abbreviated archives

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
			for all nodes. The PARTITION scope collects the admin partition in addition to those specified. The partitionName option is only required for the PARTITION scope. Default value: NODE Possible values = NODE, CLUSTER, PARTITION
partitionname	<String>	Read-write	Name of the partition. Minimum length = 1
upload	<Boolean>	Read-write	Securely upload the collector archive to Citrix Technical Support using SSL. MyCitrix credentials will be required. If used with the -file option, no new collector archive is generated. Instead, the specified archive is uploaded. Note that the upload operation time depends on the size of the archive file, and the connection bandwidth.

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
proxy	<String>	Read-write	Specifies the proxy server to be used when uploading a collector archive. Use this parameter if the Citrix ADC does not have direct internet connectivity. The basic format of the proxy string is: "proxy_IP:<proxy_port>" (without quotes). If the proxy requires authentication the format is: "username:password@proxy_IP:<proxy_port>".
casenumber	<String>	Read-write	Specifies the associated case or service request number if it has already been opened with Citrix Technical Support.
file	<String>	Read-write	Specifies the name (with full path) of the collector archive file to be uploaded. If this is specified, no new collector archive is generated.
description	<String>	Read-write	Provides a text description for the the upload, and can be used for logging purposes.

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
username	<String>	Read-write	Specifies My Citrix user name, which is used to login to Citrix upload server.
password	<String>	Read-write	Specifies My Citrix password, which is used to login to Citrix upload server.
response	<String>	Read-only	Output as text printed to console and syslog at NOTICE level.
servername	<String>	Read-only	.

Operations

(click to see Properties)

GET (ALL)

Some options that you can use for each operations:

- **Getting warnings in response:**NITRO allows you to get warnings in an operation by specifying the "warning" query parameter as "yes". For example, to get warnings while connecting to the Citrix ADC appliance, the URL is as follows:

  http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/login?warning=yes

  If any, the warnings are displayed in the response payload with the HTTP code "209 X-NITRO-WARNING".

- **Authenticated access for individual NITRO operations:**NITRO allows you to logon to the Citrix ADC appliance to perform individual operations. You can use this option instead of creating a NITRO session (using the login object) and then using that session to perform all operations,

  To do this, you must specify the username and password in the request header of the NITRO request as follows:

X-NITRO-USER:<username>

X-NITRO-PASS:<password>

Note: In such cases, make sure that the request header DOES not include the following:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue>

`*Note:`*

Mandatory parameters are marked in redand placeholder content is marked in <green>.

get (all)

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/techsupport

Query-parameters:

args

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/techsupport?
args=scope:<String_value>,partitionname:<String_value>,upload:<Boolean_value>,proxy:<String_valu

Use this query-parameter to get techsupport resources based on additional properties.

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "techsupport": [ {

scope:<String_value>,partitionname:<String_value>,upload:<Boolean_value>,p
```

```
roxy:<String_value>,casenumber:<String_value>,file:<String_value>,descript
ion:<String_value>,username:<String_value>,password:<String_value>"respons
e":<String_value>,

"servername":<String_value>

}]}
```

traceroute

Configuration for 0 resource.

Properties

(click to see Operations)

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
S	<Boolean>	Read-write	Print a summary of how many probes were not answered for each hop.
n	<Boolean>	Read-write	Print hop addresses numerically instead of symbolically and numerically.
r	<Boolean>	Read-write	Bypass normal routing tables and send directly to a host on an attached network. If the host is not on a directly attached network, an error is returned.
v	<Boolean>	Read-write	Verbose output. List received ICMP packets other than TIME_EXCEEDED

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
			and UNREACHABLE.
M	<Double>	Read-write	Minimum TTL value used in outgoing probe packets. Default value: 1 Minimum value = 1 Maximum value = 255
m	<Integer>	Read-write	Maximum TTL value used in outgoing probe packets. For Nitro API, default value is taken as 10. Default value: 64 Minimum value = 1 Maximum value = 255
P	<String>	Read-write	Send packets of specified IP protocol. The currently supported protocols are UDP and ICMP. Minimum length = 1 Maximum length = 20
p	<Integer>	Read-write	Base port number used in probes. Default value: 33434

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
			Minimum value = 1 Maximum value = 65535
q	<Integer>	Read-write	Number of queries per hop. For Nitro API, defalut value is taken as 1. Default value: 3 Minimum value = 1 Maximum value = 65535
s	<String>	Read-write	Source IP address to use in the outgoing query packets. If the IP address does not belong to this appliance, an error is returned and nothing is sent. Minimum length = 1
T	<Double>	Read-write	Traffic Domain Id. Minimum value = 1 Maximum value = 4094
t	<Integer>	Read-write	Type-of-service in query packets. Minimum value = 0 Maximum value = 255
w	<Integer>	Read-write	Time (in seconds) to wait for a response

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
			to a query. For Nitro API, defalut value is set to 3. Default value: 5 Minimum value = 2 Maximum value = 86399
host	<String>	Read-write	Destination host IP address or name. Minimum length = 1 Maximum length = 64
packetlen	<Integer>	Read-write	Length (in bytes) of the query packets. Default value: 44 Minimum value = 44 Maximum value = 32768
response	<String>	Read-only	.

Operations

(click to see Properties)

TRACEROUTE

Some options that you can use for each operations:

- **Getting warnings in response:**NITRO allows you to get warnings in an operation by specifying the "warning" query parameter as "yes". For example, to get warnings while connecting to the Citrix ADC appliance, the URL is as follows:

  http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/login?warning=yes

If any, the warnings are displayed in the response payload with the HTTP code "209 X-NITRO-WARNING".

- **Authenticated access for individual NITRO operations:**NITRO allows you to logon to the Citrix ADC appliance to perform individual operations. You can use this option instead of creating a NITRO session (using the login object) and then using that session to perform all operations,

  To do this, you must specify the username and password in the request header of the NITRO request as follows:

  X-NITRO-USER:<username>

  X-NITRO-PASS:<password>

  Note: In such cases, make sure that the request header DOES not include the following:

  Cookie:NITRO_AUTH_TOKEN=<tokenvalue>

`*Note:`*

Mandatory parameters are marked in redand placeholder content is marked in <green>.

Traceroute

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/traceroute

HTTP Method:POST

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Content-Type:application/json

Request Payload:

```
{"traceroute":{

"S":<Boolean_value>,

"n":<Boolean_value>,

"r":<Boolean_value>,
```

```
"v":<Boolean_value>,

"M":<Double_value>,

"m":<Integer_value>,

"P":<String_value>,

"p":<Integer_value>,

"q":<Integer_value>,

"s":<String_value>,

"T":<Double_value>,

"t":<Integer_value>,

"w":<Integer_value>,

<b>"host":<String_value>,

</b>"packetlen":<Integer_value>

}}
```

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Payload:**

```
{ "traceroute": [ {

"response":<String_value>



}]}
```

traceroute6

Configuration for 0 resource.

Properties

(click to see Operations)

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
n	<Boolean>	Read-write	Print hop addresses numerically rather than symbolically and numerically.
I	<Boolean>	Read-write	Use ICMP ECHO for probes.
r	<Boolean>	Read-write	Bypass normal routing tables and send directly to a host on an attached network. If the host is not on a directly attached network, an error is returned.
v	<Boolean>	Read-write	Verbose output. List received ICMP packets other than TIME_EXCEEDED and UNREACHABLE.
m	<Integer>	Read-write	Maximum hop value for outgoing probe packets. For Nitro API, default value is taken as 10. Default value: 64

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
			Minimum value = 1 Maximum value = 255
p	<Integer>	Read-write	Base port number used in probes. Default value: 33434 Minimum value = 1 Maximum value = 65535
q	<Integer>	Read-write	Number of probes per hop. For Nitro API, default value is taken as 1. Default value: 3 Minimum value = 1 Maximum value = 65535
s	<String>	Read-write	Source IP address to use in the outgoing query packets. If the IP address does not belong to this appliance, an error is returned and nothing is sent. Minimum length = 1
T	<Double>	Read-write	Traffic Domain Id. Minimum value = 1 Maximum value = 4094

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
w	<Integer>	Read-write	Time (in seconds) to wait for a response to a query. For Nitro API, defalut value is set to 3. Default value: 5 Minimum value = 2 Maximum value = 86399
host	<String>	Read-write	Destination host IP address or name. Minimum length = 1 Maximum length = 256
packetlen	<Integer>	Read-write	Length (in bytes) of the query packets. Default value: 44 Minimum value = 44 Maximum value = 32768
response	<String>	Read-only	.

Operations

(click to see Properties)

TRACEROUTE6

Some options that you can use for each operations:

- **Getting warnings in response:**NITRO allows you to get warnings in an operation by specifying the "warning" query parameter as "yes". For example, to get warnings while connecting to the Citrix ADC appliance, the URL is as follows:

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/login?warning=yes

If any, the warnings are displayed in the response payload with the HTTP code "209 X-NITRO-WARNING".

- **Authenticated access for individual NITRO operations:**NITRO allows you to logon to the Citrix ADC appliance to perform individual operations. You can use this option instead of creating a NITRO session (using the login object) and then using that session to perform all operations,

  To do this, you must specify the username and password in the request header of the NITRO request as follows:

  X-NITRO-USER:<username>

  X-NITRO-PASS:<password>

  Note: In such cases, make sure that the request header DOES not include the following:

  Cookie:NITRO_AUTH_TOKEN=<tokenvalue>

`*Note:`*

Mandatory parameters are marked in redand placeholder content is marked in <green>.

Traceroute6

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/traceroute6

HTTP Method:POST

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Content-Type:application/json

Request Payload:

```
{"traceroute6":{

"n":<Boolean_value>,

"I":<Boolean_value>,

"r":<Boolean_value>,
```

```
"v":<Boolean_value>,

"m":<Integer_value>,

"p":<Integer_value>,

"q":<Integer_value>,

"s":<String_value>,

"T":<Double_value>,

"w":<Integer_value>,

<b>"host":<String_value>,

</b>"packetlen":<Integer_value>

}}
```

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Payload:**

```
{ "traceroute6": [ {

"response":<String_value>



}]}
```

Video Optimization

VideoOptimization

Video optimization feature is used to show (i) the stats of different media types that are being served by the Citrix ADC and (ii) the details of optimization applied on ABR videos

NAME	DESCRIPTION
videooptimizationdetectionaction	Configuration for videooptimization detectionaction
videooptimizationdetectionpolicy	Configuration for videooptimization detectionpolicy
videooptimizationdetectionpolicy_binding	Binding object showing the resources that can be bound to videooptimizationdetectionpolicy
videooptimizationdetectionpolicy_lbvserver_binding	Binding object showing the lbvserver that can be bound to videooptimizationdetectionpolicy
videooptimizationdetectionpolicy_videooptimizationglobaldetection_binding	Binding object showing the videooptimizationglobaldetection that can be bound to videooptimizationdetectionpolicy
videooptimizationdetectionpolicylabel	Configuration for videooptimization detection policy label
videooptimizationdetectionpolicylabel_binding	Binding object showing the resources that can be bound to videooptimizationdetectionpolicylabel
videooptimizationdetectionpolicylabel_policybinding_binding	Binding object showing the policybinding that can be bound to videooptimizationdetectionpolicylabel
videooptimizationdetectionpolicylabel_videooptimizationdetectionpolicy_binding	Binding object showing the videooptimizationdetectionpolicy that can be bound to videooptimizationdetectionpolicylabel
videooptimizationglobaldetection_binding	Binding object showing the resources that can be bound to videooptimizationglobaldetection

NAME	DESCRIPTION
videooptimizationglobaldetection_videooptimizationdetectionpolicy_binding	Binding object showing the videooptimizationdetectionpolicy that can be bound to videooptimizationglobaldetection
videooptimizationglobalpacing_binding	Binding object showing the resources that can be bound to videooptimizationglobalpacing
videooptimizationglobalpacing_videooptimizationpacingpolicy_binding	Binding object showing the videooptimizationpacingpolicy that can be bound to videooptimizationglobalpacing
videooptimizationpacingaction	Configuration for videooptimization pacingaction
videooptimizationpacingpolicy	Configuration for videooptimization pacingpolicy
videooptimizationpacingpolicy_binding	Binding object showing the resources that can be bound to videooptimizationpacingpolicy
videooptimizationpacingpolicy_lbvserver_binding	Binding object showing the lbvserver that can be bound to videooptimizationpacingpolicy
videooptimizationpacingpolicy_videooptimizationglobalpacing_binding	Binding object showing the videooptimizationglobalpacing that can be bound to videooptimizationpacingpolicy
videooptimizationpacingpolicylabel	Configuration for videooptimization pacing policy label
videooptimizationpacingpolicylabel_binding	Binding object showing the resources that can be bound to videooptimizationpacingpolicylabel
videooptimizationpacingpolicylabel_policybinding_binding	Binding object showing the policybinding that can be bound to videooptimizationpacingpolicylabel
videooptimizationpacingpolicylabel_videooptimizationpacingpolicy_binding	Binding object showing the videooptimizationpacingpolicy that can be bound to videooptimizationpacingpolicylabel

NAME	DESCRIPTION
videooptimizationparameter	Configuration for VideoOptimization parameter

videooptimizationparameter

Configuration for VideoOptimization parameter resource.

Properties

(click to see Operations)

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
randomsamplingpercentage	<Double>	Read-write	Random Sampling Percentage. Default value: 0 Minimum value = 0 Maximum value = 100
quicpacingrate	<Double>	Read-write	QUIC Video Pacing Rate (Kbps). Minimum value = 0 Maximum value = 2147483647

Operations

(click to see Properties)

UPDATE| UNSET| GET (ALL)

Some options that you can use for each operations:

- **Getting warnings in response:**NITRO allows you to get warnings in an operation by specifying the "warning" query parameter as "yes". For example, to get warnings while connecting to the Citrix ADC appliance, the URL is as follows:

  http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/login?warning=yes

If any, the warnings are displayed in the response payload with the HTTP code "209 X-NITRO-WARNING".

- **Authenticated access for individual NITRO operations:**NITRO allows you to logon to the Citrix ADC appliance to perform individual operations. You can use this option instead of creating a NITRO session (using the login object) and then using that session to perform all operations,

  To do this, you must specify the username and password in the request header of the NITRO request as follows:

  X-NITRO-USER:<username>

  X-NITRO-PASS:<password>

  Note: In such cases, make sure that the request header DOES not include the following:

  Cookie:NITRO_AUTH_TOKEN=<tokenvalue>

`*Note:`*

Mandatory parameters are marked in redand placeholder content is marked in <green>.

update

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/videooptimizationparameter

HTTP Method:PUT

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Content-Type:application/json

Request Payload:

```
{"videooptimizationparameter":{

"randomsamplingpercentage":<Double_value>,

"quicpacingrate":<Double_value>

}}
```

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error

unset

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/videooptimizationparameter?**action=unset**

HTTP Method:POST

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Content-Type:application/json

Request Payload:

```
{"videooptimizationparameter":{

"randomsamplingpercentage":true,

"quicpacingrate":true

}}
```

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error

get (all)

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/videooptimizationparameter

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "videooptimizationparameter": [ {

"randomsamplingpercentage":<Double_value>,

"quicpacingrate":<Double_value>

}]}
```

videooptimizationdetectionaction

Configuration for videooptimization detectionaction resource.

Properties

(click to see Operations)

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
name	<String>	Read-write	Name for the video optimization detection action. Must begin with a letter, number, or the underscore character (_), and must contain only letters, numbers, and the

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
			hyphen (-), period (.) hash (#), space ( ), at (@), equals (=), colon (:), and underscore characters.
type	<String>	Read-write	Type of video optimization action. Available settings function as follows: * clear_text_pd - Cleartext PD type is detected. * clear_text_abr - Cleartext ABR is detected. * encrypted_abr - Encrypted ABR is detected. * trigger_enc_abr - Possible encrypted ABR is detected. * trigger_body_detection - Possible cleartext ABR is detected. Triggers body content detection. Possible values = clear_text_pd, clear_text_abr, encrypted_abr,

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
			trigger_enc_abr, trigger_body_detection
comment	<String>	Read-write	Comment. Any type of information about this video optimization detection action.
newname	<String>	Read-write	New name for the videooptimization detection action. Must begin with a letter, number, or the underscore character (_), and must contain only letters, numbers, and the hyphen (-), period (.) hash (#), space ( ), at (@), equals (=), colon (:), and underscore characters. Minimum length = 1
hits	<Double>	Read-only	The number of times the action has been taken.
referencecount	<Double>	Read-only	The number of references to the action.
undefhits	<Double>	Read-only	The number of times the action resulted in UNDEF.

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
__count	<Double>	Read-only	count parameter

Operations

(click to see Properties)

ADD| DELETE| UPDATE| UNSET| RENAME| GET (ALL)| GET| COUNT

Some options that you can use for each operations:

- **Getting warnings in response:**NITRO allows you to get warnings in an operation by specifying the "warning" query parameter as "yes". For example, to get warnings while connecting to the Citrix ADC appliance, the URL is as follows:

  http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/login?warning=yes

  If any, the warnings are displayed in the response payload with the HTTP code "209 X-NITRO-WARNING".

- **Authenticated access for individual NITRO operations:**NITRO allows you to logon to the Citrix ADC appliance to perform individual operations. You can use this option instead of creating a NITRO session (using the login object) and then using that session to perform all operations,

  To do this, you must specify the username and password in the request header of the NITRO request as follows:

  X-NITRO-USER:<username>

  X-NITRO-PASS:<password>

  Note: In such cases, make sure that the request header DOES not include the following:

  Cookie:NITRO_AUTH_TOKEN=<tokenvalue>

`*Note:`*

Mandatory parameters are marked in redand placeholder content is marked in <green>.

add

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/videooptimizationdetectionaction

HTTP Method:POST

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Content-Type:application/json

Request Payload:

```
{"videooptimizationdetectionaction":{

<b>"name":<String_value>,

</b><b>"type":<String_value>,

</b>"comment":<String_value>

}}
```

Response:

HTTP Status Code on Success: 201 Created HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error

delete

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/videooptimizationdetectionaction/name_value<String>

HTTP Method:DELETE

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue>

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error

update

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/videooptimizationdetectionaction

HTTP Method:PUT

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Content-Type:application/json

Request Payload:

```
{"videooptimizationdetectionaction":{

<b>"name":<String_value>,

</b>"type":<String_value>,

"comment":<String_value>

}}
```

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error

unset

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/videooptimizationdetectionaction?**action=unset**

HTTP Method:POST

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Content-Type:application/json

Request Payload:

```
{"videooptimizationdetectionaction":{
```

```
<b>"name":<String_value>,

</b>"comment":true

}}
```

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error

rename

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/videooptimizationdetectionaction?**action=rename**

HTTP Method:POST

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Content-Type:application/json

Request Payload:

```
{"videooptimizationdetectionaction":{

<b>"name":<String_value>,

</b><b>"newname":<String_value>

</b>}}
```

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error

get (all)

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/videooptimizationdetectionaction

Query-parameters:

attrs

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/videooptimizationdetectionaction?**attrs=property-name1,property-name2**

Use this query parameter to specify the resource details that you want to retrieve.

filter

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/videooptimizationdetectionaction?**filter=property-name1:property-val1,property-name2:property-val2**

Use this query-parameter to get the filtered set of videooptimizationdetectionaction resources configured on Citrix ADC. Filtering can be done on any of the properties of the resource.

view

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/videooptimizationdetectionaction?**view=summary**

Note: By default, the retrieved results are displayed in detail view (?view=detail).

pagination

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/videooptimizationdetectionaction?**pagesize=#no;pageno=#no**

Use this query-parameter to get the videooptimizationdetectionaction resources in chunks.

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "videooptimizationdetectionaction": [ {

"name":<String_value>,

"type":<String_value>,

"hits":<Double_value>,

"referencecount":<Double_value>,

"undefhits":<Double_value>,

"comment":<String_value>

}]}
```

get

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/videooptimizationdetectionaction/name_value<String>

Query-parameters:

attrs

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/videooptimizationdetectionaction/name_value<String>?**attrs=property-name1,property-name2**

Use this query parameter to specify the resource details that you want to retrieve.

view

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/videooptimizationdetectionaction/name_value<String>?**view=summary**

Note: By default, the retrieved results are displayed in detail view (?view=detail).

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "videooptimizationdetectionaction": [ {

"name":<String_value>,

"type":<String_value>,

"hits":<Double_value>,

"referencecount":<Double_value>,

"undefhits":<Double_value>,

"comment":<String_value>

}]}
```

count

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/videooptimizationdetectionaction?**count=yes**

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "videooptimizationdetectionaction": [ { "__count": "#no"} ] }
```

videooptimizationdetectionpolicy

Configuration for videooptimization detectionpolicy resource.

Properties

(click to see Operations)

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
name	<String>	Read-write	Name for the videooptimization detection policy. Must begin with a letter, number, or the underscore character (_), and must contain only letters, numbers, and the hyphen (-), period (.) pound (#), space ( ), at (@), equals (=), colon (:), and underscore characters.Can be

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
			modified, removed or renamed.
rule	<String>	Read-write	Expression that determines which request or response match the video optimization detection policy. The following requirements apply only to the Citrix ADC CLI: * If the expression includes one or more spaces, enclose the entire expression in double quotation marks. * If the expression itself includes double quotation marks, escape the quotations by using the \ character. * Alternatively, you can use single quotation marks to enclose the rule, in which case you do not have to escape the double quotation marks.

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
action	<String>	Read-write	Name of the videooptimization detection action to perform if the request matches this videooptimization detection policy. Built-in actions should be used. These are: * DETECT_CLEARTEXT_PD - Cleartext PD is detected and increment related counters. * DETECT_CLEARTEXT_ABR - Cleartext ABR is detected and increment related counters. * DETECT_ENCRYPTED_ABR - Encrypted ABR is detected and increment related counters. * TRIGGER_ENC_ABR_DETECTION - This is potentially encrypted ABR.

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
			Internal traffic heuristics algorithms will further process traffic to confirm detection. * TRIGGER_CT_ABR_BODY_DETECTION - This is potentially cleartext ABR. Internal traffic heuristics algorithms will further process traffic to confirm detection. * RESET - Reset the client connection by closing it. * DROP - Drop the connection without sending a response.
undefaction	<String>	Read-write	Action to perform if the result of policy evaluation is undefined (UNDEF). An UNDEF event indicates an internal error condition. Only the above built-in actions can be used.

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
comment	<String>	Read-write	Any type of information about this videooptimization detection policy.
logaction	<String>	Read-write	Name of the messagelog action to use for requests that match this policy.
newname	<String>	Read-write	New name for the videooptimization detection policy. Must begin with a letter, number, or the underscore character (_), and must contain only letters, numbers, and the hyphen (-), period (.) hash (#), space ( ), at (@), equals (=), colon (:), and underscore characters. Minimum length = 1
hits	<Double>	Read-only	Number of hits.
undefhits	<Double>	Read-only	Number of policy UNDEF hits.
__count	<Double>	Read-only	count parameter

Operations

(click to see Properties)

ADD| DELETE| UPDATE| UNSET| RENAME| GET (ALL)| GET| COUNT

Some options that you can use for each operations:

- **Getting warnings in response:**NITRO allows you to get warnings in an operation by specifying the "warning" query parameter as "yes". For example, to get warnings while connecting to the Citrix ADC appliance, the URL is as follows:

  http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/login?warning=yes

  If any, the warnings are displayed in the response payload with the HTTP code "209 X-NITRO-WARNING".

- **Authenticated access for individual NITRO operations:**NITRO allows you to logon to the Citrix ADC appliance to perform individual operations. You can use this option instead of creating a NITRO session (using the login object) and then using that session to perform all operations,

  To do this, you must specify the username and password in the request header of the NITRO request as follows:

  X-NITRO-USER:<username>

  X-NITRO-PASS:<password>

  Note: In such cases, make sure that the request header DOES not include the following:

  Cookie:NITRO_AUTH_TOKEN=<tokenvalue>

`*Note:`*

Mandatory parameters are marked in redand placeholder content is marked in <green>.

add

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/videooptimizationdetectionpolicy

HTTP Method:POST

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Content-Type:application/json

Request Payload:

```
{"videooptimizationdetectionpolicy":{

<b>"name":<String_value>,

</b><b>"rule":<String_value>,

</b><b>"action":<String_value>,

</b>"undefaction":<String_value>,

"comment":<String_value>,

"logaction":<String_value>

}}
```

Response:

HTTP Status Code on Success: 201 Created HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error

delete

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/videooptimizationdetectionpolicy/name_value<String>

HTTP Method:DELETE

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue>

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error

update

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/videooptimizationdetectionpolicy

HTTP Method:PUT

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Content-Type:application/json

Request Payload:

```
{"videooptimizationdetectionpolicy":{

<b>"name":<String_value>,

</b>"rule":<String_value>,

"action":<String_value>,

"undefaction":<String_value>,

"comment":<String_value>,

"logaction":<String_value>

}}
```

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error

unset

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/videooptimizationdetectionpolicy?**action=unset**

HTTP Method:POST

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Content-Type:application/json

Request Payload:

```
{"videooptimizationdetectionpolicy":{

<b>"name":<String_value>,

</b>"undefaction":true,

"comment":true,

"logaction":true

}}
```

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error

rename

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/videooptimizationdetectionpolicy?**action=rename**

HTTP Method:POST

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Content-Type:application/json

Request Payload:

```
{"videooptimizationdetectionpolicy":{

<b>"name":<String_value>,
```

```
</b><b>"newname":<String_value>

</b>}}
```

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error

get (all)

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/videooptimizationdetectionpolicy

Query-parameters:

attrs

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/videooptimizationdetectionpolicy?**attrs=property-name1,property-name2**

Use this query parameter to specify the resource details that you want to retrieve.

filter

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/videooptimizationdetectionpolicy?**filter=property-name1:property-val1,property-name2:property-val2**

Use this query-parameter to get the filtered set of videooptimizationdetectionpolicy resources configured on Citrix ADC. Filtering can be done on any of the properties of the resource.

view

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/videooptimizationdetectionpolicy?**view=summary**

Note: By default, the retrieved results are displayed in detail view (?view=detail).

pagination

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/videooptimizationdetectionpolicy?**pagesize=#no;pageno=#no**

Use this query-parameter to get the videooptimizationdetectionpolicy resources in chunks.

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "videooptimizationdetectionpolicy": [ {

"name":<String_value>,

"rule":<String_value>,

"action":<String_value>,

"undefaction":<String_value>,

"hits":<Double_value>,

"undefhits":<Double_value>,

"comment":<String_value>,

"logaction":<String_value>

}]}
```

get

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/videooptimizationdetectionpolicy/name_value<String>

Query-parameters:

attrs

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/videooptimizationdetectionpolicy/name_value<String>?**attrs=property-name1,property-name2**

Use this query parameter to specify the resource details that you want to retrieve.

view

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/videooptimizationdetectionpolicy/name_value<String>?**view=summary**

Note: By default, the retrieved results are displayed in detail view (?view=detail).

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "videooptimizationdetectionpolicy": [ {

"name":<String_value>,

"rule":<String_value>,
```

```
"action":<String_value>,

"undefaction":<String_value>,

"hits":<Double_value>,

"undefhits":<Double_value>,

"comment":<String_value>,

"logaction":<String_value>

}]}
```

count

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/videooptimizationdetectionpolicy?**count=yes**

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "videooptimizationdetectionpolicy": [ { "__count": "#no"} ] }
```

videooptimizationdetectionpolicy_binding

Binding object which returns the resources bound to videooptimizationdetectionpolicy.

Properties

(click to see Operations)

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
name	<String>	Read-write	Name of the videooptimization detection policy for which to display settings.Must provide policy name.
videooptimizationdetectionpolicy_lbvserver_binding	<videooptimizationdetectionpolicy_lbvserver_binding[]>	Read-only	lbvserver that can be bound to videooptimizationdetectionpolicy.
videooptimizationdetectionpolicy_videooptimizationglobaldetection_binding	<videooptimizationdetectionpolicy_videooptimizationglobaldetection_binding[]>	Read-only	videooptimizationglobaldetection that can be bound to videooptimizationdetectionpolicy.

Operations

(click to see Properties)

GET| GET (ALL)

Some options that you can use for each operations:

- **Getting warnings in response:**NITRO allows you to get warnings in an operation by specifying the "warning" query parameter as "yes". For example, to get warnings while connecting to the Citrix ADC appliance, the URL is as follows:

  http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/login?warning=yes

  If any, the warnings are displayed in the response payload with the HTTP code "209 X-NITRO-WARNING".

- **Authenticated access for individual NITRO operations:**NITRO allows you to logon to the Citrix ADC appliance to perform individual operations. You can use this option instead of creating a NITRO session (using the login object) and then using that session to perform all operations,

  To do this, you must specify the username and password in the request header of the NITRO request as follows:

  X-NITRO-USER:<username>

  X-NITRO-PASS:<password>

  Note: In such cases, make sure that the request header DOES not include the following:

  Cookie:NITRO_AUTH_TOKEN=<tokenvalue>

`*Note:`*

Mandatory parameters are marked in redand placeholder content is marked in <green>.

get

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/videooptimizationdetectionpolicy_binding/name_value<String>

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "videooptimizationdetectionpolicy_binding": [ {
```

```
"name":<String_value>,

"videooptimizationdetectionpolicy_videooptimizationglobaldetection_binding
":<videooptimizationdetectionpolicy_videooptimizationglobaldetection_bindi
ng[]_value>,

"videooptimizationdetectionpolicy_lbvserver_binding":<videooptimizationdet
ectionpolicy_lbvserver_binding[]_value>

}]}
```

get (all)

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/videooptimizationdetectionpolicy_binding

Query-parameters:

bulkbindings

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/videooptimizationdetectionpolicy_binding?**bulkbindings=yes**

NITRO allows you to fetch bindings in bulk.

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "videooptimizationdetectionpolicy_binding": [ {

"name":<String_value>,

"videooptimizationdetectionpolicy_videooptimizationglobaldetection_binding
":<videooptimizationdetectionpolicy_videooptimizationglobaldetection_bindi
ng[]_value>,

"videooptimizationdetectionpolicy_lbvserver_binding":<videooptimizationdet
ectionpolicy_lbvserver_binding[]_value>

}]}
```

videooptimizationdetectionpolicy_lbvserver_binding

Binding object showing the lbvserver that can be bound to videooptimizationdetectionpolicy.

Properties

(click to see Operations)

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
boundto	<String>	Read-write	Location where policy is bound.
name	<String>	Read-write	Name of the videooptimization detection policy for which to display settings.Must provide policy name.
priority	<Double>	Read-only	Specifies the priority of the policy.
labelname	<String>	Read-only	Name of the label to invoke if the current policy rule evaluates to TRUE.

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
gotopriorityexpression	<String>	Read-only	Expression specifying the priority of the next policy which will get evaluated if the current policy rule evaluates to TRUE.
labeltype	<String>	Read-only	Type of policy label invocation. Possible values = vserver, policylabel
activepolicy	<Integer>	Read-only	Indicates whether policy is bound or not.
__count	<Double>	Read-only	count parameter

Operations

(click to see Properties)

GET| GET (ALL)| COUNT

Some options that you can use for each operations:

- **Getting warnings in response:**NITRO allows you to get warnings in an operation by specifying the "warning" query parameter as "yes". For example, to get warnings while connecting to the Citrix ADC appliance, the URL is as follows:

  http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/login?warning=yes

  If any, the warnings are displayed in the response payload with the HTTP code "209 X-NITRO-WARNING".

- **Authenticated access for individual NITRO operations:**NITRO allows you to logon to the Citrix ADC appliance to perform individual operations. You can use this option instead of creating a NITRO session (using the login object) and then using that session to perform all operations,

To do this, you must specify the username and password in the request header of the NITRO request as follows:

X-NITRO-USER:<username>

X-NITRO-PASS:<password>

Note: In such cases, make sure that the request header DOES not include the following:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue>

`*Note:`*

Mandatory parameters are marked in redand placeholder content is marked in <green>.

get

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/videooptimizationdetectionpolicy_lbvserver_binding/name_value<String>

Query-parameters:

filter

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/videooptimizationdetectionpolicy_lbvserver_binding/name_value<String>?**filter=property-name1:property-value1,property-name2:property-value2**

Use this query-parameter to get the filtered set of videooptimizationdetectionpolicy_lbvserver_binding resources configured on Citrix ADC. Filtering can be done on any of the properties of the resource.

pagination

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/videooptimizationdetectionpolicy_lbvserver_binding/name_value<String>?**pagesize=#no;pageno=#no**

Use this query-parameter to get the videooptimizationdetectionpolicy_lbvserver_binding resources in chunks.

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "videooptimizationdetectionpolicy_lbvserver_binding": [ {

"boundto":<String_value>,

"name":<String_value>,

"priority":<Double_value>,

"labelname":<String_value>,

"gotopriorityexpression":<String_value>,

"labeltype":<String_value>,

"activepolicy":<Integer_value>

}]}
```

get (all)

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/videooptimizationdetectionpolicy_lbvserver_binding

Query-parameters:

bulkbindings

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/videooptimizationdetectionpolicy_lbvserver_binding?**bulkbindings=yes**

NITRO allows you to fetch bindings in bulk.

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "videooptimizationdetectionpolicy_lbvserver_binding": [ {

"boundto":<String_value>,

"name":<String_value>,

"priority":<Double_value>,

"labelname":<String_value>,

"gotopriorityexpression":<String_value>,

"labeltype":<String_value>,

"activepolicy":<Integer_value>

}]}
```

count

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/videooptimizationdetectionpolicy_lbvserver_binding/name_value<String>?**count=yes**

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{"videooptimizationdetectionpolicy_lbvserver_binding": [ { "__count":
"#no"} ] }
```

videooptimizationdetectionpolicylabel

Configuration for videooptimization detection policy label resource.

Properties

(click to see Operations)

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
labelname	<String>	Read-write	Name for the Video optimization detection policy label. Must begin with a letter, number, or the underscore character (_), and must contain only letters, numbers, and the hyphen (-), period (

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
			.) hash (#), space ( ), at (@), equals (=), colon (:), and underscore characters. Cannot be changed after the videooptimization detection policy label is added. The following requirement applies only to the Citrix ADC CLI: If the name includes one or more spaces, enclose the name in double or single quotation marks (for example, "my videooptimization detection policy label" or my videooptimization detection policy label').
policylabeltype	<String>	Read-write	Type of responses sent by the policies bound to this policy label. Types are: * HTTP - HTTP responses.

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
			* OTHERTCP - NON-HTTP TCP responses. Default value: NS_PLTMAP_RSP _REQ Possible values = videoopt_req, videoopt_res
comment	<String>	Read-write	Any comments to preserve information about this videooptimization detection policy label.
newname	<String>	Read-write	New name for the videooptimization detection policy label. Must begin with a letter, number, or the underscore character (_), and must contain only letters, numbers, and the hyphen (-), period (.) hash (#), space ( ), at (@), equals (=), colon (:), and underscore characters. Minimum length = 1
numpol	<Double>	Read-only	number of polices bound to label.

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
hits	<Double>	Read-only	Number of times policy label was invoked.
priority	<Double>	Read-only	Specifies the priority of the policy.
gotopriorityexpression	<String>	Read-only	Expression specifying the priority of the next policy which will get evaluated if the current policy rule evaluates to TRUE.
labeltype	<String>	Read-only	Type of policy label to invoke. Available settings function as follows: * vserver - Invoke an unnamed policy label associated with a virtual server. * policylabel - Invoke a user-defined policy label. Possible values = vserver, policylabel
invoke_labelname	<String>	Read-only	* If labelType is policylabel, name of the policy label to invoke. * If labelType is reqvserver or resvserver, name of the virtual server.

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
__count	<Double>	Read-only	count parameter

Operations

(click to see Properties)

ADD| DELETE| RENAME| GET (ALL)| GET| COUNT

Some options that you can use for each operations:

- **Getting warnings in response:**NITRO allows you to get warnings in an operation by specifying the "warning" query parameter as "yes". For example, to get warnings while connecting to the Citrix ADC appliance, the URL is as follows:

  http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/login?warning=yes

  If any, the warnings are displayed in the response payload with the HTTP code "209 X-NITRO-WARNING".
- **Authenticated access for individual NITRO operations:**NITRO allows you to logon to the Citrix ADC appliance to perform individual operations. You can use this option instead of creating a NITRO session (using the login object) and then using that session to perform all operations,

  To do this, you must specify the username and password in the request header of the NITRO request as follows:

  X-NITRO-USER:<username>

  X-NITRO-PASS:<password>

  Note: In such cases, make sure that the request header DOES not include the following:

  Cookie:NITRO_AUTH_TOKEN=<tokenvalue>

`*Note:`*

Mandatory parameters are marked in redand placeholder content is marked in <green>.

add

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/videooptimizationdetectionpolicylabel

HTTP Method:POST

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Content-Type:application/json

Request Payload:

```
{"videooptimizationdetectionpolicylabel":{

<b>"labelname":<String_value>,

</b>"policylabeltype":<String_value>,

"comment":<String_value>

}}
```

Response:

HTTP Status Code on Success: 201 Created HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error

delete

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/videooptimizationdetectionpolicylabel/labelname_value<String>

HTTP Method:DELETE

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue>

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error

rename

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/videooptimizationdetectionpolicylabel?**action=rename**

HTTP Method:POST

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Content-Type:application/json

Request Payload:

```
{"videooptimizationdetectionpolicylabel":{

<b>"labelname":<String_value>,

</b><b>"newname":<String_value>

</b>}}
```

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error

get (all)

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/videooptimizationdetectionpolicylabel

Query-parameters:

attrs

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/videooptimizationdetectionpolicylabel?**attrs=property-name1,property-name2**

Use this query parameter to specify the resource details that you want to retrieve.

filter

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/videooptimizationdetectionpolicylabel?**filter=property-name1:property-val1,property-name2:property-val2**

Use this query-parameter to get the filtered set of videooptimizationdetectionpolicylabel resources configured on Citrix ADC. Filtering can be done on any of the properties of the resource.

view

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/videooptimizationdetectionpolicylabel?**view=summary**

Note: By default, the retrieved results are displayed in detail view (?view=detail).

pagination

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/videooptimizationdetectionpolicylabel?**pagesize=#no;pageno=#no**

Use this query-parameter to get the videooptimizationdetectionpolicylabel resources in chunks.

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "videooptimizationdetectionpolicylabel": [ {

"labelname":<String_value>,

"policylabeltype":<String_value>,
```

```
"numpol":<Double_value>,

"hits":<Double_value>,

"priority":<Double_value>,

"gotopriorityexpression":<String_value>,

"labeltype":<String_value>,

"invoke_labelname":<String_value>,

"comment":<String_value>

}]}
```

get

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/videooptimizationdetectionpolicylabel/labelname_value<String>

Query-parameters:

attrs

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/videooptimizationdetectionpolicylabel/labelname_value<String>?**attrs=property-name1,property-name2**

Use this query parameter to specify the resource details that you want to retrieve.

view

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/videooptimizationdetectionpolicylabel/labelname_value<String>?**view=summary**

Note: By default, the retrieved results are displayed in detail view (?view=detail).

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "videooptimizationdetectionpolicylabel": [ {

"labelname":<String_value>,

"policylabeltype":<String_value>,

"numpol":<Double_value>,

"hits":<Double_value>,

"priority":<Double_value>,

"gotopriorityexpression":<String_value>,

"labeltype":<String_value>,

"invoke_labelname":<String_value>,

"comment":<String_value>

}]}
```

count

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/videooptimizationdetectionpolicylabel?**count=yes**

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "videooptimizationdetectionpolicylabel": [ { "__count": "#no"} ] }
```

videooptimizationdetectionpolicylabel_binding

Binding object which returns the resources bound to videooptimizationdetectionpolicylabel.

Properties

(click to see Operations)

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
labelname	<String>	Read-write	Name of the videooptimization detection policy label.
videooptimizationdetectionpolicylabel_policybinding_binding	<videooptimizationdetectionpolicylabel_policybinding_binding[]>	Read-only	policybinding that can be bound to videooptimizationdetectionpolicylabel.
videooptimizationdetectionpolicylabel_videooptimizationdetectionpolicy_binding	<videooptimizationdetectionpolicylabel_videooptimizationdetectionpolicy_binding[]>	Read-only	videooptimizationdetectionpolicy that can be bound to videooptimizationdetectionpolicylabel.

Operations

(click to see Properties)

GET| GET (ALL)

Some options that you can use for each operations:

- **Getting warnings in response:**NITRO allows you to get warnings in an operation by specifying the "warning" query parameter as "yes". For example, to get warnings while connecting to the Citrix ADC appliance, the URL is as follows:

  http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/login?warning=yes

  If any, the warnings are displayed in the response payload with the HTTP code "209 X-NITRO-WARNING".

- **Authenticated access for individual NITRO operations:**NITRO allows you to logon to the Citrix ADC appliance to perform individual operations. You can use this option instead of creating a NITRO session (using the login object) and then using that session to perform all operations,

  To do this, you must specify the username and password in the request header of the NITRO request as follows:

  X-NITRO-USER:<username>

  X-NITRO-PASS:<password>

  Note: In such cases, make sure that the request header DOES not include the following:

  Cookie:NITRO_AUTH_TOKEN=<tokenvalue>

`*Note:`*

Mandatory parameters are marked in redand placeholder content is marked in <green>.

get

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/videooptimizationdetectionpolicylabel_binding/labelname_value<String>

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "videooptimizationdetectionpolicylabel_binding": [ {

"labelname":<String_value>,

"videooptimizationdetectionpolicylabel_videooptimizationdetectionpolicy_bi
nding":<videooptimizationdetectionpolicylabel_videooptimizationdetectionpo
licy_binding[]_value>,

"videooptimizationdetectionpolicylabel_policybinding_binding":<videooptimi
zationdetectionpolicylabel_policybinding_binding[]_value>

}]}
```

get (all)

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/videooptimizationdetectionpolicylabel_binding

Query-parameters:

bulkbindings

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/videooptimizationdetectionpolicylabel_binding?**bulkbindings=yes**

NITRO allows you to fetch bindings in bulk.

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "videooptimizationdetectionpolicylabel_binding": [ {

"labelname":<String_value>,

"videooptimizationdetectionpolicylabel_videooptimizationdetectionpolicy_binding":<videooptimizationdetectionpolicylabel_videooptimizationdetectionpolicy_binding[]_value>,

"videooptimizationdetectionpolicylabel_policybinding_binding":<videooptimizationdetectionpolicylabel_policybinding_binding[]_value>

}]}
```

videooptimizationdetectionpolicylabel_policybinding_binding

Binding object showing the policybinding that can be bound to videooptimizationdetectionpolicylabel.

Properties

(click to see Operations)

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
priority	<Double>	Read-write	Specifies the priority of the policy.

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
policyname	<String>	Read-write	Name of the videooptimization policy.
labelname	<String>	Read-write	Name of the videooptimization detection policy label to which to bind the policy.
invoke_labelname	<String>	Read-write	* If labelType is policylabel, name of the policy label to invoke. * If labelType is reqvserver or resvserver, name of the virtual server.
gotopriorityexpression	<String>	Read-write	Expression specifying the priority of the next policy which will get evaluated if the current policy rule evaluates to TRUE.
invoke	<Boolean>	Read-write	If the current policy evaluates to TRUE, terminate evaluation of policies bound to the current policy label and evaluate the specified policy label.
labeltype	<String>	Read-write	Type of policy label to invoke. Available settings function as follows: * vserver - Invoke an unnamed policy label

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
			associated with a virtual server. * policylabel - Invoke a user-defined policy label. Possible values = vserver, policylabel
__count	<Double>	Read-write	count parameter

Operations

(click to see Properties)

GET| GET (ALL)| COUNT

Some options that you can use for each operations:

- **Getting warnings in response:**NITRO allows you to get warnings in an operation by specifying the "warning" query parameter as "yes". For example, to get warnings while connecting to the Citrix ADC appliance, the URL is as follows:

  http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/login?warning=yes

  If any, the warnings are displayed in the response payload with the HTTP code "209 X-NITRO-WARNING".

- **Authenticated access for individual NITRO operations:**NITRO allows you to logon to the Citrix ADC appliance to perform individual operations. You can use this option instead of creating a NITRO session (using the login object) and then using that session to perform all operations,

  To do this, you must specify the username and password in the request header of the NITRO request as follows:

  X-NITRO-USER:<username>

  X-NITRO-PASS:<password>

  Note: In such cases, make sure that the request header DOES not include the following:

  Cookie:NITRO_AUTH_TOKEN=<tokenvalue>

`*Note:`*

Mandatory parameters are marked in redand placeholder content is marked in <green>.

get

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/videooptimizationdetectionpolicylabel_policybinding_binding/labelname_value<String>

Query-parameters:

filter

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/videooptimizationdetectionpolicylabel_policybinding_binding/labelname_value<String>?**filter=property-name1:property-value1,property-name2:property-value2**

Use this query-parameter to get the filtered set of videooptimizationdetectionpolicylabel_policybinding_binding resources configured on Citrix ADC. Filtering can be done on any of the properties of the resource.

pagination

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/videooptimizationdetectionpolicylabel_policybinding_binding/labelname_value<String>?**pagesize=#no;pageno=#no**

Use this query-parameter to get the videooptimizationdetectionpolicylabel_policybinding_binding resources in chunks.

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "videooptimizationdetectionpolicylabel_policybinding_binding": [ {

"priority":<Double_value>,

"policyname":<String_value>,

"labelname":<String_value>,

"invoke_labelname":<String_value>,

"gotopriorityexpression":<String_value>,

"invoke":<Boolean_value>,

"labeltype":<String_value>

}]}
```

get (all)

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/videooptimizationdetectionpolicylabel_policybinding_binding

Query-parameters:

bulkbindings

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/videooptimizationdetectionpolicylabel_policybinding_binding?**bulkbindings=yes**

NITRO allows you to fetch bindings in bulk.

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "videooptimizationdetectionpolicylabel_policybinding_binding": [ {

"priority":<Double_value>,

"policyname":<String_value>,

"labelname":<String_value>,

"invoke_labelname":<String_value>,

"gotopriorityexpression":<String_value>,

"invoke":<Boolean_value>,

"labeltype":<String_value>

}]}
```

count

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/videooptimizationdetectionpolicylabel_policybinding_binding/labelname_value<String>?**count=yes**

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{"videooptimizationdetectionpolicylabel_policybinding_binding":
[ { "__count": "#no"} ] }
```

videooptimizationglobaldetection_binding

Binding object which returns the resources bound to videooptimizationglobaldetection.

Properties

(click to see Operations)

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
videooptimizationglobaldetection_videooptimizationdetectionpolicy_binding	<videooptimizationglobaldetection_videooptimizationdetectionpolicy_binding[]>	Read-only	videooptimizationdetectionpolicy that can be bound to videooptimizationglobaldetection.

Operations

(click to see Properties)

GET| GET (ALL)

Some options that you can use for each operations:

- **Getting warnings in response:**NITRO allows you to get warnings in an operation by specifying the "warning" query parameter as "yes". For example, to get warnings while connecting to the Citrix ADC appliance, the URL is as follows:

  http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/login?warning=yes

If any, the warnings are displayed in the response payload with the HTTP code "209 X-NITRO-WARNING".

- **Authenticated access for individual NITRO operations:**NITRO allows you to logon to the Citrix ADC appliance to perform individual operations. You can use this option instead of creating a NITRO session (using the login object) and then using that session to perform all operations,

  To do this, you must specify the username and password in the request header of the NITRO request as follows:

  X-NITRO-USER:<username>

  X-NITRO-PASS:<password>

  Note: In such cases, make sure that the request header DOES not include the following:

  Cookie:NITRO_AUTH_TOKEN=<tokenvalue>

`*Note:`*

Mandatory parameters are marked in redand placeholder content is marked in <green>.

get

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/videooptimizationglobaldetection_binding

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "videooptimizationglobaldetection_binding": [ {

"videooptimizationglobaldetection_videooptimizationdetectionpolicy_binding
":<videooptimizationglobaldetection_videooptimizationdetectionpolicy_bindi
ng[]_value>

}]}
```

get (all)

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/videooptimizationglobaldetection_binding

Query-parameters:

bulkbindings

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/videooptimizationglobaldetection_binding?**bulkbindings=yes**

NITRO allows you to fetch bindings in bulk.

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "videooptimizationglobaldetection_binding": [ {
```

```
"videooptimizationglobaldetection_videooptimizationdetectionpolicy_binding
":<videooptimizationglobaldetection_videooptimizationdetectionpolicy_bindi
ng[]_value>

}]}
```

videooptimizationglobalpacing_binding

Binding object which returns the resources bound to videooptimizationglobalpacing.

Properties

(click to see Operations)

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
videooptimizationglobalpacing_videooptimizationpacingpolicy_binding	<videooptimizationglobalpacing_videooptimizationpacingpolicy_binding[]>	Read-only	videooptimizationpacingpolicy that can be bound to videooptimizationglobalpacing.

Operations

(click to see Properties)

GET| GET (ALL)

Some options that you can use for each operations:

- **Getting warnings in response:**NITRO allows you to get warnings in an operation by specifying the "warning" query parameter as "yes". For example, to get warnings while connecting to the Citrix ADC appliance, the URL is as follows:

  http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/login?warning=yes

  If any, the warnings are displayed in the response payload with the HTTP code "209 X-NITRO-WARNING".
- **Authenticated access for individual NITRO operations:**NITRO allows you to logon to the Citrix ADC appliance to perform individual operations. You can use this option instead of

creating a NITRO session (using the login object) and then using that session to perform all operations,

To do this, you must specify the username and password in the request header of the NITRO request as follows:

X-NITRO-USER:<username>

X-NITRO-PASS:<password>

Note: In such cases, make sure that the request header DOES not include the following:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue>

`*Note:`*

Mandatory parameters are marked in redand placeholder content is marked in <green>.

get

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/videooptimizationglobalpacing_binding

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "videooptimizationglobalpacing_binding": [ {

"videooptimizationglobalpacing_videooptimizationpacingpolicy_binding":<vid
eooptimizationglobalpacing_videooptimizationpacingpolicy_binding[]_value>

}]}
```

get (all)

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/videooptimizationglobalpacing_binding

Query-parameters:

bulkbindings

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/videooptimizationglobalpacing_binding?**bulkbindings=yes**

NITRO allows you to fetch bindings in bulk.

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "videooptimizationglobalpacing_binding": [ {

"videooptimizationglobalpacing_videooptimizationpacingpolicy_binding":<vid
eooptimizationglobalpacing_videooptimizationpacingpolicy_binding[]_value>

}]}
```

videooptimizationglobalpacing_videooptimizationpacingpolicy_binding

Binding object showing the videooptimizationpacingpolicy that can be bound to videooptimizationglobalpacing.

Properties

(click to see Operations)

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
priority	<Double>	Read-write	Specifies the priority of the policy.
globalbindtype	<String>	Read-write	. Default value: SYSTEM_GLOBAL Possible values = SYSTEM_GLOBAL, VPN_GLOBAL, RNAT_GLOBAL
policyname	<String>	Read-write	Name of the videooptimization pacing policy.
labelname	<String>	Read-write	Name of the policy label to invoke. If the current policy evaluates to TRUE, the invoke parameter is set, and Label Type is policylabel.
gotopriorityexpression	<String>	Read-write	Expression specifying the priority of the next policy which will get evaluated if the current policy rule evaluates to TRUE.

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
invoke	<Boolean>	Read-write	If the current policy evaluates to TRUE, terminate evaluation of policies bound to the current policy label, and then forward the request to the specified virtual server or evaluate the specified policy label.
type	<String>	Read-write	Specifies the bind point whose policies you want to display. Possible values = REQ_OVERRIDE, REQ_DEFAULT, RES_OVERRIDE, RES_DEFAULT
labeltype	<String>	Read-write	Type of invocation, Available settings function as follows: * vserver - Forward the request to the specified virtual server. * policylabel - Invoke the specified policy label. Possible values = vserver, policylabel
numpol	<Double>	Read-only	number of polices bound.
__count	<Double>	Read-only	count parameter

Operations

(click to see Properties)

ADD:| DELETE:| GET| GET (ALL)| COUNT

Some options that you can use for each operations:

- **Getting warnings in response:**NITRO allows you to get warnings in an operation by specifying the "warning" query parameter as "yes". For example, to get warnings while connecting to the Citrix ADC appliance, the URL is as follows:

  http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/login?warning=yes

  If any, the warnings are displayed in the response payload with the HTTP code "209 X-NITRO-WARNING".

- **Authenticated access for individual NITRO operations:**NITRO allows you to logon to the Citrix ADC appliance to perform individual operations. You can use this option instead of creating a NITRO session (using the login object) and then using that session to perform all operations,

  To do this, you must specify the username and password in the request header of the NITRO request as follows:

  X-NITRO-USER:<username>

  X-NITRO-PASS:<password>

  Note: In such cases, make sure that the request header DOES not include the following:

  Cookie:NITRO_AUTH_TOKEN=<tokenvalue>

`*Note:`*

Mandatory parameters are marked in redand placeholder content is marked in <green>.

add

URL:http://<Citrix-ADC-IP-address(NSIP)/nitro/v1/config/videooptimizationglobalpacing_videooptimizationpacingpolicy_binding

HTTP Method:PUT

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Content-Type:application/json

Request Payload:

```
{
"videooptimizationglobalpacing_videooptimizationpacingpolicy_binding":{
<b>"policyname":<String_value>,
</b><b>"priority":<Double_value>,
</b>"gotopriorityexpression":<String_value>,
"type":<String_value>,
"invoke":<Boolean_value>,
"labeltype":<String_value>,
"labelname":<String_value>
}}
```

Response:

HTTP Status Code on Success: 201 Created HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error

delete

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/videooptimizationglobalpacing_videooptimizationpacingpolicy_binding

Query-parameters:

args

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/videooptimizationglobalpacing_videooptimizationpacingpolicy_binding?**args=policyname:<String_value>,type:<String_value>,priority:<Double_value>**

HTTP Method:DELETE

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue>

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error

get

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/videooptimizationglobalpacing_videooptimizationpacingpolicy_binding

Query-parameters:

filter

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/videooptimizationglobalpacing_videooptimizationpacingpolicy_binding?**filter=property-name1:property-value1,property-name2:property-value2**

Use this query-parameter to get the filtered set of videooptimizationglobalpacing_videooptimizationpacingpolicy_binding resources configured on Citrix ADC. Filtering can be done on any of the properties of the resource.

pagination

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/videooptimizationglobalpacing_videooptimizationpacingpolicy_binding?**pagesize=#no;pageno=#no**

Use this query-parameter to get the videooptimizationglobalpacing_videooptimizationpacingpolicy_binding resources in chunks.

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "videooptimizationglobalpacing_videooptimizationpacingpolicy_binding":
[ {

"priority":<Double_value>,

"globalbindtype":<String_value>,

"policyname":<String_value>,

"labelname":<String_value>,

"gotopriorityexpression":<String_value>,

"invoke":<Boolean_value>,

"type":<String_value>,

"labeltype":<String_value>,

"numpol":<Double_value>

}]}
```

get (all)

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/videooptimizationglobalpacing_videooptimizationpacingpolicy_binding

Query-parameters:

bulkbindings

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/videooptimizationglobalpacing_videooptimizationpacingpolicy_binding?**bulkbindings=yes**

NITRO allows you to fetch bindings in bulk.

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "videooptimizationglobalpacing_videooptimizationpacingpolicy_binding":
[ {

"priority":<Double_value>,

"globalbindtype":<String_value>,

"policyname":<String_value>,

"labelname":<String_value>,

"gotopriorityexpression":<String_value>,

"invoke":<Boolean_value>,

"type":<String_value>,

"labeltype":<String_value>,
```

```
"numpol":<Double_value>

}]}
```

count

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/videooptimizationglobalpacing_videooptimizationpacingpolicy_binding?**count=yes**

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "videooptimizationglobalpacing_videooptimizationpacingpolicy_binding":
[ { "__count": "#no"} ] }
```

videooptimizationpacingaction

Configuration for videooptimization pacingaction resource.

Properties

(click to see Operations)

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
name	<String>	Read-write	Name for the video optimization pacing action. Must begin with a letter, number, or the underscore character (_), and must contain only letters, numbers, and the hyphen (-), period (.) hash (#), space ( ), at (@), equals (=), colon (:), and underscore characters.
rate	<Integer>	Read-write	ABR Video Optimization Pacing Rate (in Kbps). Default value: 1000 Minimum value = 1 Maximum value = 2147483647
comment	<String>	Read-write	Comment. Any type of information about this video optimization detection action.
newname	<String>	Read-write	New name for the videooptimization pacing action. Must begin with a letter, number, or the underscore character (_), and must contain only letters, numbers,

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
			and the hyphen (-), period (.) hash (#), space ( ), at (@), equals (=), colon (:), and underscore characters. Minimum length = 1
hits	<Double>	Read-only	The number of times the action has been taken.
referencecount	<Double>	Read-only	The number of references to the action.
undefhits	<Double>	Read-only	The number of times the action resulted in UNDEF.
__count	<Double>	Read-only	count parameter

Operations

(click to see Properties)

ADD| DELETE| UPDATE| UNSET| RENAME| GET (ALL)| GET| COUNT

Some options that you can use for each operations:

- **Getting warnings in response:**NITRO allows you to get warnings in an operation by specifying the "warning" query parameter as "yes". For example, to get warnings while connecting to the Citrix ADC appliance, the URL is as follows:

  http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/login?warning=yes

  If any, the warnings are displayed in the response payload with the HTTP code "209 X-NITRO-WARNING".

- **Authenticated access for individual NITRO operations:**NITRO allows you to logon to the Citrix ADC appliance to perform individual operations. You can use this option instead of creating a NITRO session (using the login object) and then using that session to perform all operations,

  To do this, you must specify the username and password in the request header of the NITRO request as follows:

  X-NITRO-USER:<username>

  X-NITRO-PASS:<password>

  Note: In such cases, make sure that the request header DOES not include the following:

  Cookie:NITRO_AUTH_TOKEN=<tokenvalue>

`*Note:`*

Mandatory parameters are marked in redand placeholder content is marked in <green>.

add

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/videooptimizationpacingaction

HTTP Method:POST

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Content-Type:application/json

Request Payload:

```
{"videooptimizationpacingaction":{

<b>"name":<String_value>,

</b><b>"rate":<Integer_value>,

</b>"comment":<String_value>

}}
```

Response:

HTTP Status Code on Success: 201 Created HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error

delete

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/videooptimizationpacingaction/name_value<String>

HTTP Method:DELETE

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue>

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error

update

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/videooptimizationpacingaction

HTTP Method:PUT

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Content-Type:application/json

Request Payload:

```
{"videooptimizationpacingaction":{

<b>"name":<String_value>,

</b>"rate":<Integer_value>,

"comment":<String_value>

}}
```

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error

unset

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/videooptimizationpacingaction?**action=unset**

HTTP Method:POST

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Content-Type:application/json

Request Payload:

```
{"videooptimizationpacingaction":{

<b>"name":<String_value>,

</b>"rate":true,

"comment":true

}}
```

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error

rename

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/videooptimizationpacingaction?**action=rename**

HTTP Method:POST

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Content-Type:application/json

Request Payload:

```
{"videooptimizationpacingaction":{

<b>"name":<String_value>,

</b><b>"newname":<String_value>

</b>}}
```

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error

get (all)

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/videooptimizationpacingaction

Query-parameters:

attrs

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/videooptimizationpacingaction?**attrs=property-name1,property-name2**

Use this query parameter to specify the resource details that you want to retrieve.

filter

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/videooptimizationpacingaction?**filter=property-name1:property-val1,property-name2:property-val2**

Use this query-parameter to get the filtered set of videooptimizationpacingaction resources configured on Citrix ADC. Filtering can be done on any of the properties of the resource.

view

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/videooptimizationpacingaction?**view=summary**

Note: By default, the retrieved results are displayed in detail view (?view=detail).

pagination

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/videooptimizationpacingaction?**pagesize=#no;pageno=#no**

Use this query-parameter to get the videooptimizationpacingaction resources in chunks.

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "videooptimizationpacingaction": [ {

"name":<String_value>,

"hits":<Double_value>,

"referencecount":<Double_value>,

"undefhits":<Double_value>,
```

```
"rate":<Integer_value>,

"comment":<String_value>

}]}
```

get

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/videooptimizationpacingaction/name_value<String>

Query-parameters:

attrs

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/videooptimizationpacingaction/name_value<String>?**attrs=property-name1,property-name2**

Use this query parameter to specify the resource details that you want to retrieve.

view

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/videooptimizationpacingaction/name_value<String>?**view=summary**

Note: By default, the retrieved results are displayed in detail view (?view=detail).

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "videooptimizationpacingaction": [ {

"name":<String_value>,

"hits":<Double_value>,

"referencecount":<Double_value>,

"undefhits":<Double_value>,

"rate":<Integer_value>,

"comment":<String_value>

}]}
```

count

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/videooptimizationpacingaction?**count=yes**

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "videooptimizationpacingaction": [ { "__count": "#no"} ] }
```

videooptimizationpacingpolicy

Configuration for videooptimization pacingpolicy resource.

Properties

(click to see Operations)

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
name	<String>	Read-write	Name for the videooptimization pacing policy. Must begin with a letter, number, or the underscore character (_), and must contain only letters, numbers, and the hyphen (-), period (.) pound (#), space ( ), at (@), equals (=), colon (:), and underscore characters.Can be modified, removed or renamed.
rule	<String>	Read-write	Expression that determines which request or response match the video optimization pacing policy. The following requirements apply only to the Citrix ADC CLI:

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
			* If the expression includes one or more spaces, enclose the entire expression in double quotation marks. * If the expression itself includes double quotation marks, escape the quotations by using the \ character. * Alternatively, you can use single quotation marks to enclose the rule, in which case you do not have to escape the double quotation marks.
action	<String>	Read-write	Name of the videooptimization pacing action to perform if the request matches this videooptimization pacing policy.
undefaction	<String>	Read-write	Action to perform if the result of policy evaluation is undefined (UNDEF). An UNDEF event indicates an internal error condition. Only

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
			the above built-in actions can be used.
comment	<String>	Read-write	Any type of information about this videooptimization pacing policy.
logaction	<String>	Read-write	Name of the messagelog action to use for requests that match this policy.
newname	<String>	Read-write	New name for the videooptimization pacing policy. Must begin with a letter, number, or the underscore character (_), and must contain only letters, numbers, and the hyphen (-), period (.) hash (#), space ( ), at (@), equals (=), colon (:), and underscore characters. Minimum length = 1
hits	<Double>	Read-only	Number of hits.
undefhits	<Double>	Read-only	Number of policy UNDEF hits.
__count	<Double>	Read-only	count parameter

Operations

(click to see Properties)

ADD| DELETE| UPDATE| UNSET| RENAME| GET (ALL)| GET| COUNT

Some options that you can use for each operations:

- **Getting warnings in response:**NITRO allows you to get warnings in an operation by specifying the "warning" query parameter as "yes". For example, to get warnings while connecting to the Citrix ADC appliance, the URL is as follows:

  http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/login?warning=yes

  If any, the warnings are displayed in the response payload with the HTTP code "209 X-NITRO-WARNING".

- **Authenticated access for individual NITRO operations:**NITRO allows you to logon to the Citrix ADC appliance to perform individual operations. You can use this option instead of creating a NITRO session (using the login object) and then using that session to perform all operations,

  To do this, you must specify the username and password in the request header of the NITRO request as follows:

  X-NITRO-USER:<username>

  X-NITRO-PASS:<password>

  Note: In such cases, make sure that the request header DOES not include the following:

  Cookie:NITRO_AUTH_TOKEN=<tokenvalue>

`*Note:`*

Mandatory parameters are marked in redand placeholder content is marked in <green>.

add

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/videooptimizationpacingpolicy

HTTP Method:POST

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Content-Type:application/json

Request Payload:

```
{"videooptimizationpacingpolicy":{

<b>"name":<String_value>,

</b><b>"rule":<String_value>,

</b><b>"action":<String_value>,

</b>"undefaction":<String_value>,

"comment":<String_value>,

"logaction":<String_value>

}}
```

Response:

HTTP Status Code on Success: 201 Created HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error

delete

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/videooptimizationpacingpolicy/name_value<String>

HTTP Method:DELETE

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue>

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error

update

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/videooptimizationpacingpolicy

HTTP Method:PUT

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Content-Type:application/json

Request Payload:

```
{"videooptimizationpacingpolicy":{

<b>"name":<String_value>,

</b>"rule":<String_value>,

"action":<String_value>,

"undefaction":<String_value>,

"comment":<String_value>,

"logaction":<String_value>

}}
```

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error

unset

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/videooptimizationpacingpolicy?**action=unset**

HTTP Method:POST

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Content-Type:application/json

Request Payload:

```
{"videooptimizationpacingpolicy":{

<b>"name":<String_value>,

</b>"undefaction":true,

"comment":true,

"logaction":true

}}
```

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error

rename

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/videooptimizationpacingpolicy?**action=rename**

HTTP Method:POST

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Content-Type:application/json

Request Payload:

```
{"videooptimizationpacingpolicy":{

<b>"name":<String_value>,
```

```
</b><b>"newname":<String_value>

</b>}}
```

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error

get (all)

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/videooptimizationpacingpolicy

Query-parameters:

attrs

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/videooptimizationpacingpolicy?**attrs=property-name1,property-name2**

Use this query parameter to specify the resource details that you want to retrieve.

filter

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/videooptimizationpacingpolicy?**filter=property-name1:property-val1,property-name2:property-val2**

Use this query-parameter to get the filtered set of videooptimizationpacingpolicy resources configured on Citrix ADC. Filtering can be done on any of the properties of the resource.

view

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/videooptimizationpacingpolicy?**view=summary**

Note: By default, the retrieved results are displayed in detail view (?view=detail).

pagination

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/videooptimizationpacingpolicy?**pagesize=#no;pageno=#no**

Use this query-parameter to get the videooptimizationpacingpolicy resources in chunks.

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "videooptimizationpacingpolicy": [ {

"name":<String_value>,

"rule":<String_value>,

"action":<String_value>,

"undefaction":<String_value>,

"hits":<Double_value>,

"undefhits":<Double_value>,

"comment":<String_value>,

"logaction":<String_value>

}]}
```

get

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/videooptimizationpacingpolicy/name_value<String>

Query-parameters:

attrs

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/videooptimizationpacingpolicy/name_value<String>?**attrs=property-name1,property-name2**

Use this query parameter to specify the resource details that you want to retrieve.

view

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/videooptimizationpacingpolicy/name_value<String>?**view=summary**

Note: By default, the retrieved results are displayed in detail view (?view=detail).

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "videooptimizationpacingpolicy": [ {

"name":<String_value>,

"rule":<String_value>,
```

```
"action":<String_value>,

"undefaction":<String_value>,

"hits":<Double_value>,

"undefhits":<Double_value>,

"comment":<String_value>,

"logaction":<String_value>

}]}
```

count

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/videooptimizationpacingpolicy?**count=yes**

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "videooptimizationpacingpolicy": [ { "__count": "#no"} ] }
```

videooptimizationpacingpolicy_binding

Binding object which returns the resources bound to videooptimizationpacingpolicy.

Properties

(click to see Operations)

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
name	<String>	Read-write	Name of the videooptimization pacing policy for which to display settings.Must provide policy name.
videooptimizationpacingpolicy_videooptimizationglobalpacing_binding	<videooptimizationpacingpolicy_videooptimizationglobalpacing_binding[]>	Read-only	videooptimizationglobalpacing that can be bound to videooptimizationpacingpolicy.
videooptimizationpacingpolicy_lbvserver_binding	<videooptimizationpacingpolicy_lbvserver_binding[]>	Read-only	lbvserver that can be bound to videooptimizationpacingpolicy.

Operations

(click to see Properties)

GET| GET (ALL)

Some options that you can use for each operations:

- **Getting warnings in response:**NITRO allows you to get warnings in an operation by specifying the "warning" query parameter as "yes". For example, to get warnings while connecting to the Citrix ADC appliance, the URL is as follows:

  http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/login?warning=yes

  If any, the warnings are displayed in the response payload with the HTTP code "209 X-NITRO-WARNING".

- **Authenticated access for individual NITRO operations:**NITRO allows you to logon to the Citrix ADC appliance to perform individual operations. You can use this option instead of creating a NITRO session (using the login object) and then using that session to perform all operations,

  To do this, you must specify the username and password in the request header of the NITRO request as follows:

  X-NITRO-USER:<username>

  X-NITRO-PASS:<password>

  Note: In such cases, make sure that the request header DOES not include the following:

  Cookie:NITRO_AUTH_TOKEN=<tokenvalue>

`*Note:`*

Mandatory parameters are marked in redand placeholder content is marked in <green>.

get

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/videooptimizationpacingpolicy_binding/name_value<String>

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "videooptimizationpacingpolicy_binding": [ {
```

```
"name":<String_value>,

"videooptimizationpacingpolicy_lbvserver_binding":<videooptimizationpacing
policy_lbvserver_binding[]_value>,

"videooptimizationpacingpolicy_videooptimizationglobalpacing_binding":<vid
eooptimizationpacingpolicy_videooptimizationglobalpacing_binding[]_value>

}]}
```

get (all)

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/videooptimizationpacingpolicy_binding

Query-parameters:

bulkbindings

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/videooptimizationpacingpolicy_binding?**bulkbindings=yes**

NITRO allows you to fetch bindings in bulk.

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "videooptimizationpacingpolicy_binding": [ {
```

```
"name":<String_value>,

"videooptimizationpacingpolicy_lbvserver_binding":<videooptimizationpacing
policy_lbvserver_binding[]_value>,

"videooptimizationpacingpolicy_videooptimizationglobalpacing_binding":<vid
eooptimizationpacingpolicy_videooptimizationglobalpacing_binding[]_value>

}]}
```

videooptimizationpacingpolicy_lbvserver_binding

Binding object showing the lbvserver that can be bound to videooptimizationpacingpolicy.

Properties

(click to see Operations)

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
boundto	<String>	Read-write	Location where policy is bound.
name	<String>	Read-write	Name of the videooptimization pacing policy for which to display settings.Must provide policy name.
priority	<Double>	Read-only	Specifies the priority of the policy.
labelname	<String>	Read-only	Name of the label to invoke if the current policy rule evaluates to TRUE.
gotopriorityexpression	<String>	Read-only	Expression specifying the priority of the next policy which will get

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
			evaluated if the current policy rule evaluates to TRUE.
labeltype	<String>	Read-only	Type of policy label invocation. Possible values = vserver, policylabel
activepolicy	<Integer>	Read-only	Indicates whether policy is bound or not.
__count	<Double>	Read-only	count parameter

Operations

(click to see Properties)

GET| GET (ALL)| COUNT

Some options that you can use for each operations:

- **Getting warnings in response:**NITRO allows you to get warnings in an operation by specifying the "warning" query parameter as "yes". For example, to get warnings while connecting to the Citrix ADC appliance, the URL is as follows:

  http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/login?warning=yes

  If any, the warnings are displayed in the response payload with the HTTP code "209 X-NITRO-WARNING".

- **Authenticated access for individual NITRO operations:**NITRO allows you to logon to the Citrix ADC appliance to perform individual operations. You can use this option instead of creating a NITRO session (using the login object) and then using that session to perform all operations,

  To do this, you must specify the username and password in the request header of the NITRO request as follows:

  X-NITRO-USER:<username>

X-NITRO-PASS:<password>

Note: In such cases, make sure that the request header DOES not include the following:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue>

`*Note:`*

Mandatory parameters are marked in redand placeholder content is marked in <green>.

get

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/videooptimizationpacingpolicy_lbvserver_binding/name_value<String>

Query-parameters:

filter

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/videooptimizationpacingpolicy_lbvserver_binding/name_value<String>?**filter=property-name1:property-value1,property-name2:property-value2**

Use this query-parameter to get the filtered set of videooptimizationpacingpolicy_lbvserver_binding resources configured on Citrix ADC. Filtering can be done on any of the properties of the resource.

pagination

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/videooptimizationpacingpolicy_lbvserver_binding/name_value<String>?**pagesize=#no;pageno=#no**

Use this query-parameter to get the videooptimizationpacingpolicy_lbvserver_binding resources in chunks.

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "videooptimizationpacingpolicy_lbvserver_binding": [ {

"boundto":<String_value>,

"name":<String_value>,

"priority":<Double_value>,

"labelname":<String_value>,

"gotopriorityexpression":<String_value>,

"labeltype":<String_value>,

"activepolicy":<Integer_value>

}]}
```

get (all)

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/videooptimizationpacingpolicy_lbvserver_binding

Query-parameters:

bulkbindings

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/videooptimizationpacingpolicy_lbvserver_binding?**bulkbindings=yes**

NITRO allows you to fetch bindings in bulk.

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "videooptimizationpacingpolicy_lbvserver_binding": [ {

"boundto":<String_value>,

"name":<String_value>,

"priority":<Double_value>,

"labelname":<String_value>,

"gotopriorityexpression":<String_value>,

"labeltype":<String_value>,

"activepolicy":<Integer_value>

}]}
```

count

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/videooptimizationpacingpolicy_lbvserver_binding/name_value<String>?**count=yes**

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{"videooptimizationpacingpolicy_lbvserver_binding": [ { "__count":
"#no"} ] }
```

videooptimizationpacingpolicylabel

Configuration for videooptimization pacing policy label resource.

Properties

(click to see Operations)

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
labelname	<String>	Read-write	Name for the Video optimization pacing policy label. Must begin with a letter, number, or the underscore character (_), and must contain only letters, numbers, and the hyphen (-), period (.) hash (#), space ( ), at (@), equals (=), colon (:), and underscore

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
			characters. Cannot be changed after the videooptimization pacing policy label is added. The following requirement applies only to the Citrix ADC CLI: If the name includes one or more spaces, enclose the name in double or single quotation marks (for example, "my videooptimization pacing policy label" or my videooptimization pacing policy label').
policylabeltype	<String>	Read-write	Type of responses sent by the policies bound to this policy label. Types are: * HTTP - HTTP responses. * OTHERTCP - NON-HTTP TCP responses.

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
			Default value: NS_PLTMAP_RSP_REQ Possible values = videoopt_req, videoopt_res
comment	<String>	Read-write	Any comments to preserve information about this videooptimization pacing policy label.
newname	<String>	Read-write	New name for the videooptimization pacing policy label. Must begin with a letter, number, or the underscore character (_), and must contain only letters, numbers, and the hyphen (-), period (.) hash (#), space ( ), at (@), equals (=), colon (:), and underscore characters. Minimum length = 1
numpol	<Double>	Read-only	number of polices bound to label.
hits	<Double>	Read-only	Number of times policy label was invoked.

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
priority	<Double>	Read-only	Specifies the priority of the policy.
gotopriorityexpression	<String>	Read-only	Expression specifying the priority of the next policy which will get evaluated if the current policy rule evaluates to TRUE.
labeltype	<String>	Read-only	Type of policy label to invoke. Available settings function as follows: * vserver - Invoke an unnamed policy label associated with a virtual server. * policylabel - Invoke a user-defined policy label. Possible values = vserver, policylabel
invoke_labelname	<String>	Read-only	* If labelType is policylabel, name of the policy label to invoke. * If labelType is reqvserver or resvserver, name of the virtual server.
__count	<Double>	Read-only	count parameter

Operations

(click to see Properties)

ADD| DELETE| RENAME| GET (ALL)| GET| COUNT

Some options that you can use for each operations:

- **Getting warnings in response:**NITRO allows you to get warnings in an operation by specifying the "warning" query parameter as "yes". For example, to get warnings while connecting to the Citrix ADC appliance, the URL is as follows:

  http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/login?warning=yes

  If any, the warnings are displayed in the response payload with the HTTP code "209 X-NITRO-WARNING".

- **Authenticated access for individual NITRO operations:**NITRO allows you to logon to the Citrix ADC appliance to perform individual operations. You can use this option instead of creating a NITRO session (using the login object) and then using that session to perform all operations,

  To do this, you must specify the username and password in the request header of the NITRO request as follows:

  X-NITRO-USER:<username>

  X-NITRO-PASS:<password>

  Note: In such cases, make sure that the request header DOES not include the following:

  Cookie:NITRO_AUTH_TOKEN=<tokenvalue>

`*Note:`*

Mandatory parameters are marked in redand placeholder content is marked in <green>.

add

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/videooptimizationpacingpolicylabel

HTTP Method:POST

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Content-Type:application/json

Request Payload:

```
{"videooptimizationpacingpolicylabel":{

<b>"labelname":<String_value>,

</b>"policylabeltype":<String_value>,

"comment":<String_value>

}}
```

Response:

HTTP Status Code on Success: 201 Created HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error

delete

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/videooptimizationpacingpolicylabel/labelname_value<String>

HTTP Method:DELETE

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue>

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error

rename

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/videooptimizationpacingpolicylabel?**action=rename**

HTTP Method:POST

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Content-Type:application/json

Request Payload:

```
{"videooptimizationpacingpolicylabel":{

<b>"labelname":<String_value>,

</b><b>"newname":<String_value>

</b>}}
```

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error

get (all)

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/videooptimizationpacingpolicylabel

Query-parameters:

attrs

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/videooptimizationpacingpolicylabel?**attrs=property-name1,property-name2**

Use this query parameter to specify the resource details that you want to retrieve.

filter

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/videooptimizationpacingpolicylabel?**filter=property-name1:property-val1,property-name2:property-val2**

Use this query-parameter to get the filtered set of videooptimizationpacingpolicylabel resources configured on Citrix ADC. Filtering can be done on any of the properties of the resource.

view

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/videooptimizationpacingpolicylabel?**view=summary**

Note: By default, the retrieved results are displayed in detail view (?view=detail).

pagination

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/videooptimizationpacingpolicylabel?**pagesize=#no;pageno=#no**

Use this query-parameter to get the videooptimizationpacingpolicylabel resources in chunks.

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "videooptimizationpacingpolicylabel": [ {

"labelname":<String_value>,

"policylabeltype":<String_value>,

"numpol":<Double_value>,

"hits":<Double_value>,

"priority":<Double_value>,
```

```
"gotopriorityexpression":<String_value>,

"labeltype":<String_value>,

"invoke_labelname":<String_value>,

"comment":<String_value>

}]}
```

get

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/videooptimizationpacingpolicylabel/labelname_value<String>

Query-parameters:

attrs

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/videooptimizationpacingpolicylabel/labelname_value<String>?**attrs=property-name1,property-name2**

Use this query parameter to specify the resource details that you want to retrieve.

view

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/videooptimizationpacingpolicylabel/labelname_value<String>?**view=summary**

Note: By default, the retrieved results are displayed in detail view (?view=detail).

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "videooptimizationpacingpolicylabel": [ {

"labelname":<String_value>,

"policylabeltype":<String_value>,

"numpol":<Double_value>,

"hits":<Double_value>,

"priority":<Double_value>,

"gotopriorityexpression":<String_value>,

"labeltype":<String_value>,

"invoke_labelname":<String_value>,

"comment":<String_value>

}]}
```

count

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/videooptimizationpacingpolicylabel?**count=yes**

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "videooptimizationpacingpolicylabel": [ { "__count": "#no"} ] }
```

videooptimizationpacingpolicylabel_binding

Binding object which returns the resources bound to videooptimizationpacingpolicylabel.

Properties

(click to see Operations)

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
labelname	<String>	Read-write	Name of the videooptimization pacing policy label.
videooptimizationpacingpolicylabel_videooptimizationpacingpolicy_binding	<videooptimizationpacingpolicylabel_videooptimizationpacingpolicy_binding[]>	Read-only	videooptimizationpacingpolicy that can be bound to videooptimizationpacingpolicylabel.
videooptimizationpacingpolicylabel_policybinding_binding	<videooptimizationpacingpolicylabel_policybinding_binding[]>	Read-only	policybinding that can be bound to videooptimizationpacingpolicylabel.

Operations

(click to see Properties)

GET| GET (ALL)

Some options that you can use for each operations:

- **Getting warnings in response:**NITRO allows you to get warnings in an operation by specifying the "warning" query parameter as "yes". For example, to get warnings while connecting to the Citrix ADC appliance, the URL is as follows:

  http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/login?warning=yes

  If any, the warnings are displayed in the response payload with the HTTP code "209 X-NITRO-WARNING".

- **Authenticated access for individual NITRO operations:**NITRO allows you to logon to the Citrix ADC appliance to perform individual operations. You can use this option instead of creating a NITRO session (using the login object) and then using that session to perform all operations,

  To do this, you must specify the username and password in the request header of the NITRO request as follows:

  X-NITRO-USER:<username>

  X-NITRO-PASS:<password>

  Note: In such cases, make sure that the request header DOES not include the following:

  Cookie:NITRO_AUTH_TOKEN=<tokenvalue>

`*Note:`*

Mandatory parameters are marked in redand placeholder content is marked in <green>.

get

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/videooptimizationpacingpolicylabel_binding/labelname_value<String>

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "videooptimizationpacingpolicylabel_binding": [ {

"labelname":<String_value>,

"videooptimizationpacingpolicylabel_policybinding_binding":<videooptimizat
ionpacingpolicylabel_policybinding_binding[]_value>,

"videooptimizationpacingpolicylabel_videooptimizationpacingpolicy_binding"
:<videooptimizationpacingpolicylabel_videooptimizationpacingpolicy_binding
[]_value>

}]}
```

get (all)

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/videooptimizationpacingpolicylabel_binding

Query-parameters:

bulkbindings

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/videooptimizationpacingpolicylabel_binding?**bulkbindings=yes**

NITRO allows you to fetch bindings in bulk.

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "videooptimizationpacingpolicylabel_binding": [ {

"labelname":<String_value>,

"videooptimizationpacingpolicylabel_policybinding_binding":<videooptimizat
ionpacingpolicylabel_policybinding_binding[]_value>,

"videooptimizationpacingpolicylabel_videooptimizationpacingpolicy_binding"
:<videooptimizationpacingpolicylabel_videooptimizationpacingpolicy_binding
[]_value>

}]}
```

videooptimizationpacingpolicylabel_policybinding_binding

Binding object showing the policybinding that can be bound to videooptimizationpacingpolicylabel.

Properties

(click to see Operations)

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
priority	<Double>	Read-write	Specifies the priority of the policy.
policyname	<String>	Read-write	Name of the videooptimization policy.

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
labelname	<String>	Read-write	Name of the videooptimization pacing policy label to which to bind the policy.
invoke_labelname	<String>	Read-write	* If labelType is policylabel, name of the policy label to invoke. * If labelType is reqvserver or resvserver, name of the virtual server.
gotopriorityexpression	<String>	Read-write	Expression specifying the priority of the next policy which will get evaluated if the current policy rule evaluates to TRUE.
invoke	<Boolean>	Read-write	If the current policy evaluates to TRUE, terminate evaluation of policies bound to the current policy label and evaluate the specified policy label.
labeltype	<String>	Read-write	Type of policy label to invoke. Available settings function as follows: * vserver - Invoke an unnamed policy label associated with a virtual server. * policylabel - Invoke a

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
			user-defined policy label. Possible values = vserver, policylabel
__count	<Double>	Read-write	count parameter

Operations

(click to see Properties)

GET| GET (ALL)| COUNT

Some options that you can use for each operations:

- **Getting warnings in response:**NITRO allows you to get warnings in an operation by specifying the "warning" query parameter as "yes". For example, to get warnings while connecting to the Citrix ADC appliance, the URL is as follows:

  http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/login?warning=yes

  If any, the warnings are displayed in the response payload with the HTTP code "209 X-NITRO-WARNING".

- **Authenticated access for individual NITRO operations:**NITRO allows you to logon to the Citrix ADC appliance to perform individual operations. You can use this option instead of creating a NITRO session (using the login object) and then using that session to perform all operations,

  To do this, you must specify the username and password in the request header of the NITRO request as follows:

  X-NITRO-USER:<username>

  X-NITRO-PASS:<password>

  Note: In such cases, make sure that the request header DOES not include the following:

  Cookie:NITRO_AUTH_TOKEN=<tokenvalue>

`*Note:` *

Mandatory parameters are marked in redand placeholder content is marked in <green>.

get

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/videooptimizationpacingpolicylabel_policybinding_binding/labelname_value<String>

Query-parameters:

filter

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/videooptimizationpacingpolicylabel_policybinding_binding/labelname_value<String>?**filter=property-name1:property-value1,property-name2:property-value2**

Use this query-parameter to get the filtered set of videooptimizationpacingpolicylabel_policybinding_binding resources configured on Citrix ADC. Filtering can be done on any of the properties of the resource.

pagination

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/videooptimizationpacingpolicylabel_policybinding_binding/labelname_value<String>?**pagesize=#no;pageno=#no**

Use this query-parameter to get the videooptimizationpacingpolicylabel_policybinding_binding resources in chunks.

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "videooptimizationpacingpolicylabel_policybinding_binding": [ {

"priority":<Double_value>,

"policyname":<String_value>,

"labelname":<String_value>,

"invoke_labelname":<String_value>,

"gotopriorityexpression":<String_value>,

"invoke":<Boolean_value>,

"labeltype":<String_value>

}]}
```

get (all)

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/videooptimizationpacingpolicylabel_policybinding_binding

Query-parameters:

bulkbindings

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/videooptimizationpacingpolicylabel_policybinding_binding?**bulkbindings=yes**

NITRO allows you to fetch bindings in bulk.

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "videooptimizationpacingpolicylabel_policybinding_binding": [ {

"priority":<Double_value>,

"policyname":<String_value>,

"labelname":<String_value>,

"invoke_labelname":<String_value>,

"gotopriorityexpression":<String_value>,

"invoke":<Boolean_value>,

"labeltype":<String_value>

}]}
```

count

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/videooptimizationpacingpolicylabel_policybinding_binding/labelname_value<String>?**count=yes**

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{"videooptimizationpacingpolicylabel_policybinding_binding":
[ { "__count": "#no"} ] }
```

videooptimizationpacingpolicy_videooptimizationglobalpacing_binding

Binding object showing the videooptimizationglobalpacing that can be bound to videooptimizationpacingpolicy.

Properties

(click to see Operations)

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
boundto	<String>	Read-write	Location where policy is bound.
name	<String>	Read-write	Name of the videooptimization pacing policy for which to display settings.Must provide policy name.
priority	<Double>	Read-only	Specifies the priority of the policy.
labelname	<String>	Read-only	Name of the label to invoke if the current policy rule evaluates to TRUE.

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
gotopriorityexpression	<String>	Read-only	Expression specifying the priority of the next policy which will get evaluated if the current policy rule evaluates to TRUE.
labeltype	<String>	Read-only	Type of policy label invocation. Possible values = vserver, policylabel
activepolicy	<Integer>	Read-only	Indicates whether policy is bound or not.
__count	<Double>	Read-only	count parameter

Operations

(click to see Properties)

GET| GET (ALL)| COUNT

Some options that you can use for each operations:

- **Getting warnings in response:**NITRO allows you to get warnings in an operation by specifying the "warning" query parameter as "yes". For example, to get warnings while connecting to the Citrix ADC appliance, the URL is as follows:

  http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/login?warning=yes

  If any, the warnings are displayed in the response payload with the HTTP code "209 X-NITRO-WARNING".

- **Authenticated access for individual NITRO operations:**NITRO allows you to logon to the Citrix ADC appliance to perform individual operations. You can use this option instead of creating a NITRO session (using the login object) and then using that session to perform all operations,

To do this, you must specify the username and password in the request header of the NITRO request as follows:

X-NITRO-USER:<username>

X-NITRO-PASS:<password>

Note: In such cases, make sure that the request header DOES not include the following:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue>

`*Note:`*

Mandatory parameters are marked in redand placeholder content is marked in <green>.

get

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/videooptimizationpacingpolicy_videooptimizationglobalpacing_binding/name_value<String>

Query-parameters:

filter

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/videooptimizationpacingpolicy_videooptimizationglobalpacing_binding/name_value<String>?**filter=property-name1:property-value1,property-name2:property-value2**

Use this query-parameter to get the filtered set of videooptimizationpacingpolicy_videooptimizationglobalpacing_binding resources configured on Citrix ADC. Filtering can be done on any of the properties of the resource.

pagination

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/videooptimizationpacingpolicy_videooptimizationglobalpacing_binding/name_value<String>?**pagesize=#no;pageno=#no**

Use this query-parameter to get the videooptimizationpacingpolicy_videooptimizationglobalpacing_binding resources in chunks.

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "videooptimizationpacingpolicy_videooptimizationglobalpacing_binding":
[ {

"boundto":<String_value>,

"name":<String_value>,

"priority":<Double_value>,

"labelname":<String_value>,

"gotopriorityexpression":<String_value>,

"labeltype":<String_value>,

"activepolicy":<Integer_value>

}]}
```

get (all)

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/videooptimizationpacingpolicy_videooptimizationglobalpacing_binding

Query-parameters:

bulkbindings

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/videooptimizationpacingpolicy_videooptimizationglobalpacing_binding?**bulkbindings=yes**

NITRO allows you to fetch bindings in bulk.

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "videooptimizationpacingpolicy_videooptimizationglobalpacing_binding":
[ {

"boundto":<String_value>,

"name":<String_value>,

"priority":<Double_value>,

"labelname":<String_value>,

"gotopriorityexpression":<String_value>,

"labeltype":<String_value>,

"activepolicy":<Integer_value>

}]}
```

count

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/videooptimizationpacingpolicy_videooptimizationglobalpacing_binding/name_value<String>?**count=yes**

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{"videooptimizationpacingpolicy_videooptimizationglobalpacing_binding":
[ { "__count": "#no"} ] }
```

Web interface

WebInterface

Web Interface configuration.

NAME	DESCRIPTION
wipackage	Configuration for Web Interface
wisite	Configuration for WI site
wisite_accessmethod_binding	Binding object showing the accessmethod that can be bound to wisite

NAME	DESCRIPTION
wisite_binding	Binding object showing the resources that can be bound to wisite
wisite_farmname_binding	Binding object showing the farmname that can be bound to wisite
wisite_translationinternalip_binding	Binding object showing the translationinternalip that can be bound to wisite

wipackage

Configuration for Web Interface resource.

Properties

(click to see Operations)

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
jre	<String>	Read-write	Complete path to the JRE tar file. You can use OpenJDK7 package for FreeBSD 8.x/amd64.The Java package can be downloaded from http://ftp-archive.freebsd.org/pub/FreeBSD-Archive/old-releases/amd64/amd64/8.4-RELEASE/packages/java/openjdk-7.17.02_2.tbz.

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
			Default value: "file://tmp/diablo-jdk-freebsd6.amd64.1.6.0.07.02.tbz" Minimum length = 1 Maximum length = 255
wi	<String>	Read-write	Complete path to the Web Interface tar file for installing the Web Interface on the Citrix ADC. This file includes Apache Tomcat Web server. The file name has the following format: nswi-<version number>.tgz (for example, nswi-1.5.tgz). Default value: "http://citrix.com/downloads/nswi-1.7.tgz" Minimum length = 1 Maximum length = 255
maxsites	<String>	Read-write	Maximum number of Web Interface sites that can be created on the Citrix ADC; changes the amount of RAM reserved for Web Interface usage; changing its

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
			value results in restart of Tomcat server and invalidates any existing Web Interface sessions. Possible values = 3, 25, 50, 100, 200, 500

Operations

(click to see Properties)

INSTALL| GET (ALL)

Some options that you can use for each operations:

- **Getting warnings in response:**NITRO allows you to get warnings in an operation by specifying the "warning" query parameter as "yes". For example, to get warnings while connecting to the Citrix ADC appliance, the URL is as follows:

  http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/login?warning=yes

  If any, the warnings are displayed in the response payload with the HTTP code "209 X-NITRO-WARNING".
- **Authenticated access for individual NITRO operations:**NITRO allows you to logon to the Citrix ADC appliance to perform individual operations. You can use this option instead of creating a NITRO session (using the login object) and then using that session to perform all operations,

  To do this, you must specify the username and password in the request header of the NITRO request as follows:

  X-NITRO-USER:<username>

  X-NITRO-PASS:<password>

  Note: In such cases, make sure that the request header DOES not include the following:

  Cookie:NITRO_AUTH_TOKEN=<tokenvalue>

`*Note:` *

Mandatory parameters are marked in redand placeholder content is marked in <green>.

Install

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/wipackage

HTTP Method:POST

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Content-Type:application/json

Request Payload:

```
{"wipackage":{

"jre":<String_value>,

"wi":<String_value>,

"maxsites":<String_value>

}}
```

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error

get (all)

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/wipackage

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "wipackage": [ {

"jre":<String_value>,

"wi":<String_value>,

"maxsites":<String_value>

}]}
```

wisite

Configuration for WI site resource.

Properties

(click to see Operations)

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
sitepath	<String>	Read-write	Path to the Web Interface site being created on the Citrix ADC. Minimum length = 1 Maximum length = 250
agurl	<String>	Read-write	Call back URL of the Gateway.

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
			Minimum length = 1 Maximum length = 255
staurl	<String>	Read-write	URL of the Secure Ticket Authority (STA) server. Minimum length = 1 Maximum length = 255
secondstaurl	<String>	Read-write	URL of the second Secure Ticket Authority (STA) server. Minimum length = 1 Maximum length = 255
sessionreliability	<String>	Read-write	Enable session reliability through Access Gateway. Default value: OFF Possible values = ON, OFF
usetwotickets	<String>	Read-write	Request tickets issued by two separate Secure Ticket Authorities (STA) when a resource is accessed. Default value: OFF Possible values = ON, OFF

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
authenticationpoint	<String>	Read-write	Authentication point for the Web Interface site. Possible values = WebInterface, AccessGateway
agauthenticationmethod	<String>	Read-write	Method for authenticating a Web Interface site if you have specified Web Interface as the authentication point. Available settings function as follows: * Explicit - Users must provide a user name and password to log on to the Web Interface. * Anonymous - Users can log on to the Web Interface without providing a user name and password. They have access to resources published for anonymous users. Possible values = Explicit, SmartCard
wiauthenticationmethods	<String[]>	Read-write	The method of authentication to be

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
			used at Web Interface. Default value: Explicit Possible values = Explicit, Anonymous
defaultcustomtextlocale	<String>	Read-write	Default language for the Web Interface site. Default value: English Possible values = German, English, Spanish, French, Japanese, Korean, Russian, Chinese_simplified, Chinese_traditional
websessiontimeout	<Double>	Read-write	Time-out, in minutes, for idle Web Interface browser sessions. If a client's session is idle for a time that exceeds the time-out value, the Citrix ADC terminates the connection. Default value: 20 Minimum value = 1 Maximum value = 1440

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
defaultaccessmethod	<String>	Read-write	Default access method for clients accessing the Web Interface site. Note: Before you configure an access method based on the client IP address, you must enable USIP mode on the Web Interface service to make the client's IP address available with the Web Interface. Depending on whether the Web Interface site is configured to use an HTTP or HTTPS virtual server or to use access gateway, you can send clients or access gateway the IP address, or the alternate address, of a XenApp or XenDesktop server. Or, you can send the IP address translated from a mapping entry, which defines

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
			mapping of an internal address and port to an external address and port. Note: In the Citrix ADC command line, mapping entries can be created by using the bind wi site command. Possible values = Direct, Alternate, Translated, GatewayDirect, GatewayAlternate, GatewayTranslated
logintitle	<String>	Read-write	A custom login page title for the Web Interface site. Default value: "Welcome to Web Interface on Citrix ADC" Minimum length = 1 Maximum length = 255
appwelcomemessage	<String>	Read-write	Specifies localized text to appear at the top of the main content area of the Applications screen. LanguageCode is

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
			en, de, es, fr, ja, or any other supported language identifier. Minimum length = 1 Maximum length = 255
welcomemessage	<String>	Read-write	Localized welcome message that appears on the welcome area of the login screen. Minimum length = 1 Maximum length = 255
footertext	<String>	Read-write	Localized text that appears in the footer area of all pages. Minimum length = 1 Maximum length = 255
loginsysmessage	<String>	Read-write	Localized text that appears at the bottom of the main content area of the login screen. Minimum length = 1 Maximum length = 255
preloginbutton	<String>	Read-write	Localized text that appears as the name of the pre-login message confirmation button.

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
			Minimum length = 1 Maximum length = 255
preloginmessage	<String>	Read-write	Localized text that appears on the pre-login message page. Minimum length = 1 Maximum length = 2048
prelogintitle	<String>	Read-write	Localized text that appears as the title of the pre-login message page. Minimum length = 1 Maximum length = 255
domainselection	<String>	Read-write	Domain names listed on the login screen for explicit authentication. Minimum length = 1 Maximum length = 255
sitetype	<String>	Read-write	Type of access to the Web Interface site. Available settings function as follows: * XenApp/ XenDesktop web site - Configures the Web Interface site

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
			for access by a web browser. * XenApp/ XenDesktop services site - Configures the Web Interface site for access by the XenApp plug-in. Default value: XenAppWeb Possible values = XenAppWeb, XenAppServices
userinterfacebrandin g	<String>	Read-write	Specifies whether the site is focused towards users accessing applications or desktops. Setting the parameter to Desktops changes the functionality of the site to improve the experience for XenDesktop users. Citrix recommends using this setting for any deployment that includes XenDesktop. Default value: Applications

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
			Possible values = Desktops, Applications
publishedresourcetype	<String>	Read-write	Method for accessing the published XenApp and XenDesktop resources. Available settings function as follows: * Online - Allows applications to be launched on the XenApp and XenDesktop servers. * Offline - Allows streaming of applications to the client. * DualMode - Allows both online and offline modes. Default value: Online Possible values = Online, Offline, DualMode
kioskmode	<String>	Read-write	User settings do not persist from one session to another.

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
			Default value: OFF Possible values = ON, OFF
showsearch	<String>	Read-write	Enables search option on XenApp websites. Default value: OFF Possible values = ON, OFF
showrefresh	<String>	Read-write	Provides the Refresh button on the applications screen. Default value: OFF Possible values = ON, OFF
wiuserinterfacemodes	<String>	Read-write	Appearance of the login screen. * Simple - Only the login fields for the selected authentication method are displayed. * Advanced - Displays the navigation bar, which provides access to the pre-login messages and preferences screens.

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
			Default value: SIMPLE Possible values = SIMPLE, ADVANCED
userinterfacelayouts	<String>	Read-write	Specifies whether or not to use the compact user interface. Default value: AUTO Possible values = AUTO, NORMAL, COMPACT
restrictdomains	<String>	Read-write	The RestrictDomains setting is used to enable/disable domain restrictions. If domain restriction is enabled, the LoginDomains list is used for validating the login domain. It is applied to all the authentication methods except Anonymous for XenApp Web and XenApp Services sites. Default value: OFF Possible values = ON, OFF

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
logindomains	<String>	Read-write	[List of NetBIOS domain names], Domain names to use for access restriction. Only takes effect when used in conjunction with the RestrictDomains setting. Minimum length = 1 Maximum length = 255
hidedomainfield	<String>	Read-write	The HideDomainField setting is used to control whether the domain field is displayed on the logon screen. Default value: OFF Possible values = ON, OFF
agcallbackurl	<String>	Read-write	Callback AGURL to which Web Interface contacts. . Minimum length = 1 Maximum length = 255
__count	<Double>	Read-only	count parameter

Operations

(click to see Properties)

ADD| DELETE| UPDATE| UNSET| GET (ALL)| GET| COUNT

Some options that you can use for each operations:

- **Getting warnings in response:**NITRO allows you to get warnings in an operation by specifying the "warning" query parameter as "yes". For example, to get warnings while connecting to the Citrix ADC appliance, the URL is as follows:

  http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/login?warning=yes

  If any, the warnings are displayed in the response payload with the HTTP code "209 X-NITRO-WARNING".

- **Authenticated access for individual NITRO operations:**NITRO allows you to logon to the Citrix ADC appliance to perform individual operations. You can use this option instead of creating a NITRO session (using the login object) and then using that session to perform all operations,

  To do this, you must specify the username and password in the request header of the NITRO request as follows:

  X-NITRO-USER:<username>

  X-NITRO-PASS:<password>

  Note: In such cases, make sure that the request header DOES not include the following:

  Cookie:NITRO_AUTH_TOKEN=<tokenvalue>

`*Note:`*

Mandatory parameters are marked in redand placeholder content is marked in <green>.

add

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/wisite

HTTP Method:POST

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Content-Type:application/json

Request Payload:

```
{"wisite":{

<b>"sitepath":<String_value>,

</b>"agurl":<String_value>,

"staurl":<String_value>,

"secondstaurl":<String_value>,

"sessionreliability":<String_value>,

"usetwotickets":<String_value>,

"authenticationpoint":<String_value>,

"agauthenticationmethod":<String_value>,

"wiauthenticationmethods":<String[]_value>,

"defaultcustomtextlocale":<String_value>,

"websessiontimeout":<Double_value>,

"defaultaccessmethod":<String_value>,

"logintitle":<String_value>,

"appwelcomemessage":<String_value>,

"welcomemessage":<String_value>,

"footertext":<String_value>,

"loginsysmessage":<String_value>,

"preloginbutton":<String_value>,

"preloginmessage":<String_value>,
```

```
"prelogintitle":<String_value>,

"domainselection":<String_value>,

"sitetype":<String_value>,

"userinterfacebranding":<String_value>,

"publishedresourcetype":<String_value>,

"kioskmode":<String_value>,

"showsearch":<String_value>,

"showrefresh":<String_value>,

"wiuserinterfacemodes":<String_value>,

"userinterfacelayouts":<String_value>,

"restrictdomains":<String_value>,

"logindomains":<String_value>,

"hidedomainfield":<String_value>,

"agcallbackurl":<String_value>

}}
```

Response:

HTTP Status Code on Success: 201 Created HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error

delete

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/wisite/sitepath_value<String>

HTTP Method:DELETE

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue>

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error

update

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/wisite

HTTP Method:PUT

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Content-Type:application/json

Request Payload:

```
{"wisite":{

<b>"sitepath":<String_value>,

</b>"agurl":<String_value>,

"staurl":<String_value>,

"sessionreliability":<String_value>,

"usetwotickets":<String_value>,

"secondstaurl":<String_value>,

"wiauthenticationmethods":<String[]_value>,

"defaultaccessmethod":<String_value>,

"defaultcustomtextlocale":<String_value>,

"websessiontimeout":<Double_value>,
```

```
"logintitle":<String_value>,

"appwelcomemessage":<String_value>,

"welcomemessage":<String_value>,

"footertext":<String_value>,

"loginsysmessage":<String_value>,

"preloginbutton":<String_value>,

"preloginmessage":<String_value>,

"prelogintitle":<String_value>,

"domainselection":<String_value>,

"userinterfacebranding":<String_value>,

"authenticationpoint":<String_value>,

"agauthenticationmethod":<String_value>,

"publishedresourcetype":<String_value>,

"kioskmode":<String_value>,

"showsearch":<String_value>,

"showrefresh":<String_value>,

"wiuserinterfacemodes":<String_value>,

"userinterfacelayouts":<String_value>,

"restrictdomains":<String_value>,

"logindomains":<String_value>,

"hidedomainfield":<String_value>,

"agcallbackurl":<String_value>
```

```
}}
```

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error

unset

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/wisite?**action=unset**

HTTP Method:POST

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Content-Type:application/json

Request Payload:

```
{"wisite":{

<b>"sitepath":<String_value>,

</b>"appwelcomemessage":true,

"welcomemessage":true,

"footertext":true,

"loginsysmessage":true,

"preloginbutton":true,

"preloginmessage":true,

"prelogintitle":true,

"userinterfacebranding":true,
```

```
"logindomains":true

}}
```

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error

get (all)

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/wisite

Query-parameters:

attrs

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/wisite?**attrs=property-name1,property-name2**

Use this query parameter to specify the resource details that you want to retrieve.

filter

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/wisite?**filter=property-name1:property-val1,property-name2:property-val2**

Use this query-parameter to get the filtered set of wisite resources configured on Citrix ADC. Filtering can be done on any of the properties of the resource.

view

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/wisite?**view=summary**

Note: By default, the retrieved results are displayed in detail view (?view=detail).

pagination

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/wisite?**pagesize=#no;pageno=#no**

Use this query-parameter to get the wisite resources in chunks.

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "wisite": [ {

"sitepath":<String_value>,

"agurl":<String_value>,

"staurl":<String_value>,

"wiauthenticationmethods":<String[]_value>,

"logintitle":<String_value>,

"appwelcomemessage":<String_value>,

"welcomemessage":<String_value>,

"footertext":<String_value>,

"loginsysmessage":<String_value>,

"preloginbutton":<String_value>,

"preloginmessage":<String_value>,

"prelogintitle":<String_value>,

"domainselection":<String_value>,
```

```
"defaultcustomtextlocale":<String_value>,

"websessiontimeout":<Double_value>,

"sitetype":<String_value>,

"userinterfacebranding":<String_value>,

"showsearch":<String_value>,

"showrefresh":<String_value>,

"wiuserinterfacemodes":<String_value>,

"userinterfacelayouts":<String_value>,

"publishedresourcetype":<String_value>,

"defaultaccessmethod":<String_value>,

"agauthenticationmethod":<String_value>,

"sessionreliability":<String_value>,

"usetwotickets":<String_value>,

"secondstaurl":<String_value>,

"authenticationpoint":<String_value>,

"kioskmode":<String_value>,

"restrictdomains":<String_value>,

"logindomains":<String_value>,

"hidedomainfield":<String_value>,

"agcallbackurl":<String_value>

}]}
```

get

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/wisite/sitepath_value<String>

Query-parameters:

attrs

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/wisite/sitepath_value<String>?**attrs=property-name1,property-name2**

Use this query parameter to specify the resource details that you want to retrieve.

view

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/wisite/sitepath_value<String>?**view=summary**

Note: By default, the retrieved results are displayed in detail view (?view=detail).

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "wisite": [ {

"sitepath":<String_value>,

"agurl":<String_value>,
```

```
"staurl":<String_value>,

"wiauthenticationmethods":<String[]_value>,

"logintitle":<String_value>,

"appwelcomemessage":<String_value>,

"welcomemessage":<String_value>,

"footertext":<String_value>,

"loginsysmessage":<String_value>,

"preloginbutton":<String_value>,

"preloginmessage":<String_value>,

"prelogintitle":<String_value>,

"domainselection":<String_value>,

"defaultcustomtextlocale":<String_value>,

"websessiontimeout":<Double_value>,

"sitetype":<String_value>,

"userinterfacebranding":<String_value>,

"showsearch":<String_value>,

"showrefresh":<String_value>,

"wiuserinterfacemodes":<String_value>,

"userinterfacelayouts":<String_value>,

"publishedresourcetype":<String_value>,

"defaultaccessmethod":<String_value>,

"agauthenticationmethod":<String_value>,
```

```
"sessionreliability":<String_value>,

"usetwotickets":<String_value>,

"secondstaurl":<String_value>,

"authenticationpoint":<String_value>,

"kioskmode":<String_value>,

"restrictdomains":<String_value>,

"logindomains":<String_value>,

"hidedomainfield":<String_value>,

"agcallbackurl":<String_value>

}]}
```

count

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/wisite?**count=yes**

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "wisite": [ { "__count": "#no"} ] }
```

wisite_accessmethod_binding

Binding object showing the accessmethod that can be bound to wisite.

Properties

(click to see Operations)

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
clientipaddress	<String>	Read-write	IPv4 or network address of the client for which you want to associate an access method. Default value: 0
clientnetmask	<String>	Read-write	Subnet mask associated with the IPv4 or network address specified by the Client IP Address parameter. Default value: 0
accessmethod	<String>	Read-write	Secure access method to be applied to the IPv4 or network address of the client specified by the Client IP Address parameter. Depending on whether the Web Interface site is configured to use an HTTP or HTTPS

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
			virtual server or to use access gateway, you can send clients or access gateway the IP address, or the alternate address, of a XenApp or XenDesktop server. Or, you can send the IP address translated from a mapping entry, which defines mapping of an internal address and port to an external address and port. Possible values = Direct, Alternate, Translated, GatewayDirect, GatewayAlternate, GatewayTranslated
sitepath	<String>	Read-write	Path to the Web Interface site. Minimum length = 1 Maximum length = 250
__count	<Double>	Read-write	count parameter

Operations

(click to see Properties)

ADD:| DELETE:| GET| GET (ALL)| COUNT

Some options that you can use for each operations:

- **Getting warnings in response:**NITRO allows you to get warnings in an operation by specifying the "warning" query parameter as "yes". For example, to get warnings while connecting to the Citrix ADC appliance, the URL is as follows:

  http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/login?warning=yes

  If any, the warnings are displayed in the response payload with the HTTP code "209 X-NITRO-WARNING".

- **Authenticated access for individual NITRO operations:**NITRO allows you to logon to the Citrix ADC appliance to perform individual operations. You can use this option instead of creating a NITRO session (using the login object) and then using that session to perform all operations,

  To do this, you must specify the username and password in the request header of the NITRO request as follows:

  X-NITRO-USER:<username>

  X-NITRO-PASS:<password>

  Note: In such cases, make sure that the request header DOES not include the following:

  Cookie:NITRO_AUTH_TOKEN=<tokenvalue>

`*Note:`*

Mandatory parameters are marked in redand placeholder content is marked in <green>.

add

URL:http://<Citrix-ADC-IP-address(NSIP)/nitro/v1/config/wisite_accessmethod_binding

HTTP Method:PUT

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Content-Type:application/json

Request Payload:

```
{
```

```
"wisite_accessmethod_binding":{

<b>"sitepath":<String_value>,

</b>"accessmethod":<String_value>,

"clientipaddress":<String_value>,

"clientnetmask":<String_value>

}}
```

Response:

HTTP Status Code on Success: 201 Created HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error

delete

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/wisite_accessmethod_binding/sitepath_value<String>

Query-parameters:

args

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/wisite_accessmethod_binding/sitepath_value<String>?**args=clientipaddress:<String_value>,clientnetmask:<String_value>**

HTTP Method:DELETE

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue>

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error

get

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/wisite_accessmethod_binding/sitepath_value<String>

Query-parameters:

filter

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/wisite_accessmethod_binding/sitepath_value<String>?**filter=property-name1:property-value1,property-name2:property-value2**

Use this query-parameter to get the filtered set of wisite_accessmethod_binding resources configured on Citrix ADC. Filtering can be done on any of the properties of the resource.

pagination

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/wisite_accessmethod_binding/sitepath_value<String>?**pagesize=#no;pageno=#no**

Use this query-parameter to get the wisite_accessmethod_binding resources in chunks.

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "wisite_accessmethod_binding": [ {

"clientipaddress":<String_value>,
```

```
"clientnetmask":<String_value>,

"accessmethod":<String_value>,

"sitepath":<String_value>

}]}
```

get (all)

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/wisite_accessmethod_binding

Query-parameters:

bulkbindings

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/wisite_accessmethod_binding?**bulkbindings=yes**

NITRO allows you to fetch bindings in bulk.

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "wisite_accessmethod_binding": [ {

"clientipaddress":<String_value>,
```

```
"clientnetmask":<String_value>,

"accessmethod":<String_value>,

"sitepath":<String_value>

}]}
```

count

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/wisite_accessmethod_binding/sitepath_value<String>?**count=yes**

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{"wisite_accessmethod_binding": [ { "__count": "#no"} ] }
```

wisite_binding

Binding object which returns the resources bound to wisite.

Properties

(click to see Operations)

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
sitepath	<String>	Read-write	Path of a Web Interface site whose details you want the Citrix ADC to display. Minimum length = 1 Maximum length = 250
wisite_translationinternalip_binding	<wisite_translationinternalip_binding[]>	Read-only	translationinternalip that can be bound to wisite.
wisite_accessmethod_binding	<wisite_accessmethod_binding[]>	Read-only	accessmethod that can be bound to wisite.
wisite_farmname_binding	<wisite_farmname_binding[]>	Read-only	farmname that can be bound to wisite.

Operations

(click to see Properties)

GET| GET (ALL)

Some options that you can use for each operations:

- **Getting warnings in response:**NITRO allows you to get warnings in an operation by specifying the "warning" query parameter as "yes". For example, to get warnings while connecting to the Citrix ADC appliance, the URL is as follows:

  http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/login?warning=yes

  If any, the warnings are displayed in the response payload with the HTTP code "209 X-NITRO-WARNING".

- **Authenticated access for individual NITRO operations:**NITRO allows you to logon to the Citrix ADC appliance to perform individual operations. You can use this option instead of creating a NITRO session (using the login object) and then using that session to perform all operations,

To do this, you must specify the username and password in the request header of the NITRO request as follows:

X-NITRO-USER:<username>

X-NITRO-PASS:<password>

Note: In such cases, make sure that the request header DOES not include the following:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue>

`*Note:`*

Mandatory parameters are marked in redand placeholder content is marked in <green>.

get

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/wisite_binding/sitepath_value<String>

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "wisite_binding": [ {

"sitepath":<String_value>,

"wisite_translationinternalip_binding":<wisite_translationinternalip_binding[]_value>,

"wisite_accessmethod_binding":<wisite_accessmethod_binding[]_value>,
```

```
"wisite_farmname_binding":<wisite_farmname_binding[]_value>

}]}
```

get (all)

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/wisite_binding

Query-parameters:

bulkbindings

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/wisite_binding?**bulkbindings=yes**

NITRO allows you to fetch bindings in bulk.

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "wisite_binding": [ {

"sitepath":<String_value>,

"wisite_translationinternalip_binding":<wisite_translationinternalip_bindi
ng[]_value>,

"wisite_accessmethod_binding":<wisite_accessmethod_binding[]_value>,
```

```
"wisite_farmname_binding":<wisite_farmname_binding[]_value>

}]}
```

wisite_farmname_binding

Binding object showing the farmname that can be bound to wisite.

Properties

(click to see Operations)

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
sitepath	<String>	Read-write	Path to the Web Interface site. Minimum length = 1 Maximum length = 250
groups	<String>	Read-write	Active Directory groups that are permitted to enumerate resources from server farms. Including a setting for this parameter activates the user roaming feature. A maximum of 512 user groups can be specified for each farm defined with the Farm<n> parameter. The groups must be comma separated.
xmlport	<Double>	Read-write	Port number at which to contact the XML service.

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
transport	<String>	Read-write	Transport protocol to use for transferring data, related to the Web Interface site, between the Citrix ADC and the XML service. Possible values = HTTP, HTTPS, SSLRELAY
sslrelayport	<Double>	Read-write	TCP port at which the XenApp or XenDesktop servers listenfor SSL Relay traffic from the Citrix ADC. This parameter is required if you have set SSL Relay as the transport protocol. Web Interface uses root certificates when authenticating a server running SSL Relay. Make sure that all the servers running SSL Relay are configured to listen on the same port.
farmname	<String>	Read-write	Name for the logical representation of a XenApp or XenDesktop farm to be bound to the Web Interface site. Must begin with an ASCII alphabetic or

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
			underscore (_) character, and must contain only ASCII alphanumeric, underscore, hash (#), period (.), space, colon (:), at (@), equals (=), and hyphen (-) characters.
loadbalance	<String>	Read-write	Use all the XML servers (load balancing mode) or only one server (failover mode). Possible values = ON, OFF
recoveryfarm	<String>	Read-write	Binded farm is set as a recovery farm. Default value: OFF Possible values = ON, OFF
xmlserveraddresses	<String>	Read-write	Comma-separated IP addresses or host names of XenApp or XenDesktop servers providing XML services.
__count	<Double>	Read-write	count parameter

Operations

(click to see Properties)

ADD:| DELETE:| GET| GET (ALL)| COUNT

Some options that you can use for each operations:

- **Getting warnings in response:**NITRO allows you to get warnings in an operation by specifying the "warning" query parameter as "yes". For example, to get warnings while connecting to the Citrix ADC appliance, the URL is as follows:

  http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/login?warning=yes

  If any, the warnings are displayed in the response payload with the HTTP code "209 X-NITRO-WARNING".

- **Authenticated access for individual NITRO operations:**NITRO allows you to logon to the Citrix ADC appliance to perform individual operations. You can use this option instead of creating a NITRO session (using the login object) and then using that session to perform all operations,

  To do this, you must specify the username and password in the request header of the NITRO request as follows:

  X-NITRO-USER:<username>

  X-NITRO-PASS:<password>

  Note: In such cases, make sure that the request header DOES not include the following:

  Cookie:NITRO_AUTH_TOKEN=<tokenvalue>

`*Note:`*

Mandatory parameters are marked in redand placeholder content is marked in <green>.

add

URL:http://<Citrix-ADC-IP-address(NSIP)/nitro/v1/config/wisite_farmname_binding

HTTP Method:PUT

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Content-Type:application/json

Request Payload:

```
{
```

```
"wisite_farmname_binding":{

<b>"sitepath":<String_value>,

</b>"farmname":<String_value>,

"xmlserveraddresses":<String_value>,

"groups":<String_value>,

"recoveryfarm":<String_value>,

"xmlport":<Double_value>,

"transport":<String_value>,

"sslrelayport":<Double_value>,

"loadbalance":<String_value>

}}
```

Response:

HTTP Status Code on Success: 201 Created HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error

delete

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/wisite_farmname_binding/sitepath_value<String>

Query-parameters:

args

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/wisite_farmname_binding/sitepath_value<String>?**args=farmname:<String_value>**

HTTP Method:DELETE

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue>

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error

get

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/wisite_farmname_binding/sitepath_value<String>

Query-parameters:

filter

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/wisite_farmname_binding/sitepath_value<String>?**filter=property-name1:property-value1,property-name2:property-value2**

Use this query-parameter to get the filtered set of wisite_farmname_binding resources configured on Citrix ADC. Filtering can be done on any of the properties of the resource.

pagination

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/wisite_farmname_binding/sitepath_value<String>?**pagesize=#no;pageno=#no**

Use this query-parameter to get the wisite_farmname_binding resources in chunks.

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "wisite_farmname_binding": [ {

"sitepath":<String_value>,

"groups":<String_value>,

"xmlport":<Double_value>,

"transport":<String_value>,

"sslrelayport":<Double_value>,

"farmname":<String_value>,

"loadbalance":<String_value>,

"recoveryfarm":<String_value>,

"xmlserveraddresses":<String_value>

}]}
```

get (all)

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/wisite_farmname_binding

Query-parameters:

bulkbindings

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/wisite_farmname_binding?**bulkbindings=yes**

NITRO allows you to fetch bindings in bulk.

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "wisite_farmname_binding": [ {

"sitepath":<String_value>,

"groups":<String_value>,

"xmlport":<Double_value>,

"transport":<String_value>,

"sslrelayport":<Double_value>,

"farmname":<String_value>,

"loadbalance":<String_value>,

"recoveryfarm":<String_value>,

"xmlserveraddresses":<String_value>

}]}
```

count

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/wisite_farmname_binding/sitepath_value<String>?**count=yes**

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{"wisite_farmname_binding": [ { "__count": "#no"} ] }
```

wisite_translationinternalip_binding

Binding object showing the translationinternalip that can be bound to wisite.

Properties

(click to see Operations)

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
sitepath	<String>	Read-write	Path to the Web Interface site. Minimum length = 1 Maximum length = 250
accesstype	<String>	Read-write	Type of access to the XenApp or XenDesktop server. Available settings function as follows: * User Device - Clients can use the

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
			translated address of the mapping entry to connect to the XenApp or XenDesktop server. * Gateway - Access Gateway can use the translated address of the mapping entry to connect to the XenApp or XenDesktop server. * User Device and Gateway - Both clients and Access Gateway can use the translated address of the mapping entry to connect to the XenApp or XenDesktop server. Default value: UserDevice Possible values = UserDevice, Gateway, UserDeviceAndGateway
translationexternalport	<Integer>	Read-write	External port number associated with the server's port number. Range 1 - 65535 * in CLI is represented as 65535 in NITRO API

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
translationinternalip	<String>	Read-write	IP address of the server for which you want to associate an external IP address. (Clients access the server through the associated external address and port.). Default value: 0
translationexternalip	<String>	Read-write	External IP address associated with server's IP address.
translationinternalport	<Integer>	Read-write	Port number of the server for which you want to associate an external port. (Clients access the server through the associated external address and port.). Range 1 - 65535 * in CLI is represented as 65535 in NITRO API
__count	<Double>	Read-write	count parameter

Operations

(click to see Properties)

ADD:| DELETE:| GET| GET (ALL)| COUNT

Some options that you can use for each operations:

- **Getting warnings in response:**NITRO allows you to get warnings in an operation by specifying the "warning" query parameter as "yes". For example, to get warnings while connecting to the Citrix ADC appliance, the URL is as follows:

  http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/login?warning=yes

  If any, the warnings are displayed in the response payload with the HTTP code "209 X-NITRO-WARNING".

- **Authenticated access for individual NITRO operations:**NITRO allows you to logon to the Citrix ADC appliance to perform individual operations. You can use this option instead of creating a NITRO session (using the login object) and then using that session to perform all operations,

  To do this, you must specify the username and password in the request header of the NITRO request as follows:

  X-NITRO-USER:<username>

  X-NITRO-PASS:<password>

  Note: In such cases, make sure that the request header DOES not include the following:

  Cookie:NITRO_AUTH_TOKEN=<tokenvalue>

`*Note:`*

Mandatory parameters are marked in redand placeholder content is marked in <green>.

add

URL:http://<Citrix-ADC-IP-address(NSIP)/nitro/v1/config/wisite_translationinternalip_binding

HTTP Method:PUT

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Content-Type:application/json

Request Payload:

```
{
```

```
"wisite_translationinternalip_binding":{

<b>"sitepath":<String_value>,

</b>"translationinternalip":<String_value>,

"translationinternalport":<Integer_value>,

"translationexternalip":<String_value>,

"translationexternalport":<Integer_value>,

"accesstype":<String_value>

}}
```

Response:

HTTP Status Code on Success: 201 Created HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error

delete

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/wisite_translationinternalip_binding/sitepath_value<String>

Query-parameters:

args

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/wisite_translationinternalip_binding/sitepath_value<String>?
args=translationinternalip:<String_value>,translationinternalport:<Integer_value>,translationexternalip:

HTTP Method:DELETE

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue>

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error

get

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/wisite_translationinternalip_binding/sitepath_value<String>

Query-parameters:

filter

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/wisite_translationinternalip_binding/sitepath_value<String>?**filter=property-name1:property-value1,property-name2:property-value2**

Use this query-parameter to get the filtered set of wisite_translationinternalip_binding resources configured on Citrix ADC. Filtering can be done on any of the properties of the resource.

pagination

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/wisite_translationinternalip_binding/sitepath_value<String>?**pagesize=#no;pageno=#no**

Use this query-parameter to get the wisite_translationinternalip_binding resources in chunks.

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "wisite_translationinternalip_binding": [ {

"sitepath":<String_value>,

"accesstype":<String_value>,

"translationexternalport":<Integer_value>,

"translationinternalip":<String_value>,

"translationexternalip":<String_value>,

"translationinternalport":<Integer_value>

}]}
```

get (all)

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/wisite_translationinternalip_binding

Query-parameters:

bulkbindings

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/wisite_translationinternalip_binding?**bulkbindings=yes**

NITRO allows you to fetch bindings in bulk.

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "wisite_translationinternalip_binding": [ {

"sitepath":<String_value>,

"accesstype":<String_value>,

"translationexternalport":<Integer_value>,

"translationinternalip":<String_value>,

"translationexternalip":<String_value>,

"translationinternalport":<Integer_value>

}]}
```

count

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/wisite_translationinternalip_binding/sitepath_value<String>?**count=yes**

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{"wisite_translationinternalip_binding": [ { "__count": "#no"} ] }
```

WF

Wf

Web Front configuration.

NAME	DESCRIPTION
wfpackage	Configuration for Web Front
wfsite	Configuration for WF site

wfpackage

Configuration for Web Front resource.

Properties

(click to see Operations)

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
jre	<String>	Read-write	Complete path to the JRE tar file. You can use OpenJDK7 package for FreeBSD 8.x/amd64.The Java package can be downloaded from http://ftp-archive.freebsd.org/pub/FreeBSD-Archive/old-releases/amd64/

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
			amd64/8.4-RELEASE/packages/java/openjdk-7.17.02_2.tbz or http://www.freebsdfoundation.org/cgi-bin/download?download=diablo-jdk-freebsd6.amd64.1.7.17.07.02.tbz. Minimum length = 1 Maximum length = 255
wf	<String>	Read-write	Complete path to the WebFront tar file for installing the WebFront on the Citrix ADC. This file includes Apache Tomcat Web server. The file name has the following format: nswf-<version number>.tar (for example, nswf-1.5.tar). Minimum length = 1 Maximum length = 255

Operations

(click to see Properties)

INSTALL| GET (ALL)

Some options that you can use for each operations:

- **Getting warnings in response:**NITRO allows you to get warnings in an operation by specifying the "warning" query parameter as "yes". For example, to get warnings while connecting to the Citrix ADC appliance, the URL is as follows:

  http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/login?warning=yes

  If any, the warnings are displayed in the response payload with the HTTP code "209 X-NITRO-WARNING".

- **Authenticated access for individual NITRO operations:**NITRO allows you to logon to the Citrix ADC appliance to perform individual operations. You can use this option instead of creating a NITRO session (using the login object) and then using that session to perform all operations,

  To do this, you must specify the username and password in the request header of the NITRO request as follows:

  X-NITRO-USER:<username>

  X-NITRO-PASS:<password>

  Note: In such cases, make sure that the request header DOES not include the following:

  Cookie:NITRO_AUTH_TOKEN=<tokenvalue>

`*Note:`*

Mandatory parameters are marked in redand placeholder content is marked in <green>.

Install

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/wfpackage

HTTP Method:POST

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Content-Type:application/json

Request Payload:

```
{"wfpackage":{

"jre":<String_value>,

"wf":<String_value>

}}
```

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error

get (all)

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/wfpackage

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "wfpackage": [ {

"jre":<String_value>,

"wf":<String_value>
```

```
}]}
```

wfsite

Configuration for WF site resource.

Properties

(click to see Operations)

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
sitename	<String>	Read-write	Name of the WebFront site being created on the Citrix ADC. Minimum length = 1 Maximum length = 255
storefronturl	<String>	Read-write	FQDN or IP of Windows StoreFront server where the store is configured. Minimum length = 1 Maximum length = 255
storename	<String>	Read-write	Name of the store present in StoreFront. Minimum length = 1 Maximum length = 255
html5receiver	<String>	Read-write	Specifies whether or not to use HTML5 receiver for

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
			launching apps for the WF site. Default value: FALLBACK Possible values = ALWAYS, FALLBACK, OFF
workspacecontrol	<String>	Read-write	Specifies whether of not to use workspace control for the WF site. Default value: ON Possible values = ON, OFF
displayroamingaccounts	<String>	Read-write	Specifies whether or not to display the accounts selection screen during First Time Use of Receiver . Default value: ON Possible values = ON, OFF
xframeoptions	<String>	Read-write	The value to be sent in the X-Frame-Options header. WARNING: Setting this option to ALLOW could leave the end user vulnerable to Click Jacking attacks.

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
			Default value: DENY Possible values = ALLOW, DENY
state	<String>	Read-only	State of wf site. Default value: UP Possible values = UP, INITIALIZING, DOWN-HostUnknown, DOWN-ReqTimeout, DOWN-Wrong Store, DOWN-SF Error, DOWN-SSL Error, DOWN-Conn Reset, DOWN
__count	<Double>	Read-only	count parameter

Operations

(click to see Properties)

ADD| DELETE| UPDATE| GET (ALL)| GET| COUNT

Some options that you can use for each operations:

- **Getting warnings in response:**NITRO allows you to get warnings in an operation by specifying the "warning" query parameter as "yes". For example, to get warnings while connecting to the Citrix ADC appliance, the URL is as follows:

  http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/login?warning=yes

  If any, the warnings are displayed in the response payload with the HTTP code "209 X-NITRO-WARNING".

- **Authenticated access for individual NITRO operations:**NITRO allows you to logon to the Citrix ADC appliance to perform individual operations. You can use this option instead of creating a NITRO session (using the login object) and then using that session to perform all operations,

  To do this, you must specify the username and password in the request header of the NITRO request as follows:

  X-NITRO-USER:<username>

  X-NITRO-PASS:<password>

  Note: In such cases, make sure that the request header DOES not include the following:

  Cookie:NITRO_AUTH_TOKEN=<tokenvalue>

`*Note:`*

Mandatory parameters are marked in redand placeholder content is marked in <green>.

add

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/wfsite

HTTP Method:POST

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Content-Type:application/json

Request Payload:

```
{"wfsite":{

<b>"sitename":<String_value>,

</b><b>"storefronturl":<String_value>,

</b><b>"storename":<String_value>,

</b>"html5receiver":<String_value>,

"workspacecontrol":<String_value>,
```

```
"displayroamingaccounts":<String_value>,

"xframeoptions":<String_value>

}}
```

Response:

HTTP Status Code on Success: 201 Created HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error

delete

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/wfsite/sitename_value<String>

HTTP Method:DELETE

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue>

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error

update

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/wfsite

HTTP Method:PUT

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Content-Type:application/json

Request Payload:

```
{"wfsite":{
```

```
<b>"sitename":<String_value>,

</b>"storefronturl":<String_value>,

"storename":<String_value>,

"html5receiver":<String_value>,

"workspacecontrol":<String_value>,

"displayroamingaccounts":<String_value>,

"xframeoptions":<String_value>

}}
```

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error

get (all)

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/wfsite

Query-parameters:

attrs

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/wfsite?**attrs=property-name1,property-name2**

Use this query parameter to specify the resource details that you want to retrieve.

filter

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/wfsite?**filter=property-name1:property-val1,property-name2:property-val2**

Use this query-parameter to get the filtered set of wfsite resources configured on Citrix ADC. Filtering can be done on any of the properties of the resource.

view

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/wfsite?**view=summary**

Note: By default, the retrieved results are displayed in detail view (?view=detail).

pagination

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/wfsite?**pagesize=#no;pageno=#no**

Use this query-parameter to get the wfsite resources in chunks.

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "wfsite": [ {

"sitename":<String_value>,

"storefronturl":<String_value>,

"storename":<String_value>,

"html5receiver":<String_value>,

"workspacecontrol":<String_value>,

"displayroamingaccounts":<String_value>,

"xframeoptions":<String_value>,
```

```
"state":<String_value>

}]}
```

get

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/wfsite/sitename_value<String>

Query-parameters:

attrs

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/wfsite/sitename_value<String>?**attrs=property-name1,property-name2**

Use this query parameter to specify the resource details that you want to retrieve.

view

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/wfsite/sitename_value<String>?**view=summary**

Note: By default, the retrieved results are displayed in detail view (?view=detail).

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "wfsite": [ {
```

```
"sitename":<String_value>,

"storefronturl":<String_value>,

"storename":<String_value>,

"html5receiver":<String_value>,

"workspacecontrol":<String_value>,

"displayroamingaccounts":<String_value>,

"xframeoptions":<String_value>,

"state":<String_value>

}]}
```

count

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/wfsite?**count=yes**

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "wfsite": [ { "__count": "#no"} ] }
```

XM

Xm

XenMobile deployment configuration.

NAME	DESCRIPTION
xmdeployment	Configuration for XenMobile Deployment
xmpackage	Configuration for XenMobile Deployment Package

xmdeployment

Configuration for XenMobile Deployment resource.

Properties

(click to see Operations)

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
name	<String>	Read-write	XenMobile deployment name. Minimum length = 1 Maximum length = 255
frompackage	<String>	Read-write	XenMobile package name. Minimum length = 1 Maximum length = 255
config	<String>	Read-write	XenMobile deployment config data.

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
			Minimum length = 1 Maximum length = 4095
meta	<String>	Read-only	XenMobile deployment meta data. Minimum length = 1 Maximum length = 16383
__count	<Double>	Read-only	count parameter

Operations

(click to see Properties)

ADD| DELETE| GET (ALL)| GET| COUNT

Some options that you can use for each operations:

- **Getting warnings in response:**NITRO allows you to get warnings in an operation by specifying the "warning" query parameter as "yes". For example, to get warnings while connecting to the Citrix ADC appliance, the URL is as follows:

  http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/login?warning=yes

  If any, the warnings are displayed in the response payload with the HTTP code "209 X-NITRO-WARNING".

- **Authenticated access for individual NITRO operations:**NITRO allows you to logon to the Citrix ADC appliance to perform individual operations. You can use this option instead of creating a NITRO session (using the login object) and then using that session to perform all operations,

  To do this, you must specify the username and password in the request header of the NITRO request as follows:

  X-NITRO-USER:<username>

X-NITRO-PASS:<password>

Note: In such cases, make sure that the request header DOES not include the following:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue>

`*Note:`*

Mandatory parameters are marked in redand placeholder content is marked in <green>.

add

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/xmdeployment

HTTP Method:POST

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Content-Type:application/json

Request Payload:

```
{"xmdeployment":{

<b>"name":<String_value>,

</b><b>"frompackage":<String_value>,

</b><b>"config":<String_value>

</b>}}
```

Response:

HTTP Status Code on Success: 201 Created HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error

delete

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/xmdeployment/name_value<String>

HTTP Method:DELETE

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue>

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error

get (all)

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/xmdeployment

Query-parameters:

attrs

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/xmdeployment?**attrs=property-name1,property-name2**

Use this query parameter to specify the resource details that you want to retrieve.

filter

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/xmdeployment?**filter=property-name1:property-val1,property-name2:property-val2**

Use this query-parameter to get the filtered set of xmdeployment resources configured on Citrix ADC. Filtering can be done on any of the properties of the resource.

view

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/xmdeployment?**view=summary**

Note: By default, the retrieved results are displayed in detail view (?view=detail).

pagination

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/xmdeployment?**pagesize=#no;pageno=#no**

Use this query-parameter to get the xmdeployment resources in chunks.

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "xmdeployment": [ {

"name":<String_value>,

"meta":<String_value>

}]}
```

get

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/xmdeployment/name_value<String>

Query-parameters:

attrs

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/xmdeployment/name_value<String>?**attrs=property-name1,property-name2**

Use this query parameter to specify the resource details that you want to retrieve.

view

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/xmdeployment/name_value<String>?**view=summary**

Note: By default, the retrieved results are displayed in detail view (?view=detail).

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "xmdeployment": [ {

"name":<String_value>,

"meta":<String_value>

}]}
```

count

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/xmdeployment?**count=yes**

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "xmdeployment": [ { "__count": "#no"} ] }
```

xmpackage

Configuration for XenMobile Deployment Package resource.

Properties

(click to see Operations)

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
name	<String>	Read-write	XenMobile package name. Minimum length = 1 Maximum length = 255
packagefile	<String>	Read-write	Path to the upload XenMobile package file. Minimum length = 1 Maximum length = 255
identifier	<String>	Read-only	XenMobile deployment name. Minimum length = 1 Maximum length = 255
meta	<String>	Read-only	XenMobile deployment meta data. Minimum length = 1

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
			Maximum length = 16383
__count	<Double>	Read-only	count parameter

Operations

(click to see Properties)

ADD| DELETE| GET (ALL)| GET| COUNT

Some options that you can use for each operations:

- **Getting warnings in response:**NITRO allows you to get warnings in an operation by specifying the "warning" query parameter as "yes". For example, to get warnings while connecting to the Citrix ADC appliance, the URL is as follows:

  http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/login?warning=yes

  If any, the warnings are displayed in the response payload with the HTTP code "209 X-NITRO-WARNING".

- **Authenticated access for individual NITRO operations:**NITRO allows you to logon to the Citrix ADC appliance to perform individual operations. You can use this option instead of creating a NITRO session (using the login object) and then using that session to perform all operations,

  To do this, you must specify the username and password in the request header of the NITRO request as follows:

  X-NITRO-USER:<username>

  X-NITRO-PASS:<password>

  Note: In such cases, make sure that the request header DOES not include the following:

  Cookie:NITRO_AUTH_TOKEN=<tokenvalue>

`*Note:`*

Mandatory parameters are marked in redand placeholder content is marked in <green>.

add

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/xmpackage

HTTP Method:POST

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Content-Type:application/json

Request Payload:

```
{"xmpackage":{

<b>"name":<String_value>,

</b><b>"packagefile":<String_value>

</b>}}
```

Response:

HTTP Status Code on Success: 201 Created HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error

delete

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/xmpackage/name_value<String>

HTTP Method:DELETE

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue>

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error

get (all)

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/xmpackage

Query-parameters:

attrs

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/xmpackage?**attrs=property-name1,property-name2**

Use this query parameter to specify the resource details that you want to retrieve.

filter

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/xmpackage?**filter=property-name1:property-val1,property-name2:property-val2**

Use this query-parameter to get the filtered set of xmpackage resources configured on Citrix ADC. Filtering can be done on any of the properties of the resource.

view

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/xmpackage?**view=summary**

Note: By default, the retrieved results are displayed in detail view (?view=detail).

pagination

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/xmpackage?**pagesize=#no;pageno=#no**

Use this query-parameter to get the xmpackage resources in chunks.

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "xmpackage": [ {

"name":<String_value>,

"identifier":<String_value>,

"meta":<String_value>

}]}
```

get

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/xmpackage/name_value<String>

Query-parameters:

attrs

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/xmpackage/name_value<String>?**attrs=property-name1,property-name2**

Use this query parameter to specify the resource details that you want to retrieve.

view

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/xmpackage/name_value<String>?**view=summary**

Note: By default, the retrieved results are displayed in detail view (?view=detail).

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "xmpackage": [ {

"name":<String_value>,

"identifier":<String_value>,

"meta":<String_value>

}]}
```

count

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/xmpackage?**count=yes**

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "xmpackage": [ { "__count": "#no"} ] }
```

Statistics

The statistics API is arranged in a logical groups of objects. Choose an appropriate group to find more details.

NAME	DESCRIPTION
AAA	Authentication, authorization, and accounting service statistics.
Appflow	AppFlow statistics.
Application Firewall	Application Firewall statistics.
Appqoe	Application Level Quality of Experience statistics
Audit	Audit statistics.
Authentication	Authentication statistics.
Authorization	Authorization statistics. Authorization services check which resources users are authorized to access, and grant permissions accordingly.
Autoscale	AutoScale statistics.
Basic	Basic system statistics.
Ca	Content Accelerator statistics. The Content Accelerator feature is used to cache static and dynamic web application data.
Integrated Caching	Integrated Caching statistics. The Integrated Caching feature is used to cache static and dynamic web application data.
Cluster	Cluster commands.

NAME	DESCRIPTION
Compression	Compression statistics. The systems feature for compressing HTTP responses to compression-aware browsers.
CI	Content Inspection
Cache Redirection	Cache redirection statistics. The cache redirection feature can transparently redirect cacheable HTTP requests to a cache and send non-cacheable or dynamic HTTP requests to the origin server. A cache stores or caches frequently requested web content and serves such web content to a client on behalf of the origin servers, alleviating the load on the origin server farm.
Content Switching	Content Switching statistics. Content Switching feature that enables you to direct traffic to servers on the basis of content.
Domain Name Service	Domain Name Service(DNS) statistics.
HTTP DoS Protection	HTTP DOS protection statistics.
Front-end-optimization	Front end optimization statistics. The systems feature to optimize Web content for performance.
Global Server Load Balancing	Global Server Load Balancing (GSLB) statistics. GSLB feature ensures that client requests are directed to a best performing site available in a global enterprise and distributed Internet environment.
High Availability	High-availability commands.
Ica	ICA statistics.
Ipsec	IPSEC statistics.
Ipsecalg	IPSECALG statistics.

NAME	DESCRIPTION
Load Balancing	Load Balancing statistics. The load balancing methods manage the selection of an appropriate physical server in a server farm.
LLDP	Link Layer Discovery Protocol.
Lsn	Large Scale NAT commands
Network	Networking commands.
NS	System/Global level statistics.
Pcp	Port Control Protocol statistics.
Priority Queuing	Priority Queuing statistics. Priority Queuing ensures that key customer web sites get the timely responses they need. System provides priority-based queuing, which provides different levels of service to customers.
Protocol	Protocol statistics.
Qos	QoS statistics.
Responder	Responder statistics.
Rewrite	Rewrite statistics.
Sure Connect	SureConnect statistics. SureConnect allows the system to deliver alternate content to clients when the system is under heavy load for the original URL. The original content will be served to the client, automatically, after an estimated delay period.
SNMP	SNMP(Simple Network Management Protocol) statistics.
Spillover	Spillover policies and actions.
SSL	SSL statistics

NAME	DESCRIPTION
Stream	Stream statistics.
System	System statistics.
Traffic Management	TM session/policy statistics.
Transform	URL Transformation statistics.
Tunnel	SSL VPN Tunnel statistics.
User	User protocol statistics.
VideoOptimization	Video optimization feature is used to show (i) the stats of different media types that are being served by the Citrix ADC and (ii) the details of optimization applied on ABR videos
SSL VPN	Virtual Private Network statistics.

aaa

Statistics for aaa.

Properties

(click to see Operations)

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
clearstats	<String>	Read-write	Clear the statsistics / counters. Possible values = basic, full
aaaauthsuccess	<Double>	Read-only	Count of authentication successes.
aaaauthsuccessrate	<Double>	Read-only	Rate (/s) counter for aaaauthsuccess

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
aaaauthfail	<Double>	Read-only	Count of authentication failures.
aaaauthfailrate	<Double>	Read-only	Rate (/s) counter for aaaauthfail
aaaauthonlyhttpsuccess	<Double>	Read-only	Count of HTTP connections that succeeded authorization.
aaaauthonlyhttpsuccessrate	<Double>	Read-only	Rate (/s) counter for aaaauthonlyhttpsuccess
aaaauthonlyhttpfail	<Double>	Read-only	Count of HTTP connections that failed authorization.
aaaauthonlyhttpfailrate	<Double>	Read-only	Rate (/s) counter for aaaauthonlyhttpfail
aaaauthnonhttpsuccess	<Double>	Read-only	Count of non HTTP connections that succeeded authorization.
aaaauthnonhttpsuccessrate	<Double>	Read-only	Rate (/s) counter for aaaauthnonhttpsuccess
aaaauthnonhttpfail	<Double>	Read-only	Count of non HTTP connections that failed authorization.
aaaauthnonhttpfailrate	<Double>	Read-only	Rate (/s) counter for aaaauthnonhttpfail

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
aaacursessions	<Double>	Read-only	Count of current SmartAccess AAA sessions.
aaacursessionsrate	<Double>	Read-only	Rate (/s) counter for aaacursessions
aaatotsessions	<Double>	Read-only	Count of all SmartAccess AAA sessions.
aaasessionsrate	<Double>	Read-only	Rate (/s) counter for aaatotsessions
aaatotsessiontimeout	<Double>	Read-only	Count of AAA sessions that have timed out.
aaasessiontimeoutrate	<Double>	Read-only	Rate (/s) counter for aaatotsessiontimeout
aaacuricasessions	<Double>	Read-only	Count of current Basic ICA only sessions.
aaacuricasessionsrate	<Double>	Read-only	Rate (/s) counter for aaacuricasessions
aaacuricaonlyconn	<Double>	Read-only	Count of current Basic ICA only connections.
aaacuricaonlyconnrate	<Double>	Read-only	Rate (/s) counter for aaacuricaonlyconn
aaacuricaconn	<Double>	Read-only	Count of current SmartAccess ICA connections.
aaacuricaconnrate	<Double>	Read-only	Rate (/s) counter for aaacuricaconn

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
aaacurtmsessions	<Double>	Read-only	Count of current AAATM sessions.
aaacurtmsessionsrate	<Double>	Read-only	Rate (/s) counter for aaacurtmsessions
aaatottmsessions	<Double>	Read-only	Count of all AAATM sessions.
aaatmsessionsrate	<Double>	Read-only	Rate (/s) counter for aaatottmsessions

Operations

(click to see Properties)

GET (ALL)

Some options that you can use for each operations:

- **Getting warnings in response:**NITRO allows you to get warnings in an operation by specifying the "warning" query parameter as "yes". For example, to get warnings while connecting to the Citrix ADC appliance, the URL is as follows:

  http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/login?warning=yes

  If any, the warnings are displayed in the response payload with the HTTP code "209 X-NITRO-WARNING".
- **Authenticated access for individual NITRO operations:**NITRO allows you to logon to the Citrix ADC appliance to perform individual operations. You can use this option instead of creating a NITRO session (using the login object) and then using that session to perform all operations,

  To do this, you must specify the username and password in the request header of the NITRO request as follows:

  X-NITRO-USER:<username>

  X-NITRO-PASS:<password>

  Note: In such cases, make sure that the request header DOES not include the following:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue>

`*Note:`*

Mandatory parameters are marked in redand placeholder content is marked in <green>.

get (all)

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/stat/aaa

Query-parameters:

args

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/stat/aaa?
args=detail:<Boolean_value>,fullvalues:<Boolean_value>,ntimes:<Double_value>,logfile:<String_value

Use this query-parameter to get aaa resources based on additional properties.

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "aaa": [ {

"aaatotsessiontimeout":<Double_value>,

"aaaauthfail":<Double_value>,

"aaaauthsuccess":<Double_value>,
```

```
"aaatottmsessions":<Double_value>,

"aaaauthonlyhttpfailrate":<Double_value>,

"aaaauthnonhttpsuccess":<Double_value>,

"aaacuricaonlyconnrate":<Double_value>,

"aaaauthonlyhttpsuccessrate":<Double_value>,

"aaacurtmsessions":<Double_value>,

"aaacuricasessions":<Double_value>,

"aaatotsessions":<Double_value>,

"aaaauthonlyhttpfail":<Double_value>,

"aaacuricaonlyconn":<Double_value>,

"aaatmsessionsrate":<Double_value>,

"aaaauthfailrate":<Double_value>,

"aaacursessionsrate":<Double_value>,

"aaaauthnonhttpsuccessrate":<Double_value>,

"aaacuricaconn":<Double_value>,

"aaaauthnonhttpfailrate":<Double_value>,

"aaaauthnonhttpfail":<Double_value>,

"aaacuricaconnrate":<Double_value>,

"aaaauthsuccessrate":<Double_value>,

"aaaauthonlyhttpsuccess":<Double_value>,

"aaacurtmsessionsrate":<Double_value>,

"aaasessiontimeoutrate":<Double_value>,
```

```
"aaacuricasessionsrate":<Double_value>,

"aaacursessions":<Double_value>,

"aaasessionsrate":<Double_value>

}]}
```

appflow

Statistics for appflow.

Properties

(click to see Operations)

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
clearstats	<String>	Read-write	Clear the statsistics / counters. Possible values = basic, full
appflowtotaltxoctets	<Double>	Read-only	The total number of AppFlow (IPFIX) octets that the Citrix ADC has transmitted.
appflowtxoctetsrate	<Double>	Read-only	Rate (/s) counter for appflowtotaltxoctets
appflowtotalflows	<Double>	Read-only	The total number of AppFlow (IPFIX) flows that the Citrix ADC has transmitted.
appflowflowsrate	<Double>	Read-only	Rate (/s) counter for appflowtotalflows

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
appflowtotaltxmessagess	<Double>	Read-only	The total number of AppFlow (IPFIX) messages that the Citrix ADC has transmitted.
appflowtxmessagessrate	<Double>	Read-only	Rate (/s) counter for appflowtotaltxmessagess
appflowtotalignoctets	<Double>	Read-only	The total number of octets that the Citrix ADC has ignored for AppFlow (IPFIX).
appflowignoctetsrate	<Double>	Read-only	Rate (/s) counter for appflowtotalignoctets
appflowtotalignpacketss	<Double>	Read-only	The total number of packets that the Citrix ADC has ignored for AppFlow (IPFIX).
appflowignpacketssrate	<Double>	Read-only	Rate (/s) counter for appflowtotalignpacketss
appflowtotalnotxoctets	<Double>	Read-only	The total number of AppFlow (IPFIX) octets that the Citrix ADC has not transmitted.
appflownotxoctetsrate	<Double>	Read-only	Rate (/s) counter for appflowtotalnotxoctets
appflowtotalnotxflows	<Double>	Read-only	The total number of AppFlow (IPFIX) flows that the Citrix

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
			ADC has not transmitted.
appflownotxflowsrate	<Double>	Read-only	Rate (/s) counter for appflowtotalnotxflows
appflowtotalnotxpackets	<Double>	Read-only	The total number of AppFlow (IPFIX) packets that the Citrix ADC has not transmitted.
appflownotxpacketsrate	<Double>	Read-only	Rate (/s) counter for appflowtotalnotxpackets

Operations

(click to see Properties)

GET (ALL)

Some options that you can use for each operations:

- **Getting warnings in response:**NITRO allows you to get warnings in an operation by specifying the "warning" query parameter as "yes". For example, to get warnings while connecting to the Citrix ADC appliance, the URL is as follows:

  http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/login?warning=yes

  If any, the warnings are displayed in the response payload with the HTTP code "209 X-NITRO-WARNING".

- **Authenticated access for individual NITRO operations:**NITRO allows you to logon to the Citrix ADC appliance to perform individual operations. You can use this option instead of creating a NITRO session (using the login object) and then using that session to perform all operations,

  To do this, you must specify the username and password in the request header of the NITRO request as follows:

X-NITRO-USER:<username>

X-NITRO-PASS:<password>

Note: In such cases, make sure that the request header DOES not include the following:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue>

`*Note:`*

Mandatory parameters are marked in redand placeholder content is marked in <green>.

get (all)

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/stat/appflow

Query-parameters:

args

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/stat/appflow?
args=detail:<Boolean_value>,fullvalues:<Boolean_value>,ntimes:<Double_value>,logfile:<String_value

Use this query-parameter to get appflow resources based on additional properties.

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "appflow": [ {

"appflowtxmessagessrate":<Double_value>,
```

```
"appflowtotalnotxoctets":<Double_value>,

"appflowtotaltxmessagess":<Double_value>,

"appflowtotalignpacketss":<Double_value>,

"appflowtotalflows":<Double_value>,

"appflowtotalnotxflows":<Double_value>,

"appflownotxflowsrate":<Double_value>,

"appflowtotalnotxpackets":<Double_value>,

"appflownotxoctetsrate":<Double_value>,

"appflowignpacketssrate":<Double_value>,

"appflowtxoctetsrate":<Double_value>,

"appflowignoctetsrate":<Double_value>,

"appflowtotalignoctets":<Double_value>,

"appflowtotaltxoctets":<Double_value>,

"appflownotxpacketsrate":<Double_value>,

"appflowflowsrate":<Double_value>

}]}
```

Application Firewall

Application Firewall statistics.

NAME	DESCRIPTION
appfw	Statistics for appfw.

NAME	DESCRIPTION
appfwpolicy	Statistics for application firewall policy resource.
appfwpolicylabel	Statistics for application firewall policy label resource.
appfwprofile	Statistics for application firewall profile resource.

appfw

Statistics for appfw.

Properties

(click to see Operations)

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
clearstats	<String>	Read-write	Clear the statsistics / counters. Possible values = basic, full
appfirewalltotallog	<Double>	Read-only	Total number of security check log messages generated by the Application Firewall.
appfirewalllograte	<Double>	Read-only	Rate (/s) counter for appfirewalltotallog
appfirewalltotalviol	<Double>	Read-only	Total number of security check violations seen by the Application Firewall.
appfirewallviolrate	<Double>	Read-only	Rate (/s) counter for appfirewalltotalviol

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
appfirewallshortavgresptime	<Double>	Read-only	Average backend response time in milliseconds over the last 7 seconds
appfirewalllongavgresptime	<Double>	Read-only	Average backend response time in milliseconds since reboot
appfirewallrequests	<Double>	Read-only	HTTP/HTTPS requests sent to your protected web servers via the Application Firewall.
appfirewallrequestsrate	<Double>	Read-only	Rate (/s) counter for appfirewallrequests
appfirewallreqbytes	<Double>	Read-only	Number of bytes transfered for requests
appfirewallreqbytesrate	<Double>	Read-only	Rate (/s) counter for appfirewallreqbytes
appfirewallresponses	<Double>	Read-only	HTTP/HTTPS responses sent by your protected web servers via the Application Firewall.
appfirewallresponsesrate	<Double>	Read-only	Rate (/s) counter for appfirewallresponses
appfirewallresbytes	<Double>	Read-only	Number of bytes transfered for responses
appfirewallresbytesrate	<Double>	Read-only	Rate (/s) counter for appfirewallresbytes

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
appfirewallaborts	<Double>	Read-only	Incomplete HTTP/ HTTPS requests aborted by the client before the Application Firewall could finish processing them.
appfirewallabortsrate	<Double>	Read-only	Rate (/s) counter for appfirewallaborts
appfirewallredirects	<Double>	Read-only	HTTP/HTTPS requests redirected by the Application Firewall to a different Web page or web server. (HTTP 302)
appfirewallredirectsrate	<Double>	Read-only	Rate (/s) counter for appfirewallredirects
appfirewalltrapsdropped	<Double>	Read-only	AppFirewall SNMP traps dropped due to time limit.
appfirewallviolstarturl	<Double>	Read-only	Number of Start URL security check violations seen by the Application Firewall.
appfirewallviolstarturlrate	<Double>	Read-only	Rate (/s) counter for appfirewallviolstarturl
appfirewallvioldenyurl	<Double>	Read-only	Number of Deny URL security check violations seen by the Application Firewall.

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
appfirewallvioldenyurlrate	<Double>	Read-only	Rate (/s) counter for appfirewallvioldenyurl
appfirewallviolreferer header	<Double>	Read-only	Number of Referer Header security check violations seen by the Application Firewall.
appfirewallviolreferer headerrate	<Double>	Read-only	Rate (/s) counter for appfirewallviolreferer header
appfirewallviolbuffero verflow	<Double>	Read-only	Number of Buffer Overflow security check violations seen by the Application Firewall.
appfirewallviolbuffero verflowrate	<Double>	Read-only	Rate (/s) counter for appfirewallviolbuffero verflow
appfirewallviolcookie	<Double>	Read-only	Number of Cookie Consistency security check violations seen by the Application Firewall.
appfirewallviolcookie rate	<Double>	Read-only	Rate (/s) counter for appfirewallviolcookie
appfirewallviolcsrftag	<Double>	Read-only	Number of Cross Site Request Forgery form tag security check violations seen by the Application Firewall.

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
appfirewallviolcsrftagrate	<Double>	Read-only	Rate (/s) counter for appfirewallviolcsrftag
appfirewallviolxss	<Double>	Read-only	Number of HTML Cross-Site Scripting security check violations seen by the Application Firewall.
appfirewallviolxssrate	<Double>	Read-only	Rate (/s) counter for appfirewallviolxss
appfirewallviolsql	<Double>	Read-only	Number of HTML SQL Injection security check violations seen by the Application Firewall.
appfirewallviolsqlrate	<Double>	Read-only	Rate (/s) counter for appfirewallviolsql
appfirewallviolfieldformat	<Double>	Read-only	Number of Field Format security check violations seen by the Application Firewall.
appfirewallviolfieldformatrate	<Double>	Read-only	Rate (/s) counter for appfirewallviolfieldformat
appfirewallviolfieldconsistency	<Double>	Read-only	Number of Field Consistency security check violations seen by the Application Firewall.

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
appfirewallviolfieldconsistencyrate	<Double>	Read-only	Rate (/s) counter for appfirewallviolfieldconsistency
appfirewallviolcreditcard	<Double>	Read-only	Number of Credit Card security check violations seen by the Application Firewall.
appfirewallviolcreditcardrate	<Double>	Read-only	Rate (/s) counter for appfirewallviolcreditcard
appfirewallviolsafeobject	<Double>	Read-only	Number of Safe Object security check violations seen by the Application Firewall.
appfirewallviolsafeobjectrate	<Double>	Read-only	Rate (/s) counter for appfirewallviolsafeobject
appfirewallviolsignature	<Double>	Read-only	Number of Signature violations seen by the Application Firewall.
appfirewallviolsignaturerate	<Double>	Read-only	Rate (/s) counter for appfirewallviolsignature
appfirewallviolcontenttype	<Double>	Read-only	Number of Content type security check violations seen by the Application Firewall.

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
appfirewallviolcontenttyperate	<Double>	Read-only	Rate (/s) counter for appfirewallviolcontenttype
appfirewallviolwellformednessviolations	<Double>	Read-only	Number of XML Format security check violations seen by the Application Firewall.
appfirewallviolwellformednessviolationsrate	<Double>	Read-only	Rate (/s) counter for appfirewallviolwellformednessviolations
appfirewallviolxdosviolations	<Double>	Read-only	Number of XML Denial-of-Service security check violations seen by the Application Firewall.
appfirewallviolxdosviolationsrate	<Double>	Read-only	Rate (/s) counter for appfirewallviolxdosviolations
appfirewallviolmsgvalviolations	<Double>	Read-only	Number of XML Message Validation security check violations seen by the Application Firewall.
appfirewallviolmsgvalviolationsrate	<Double>	Read-only	Rate (/s) counter for appfirewallviolmsgvalviolations
appfirewallviolwsiviolations	<Double>	Read-only	Number of Web Services Interoperability (WS-I) security check violations seen by

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
			the Application Firewall.
appfirewallviolwsiviolationsrate	<Double>	Read-only	Rate (/s) counter for appfirewallviolwsiviolations
appfirewallviolxmlsqlviolations	<Double>	Read-only	Number of XML SQL Injection security check violations seen by the Application Firewall.
appfirewallviolxmlsqlviolationsrate	<Double>	Read-only	Rate (/s) counter for appfirewallviolxmlsqlviolations
appfirewallviolxmlxssviolations	<Double>	Read-only	Number of XML Cross-Site Scripting (XSS) security check violations seen by the Application Firewall.
appfirewallviolxmlxssviolationsrate	<Double>	Read-only	Rate (/s) counter for appfirewallviolxmlxssviolations
appfirewallviolxmlattachmentviolations	<Double>	Read-only	Number of XML Attachment security check violations seen by the Application Firewall.
appfirewallviolxmlattachmentviolationsrate	<Double>	Read-only	Rate (/s) counter for appfirewallviolxmlattachmentviolations
appfirewallviolxmlsoapfaultviolations	<Double>	Read-only	Number of requests returning soap:fault

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
			from the backend server
appfirewallviolxmlsoapfaultviolationsrate	<Double>	Read-only	Rate (/s) counter for appfirewallviolxmlsoapfaultviolations
appfirewallviolxmlgenviolations	<Double>	Read-only	Number of requests returning XML generic error from the backend server
appfirewallviolxmlgenviolationsrate	<Double>	Read-only	Rate (/s) counter for appfirewallviolxmlgenviolations
appfirewalllogstarturl	<Double>	Read-only	Number of Start URL security check log messages generated by the Application Firewall.
appfirewalllogstarturlrate	<Double>	Read-only	Rate (/s) counter for appfirewalllogstarturl
appfirewalllogdenyurl	<Double>	Read-only	Number of Deny URL security check log messages generated by the Application Firewall.
appfirewalllogdenyurlrate	<Double>	Read-only	Rate (/s) counter for appfirewalllogdenyurl
appfirewalllogrefererheader	<Double>	Read-only	Number of Referer Header security check log messages generated by the Application Firewall.

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
appfirewalllogreferer headerrate	<Double>	Read-only	Rate (/s) counter for appfirewalllogreferer header
appfirewalllogbuffero verflow	<Double>	Read-only	Number of Buffer Overflow security check log messages generated by the Application Firewall.
appfirewalllogbuffero verflowrate	<Double>	Read-only	Rate (/s) counter for appfirewalllogbuffero verflow
appfirewalllogcookie	<Double>	Read-only	Number of Cookie Consistency security check log messages generated by the Application Firewall.
appfirewalllogcookier ate	<Double>	Read-only	Rate (/s) counter for appfirewalllogcookie
appfirewalllogcsrftag	<Double>	Read-only	Number of Cross Site Request Forgery form tag security check log messages generated by the Application Firewall.
appfirewalllogcsrftagr ate	<Double>	Read-only	Rate (/s) counter for appfirewalllogcsrftag
appfirewalllogxss	<Double>	Read-only	Number of HTML Cross-Site Scripting security check log messages generated by the Application Firewall.

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
appfirewalllogxssrate	<Double>	Read-only	Rate (/s) counter for appfirewalllogxss
appfirewalllogtransformxss	<Double>	Read-only	Number of HTML Cross-Site Scripting security check transform log messages generated by the Application Firewall.
appfirewalllogtransformxssrate	<Double>	Read-only	Rate (/s) counter for appfirewalllogtransformxss
appfirewalllogsql	<Double>	Read-only	Number of HTML SQL Injection security check log messages generated by the Application Firewall.
appfirewalllogsqlrate	<Double>	Read-only	Rate (/s) counter for appfirewalllogsql
appfirewalllogtransformsql	<Double>	Read-only	Number of HTML SQL Injection security check transform log messages generated by the Application Firewall.
appfirewalllogtransformsqlrate	<Double>	Read-only	Rate (/s) counter for appfirewalllogtransformsql
appfirewalllogfieldformat	<Double>	Read-only	Number of Field Format security check log messages

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
			generated by the Application Firewall.
appfirewalllogfieldformatrate	<Double>	Read-only	Rate (/s) counter for appfirewalllogfieldformat
appfirewalllogfieldconsistency	<Double>	Read-only	Number of Field Consistency security check log messages generated by the Application Firewall.
appfirewalllogfieldconsistencyrate	<Double>	Read-only	Rate (/s) counter for appfirewalllogfieldconsistency
appfirewalllogcreditcard	<Double>	Read-only	Number of Credit Card security check log messages generated by the Application Firewall.
appfirewalllogcreditcardrate	<Double>	Read-only	Rate (/s) counter for appfirewalllogcreditcard
appfirewalllogsafeobject	<Double>	Read-only	Number of Safe Object security check log messages generated by the Application Firewall.
appfirewalllogsafeobjectrate	<Double>	Read-only	Rate (/s) counter for appfirewalllogsafeobject
appfirewallsignaturelogs	<Double>	Read-only	Number of Signature logs generated by the Application Firewall.

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
appfirewallsignaturelogsrate	<Double>	Read-only	Rate (/s) counter for appfirewallsignaturelogs
appfirewalllogcontenttype	<Double>	Read-only	Number of content type security check log messages generated by the Application Firewall.
appfirewalllogcontenttyperate	<Double>	Read-only	Rate (/s) counter for appfirewalllogcontenttype
appfirewallwellformednesslogs	<Double>	Read-only	Number of XML Format security check log messages generated by the Application Firewall.
appfirewallwellformednesslogsrate	<Double>	Read-only	Rate (/s) counter for appfirewallwellformednesslogs
appfirewallxdoslogs	<Double>	Read-only	Number of XML Denial-of-Service security check log messages generated by the Application Firewall.
appfirewallxdoslogsrate	<Double>	Read-only	Rate (/s) counter for appfirewallxdoslogs
appfirewallmsgvallogs	<Double>	Read-only	Number of XML Message Validation security check log messages generated by the Application Firewall.

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
appfirewallmsgvallogsrate	<Double>	Read-only	Rate (/s) counter for appfirewallmsgvallogs
appfirewallwsilogs	<Double>	Read-only	Number of Web Services Interoperability (WS-I) security check log messages generated by the Application Firewall.
appfirewallwsilogsrate	<Double>	Read-only	Rate (/s) counter for appfirewallwsilogs
appfirewallxmlsqllogs	<Double>	Read-only	Number of XML SQL Injection security check log messages generated by the Application Firewall.
appfirewallxmlsqllogsrate	<Double>	Read-only	Rate (/s) counter for appfirewallxmlsqllogs
appfirewallxmlxsslogs	<Double>	Read-only	Number of XML Cross-Site Scripting (XSS) security check log messages generated by the Application Firewall.
appfirewallxmlxsslogsrate	<Double>	Read-only	Rate (/s) counter for appfirewallxmlxsslogs
appfirewallxmlattachmentlogs	<Double>	Read-only	Number of XML Attachment security check log messages generated by the Application Firewall.

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
appfirewallxmlattachmentlogsrate	<Double>	Read-only	Rate (/s) counter for appfirewallxmlattachmentlogs
appfirewallxmlsoapfaultlogs	<Double>	Read-only	Number of requests generating soap:fault log messages
appfirewallxmlsoapfaultlogsrate	<Double>	Read-only	Rate (/s) counter for appfirewallxmlsoapfaultlogs
appfirewallxmlgenlogs	<Double>	Read-only	Number of requests generating XML generic error log messages
appfirewallxmlgenlogsrate	<Double>	Read-only	Rate (/s) counter for appfirewallxmlgenlogs
appfirewallret4xx	<Double>	Read-only	Number of requests returning HTTP 4xx from the backend server
appfirewallret4xxrate	<Double>	Read-only	Rate (/s) counter for appfirewallret4xx
appfirewallret5xx	<Double>	Read-only	Number of requests returning HTTP 5xx from the backend server
appfirewallret5xxrate	<Double>	Read-only	Rate (/s) counter for appfirewallret5xx

Operations

(click to see Properties)

GET (ALL)

Some options that you can use for each operations:

- **Getting warnings in response:**NITRO allows you to get warnings in an operation by specifying the "warning" query parameter as "yes". For example, to get warnings while connecting to the Citrix ADC appliance, the URL is as follows:

  http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/login?warning=yes

  If any, the warnings are displayed in the response payload with the HTTP code "209 X-NITRO-WARNING".

- **Authenticated access for individual NITRO operations:**NITRO allows you to logon to the Citrix ADC appliance to perform individual operations. You can use this option instead of creating a NITRO session (using the login object) and then using that session to perform all operations,

  To do this, you must specify the username and password in the request header of the NITRO request as follows:

  X-NITRO-USER:<username>

  X-NITRO-PASS:<password>

  Note: In such cases, make sure that the request header DOES not include the following:

  Cookie:NITRO_AUTH_TOKEN=<tokenvalue>

`*Note:`*

Mandatory parameters are marked in redand placeholder content is marked in <green>.

get (all)

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/stat/appfw

Query-parameters:

args

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/stat/appfw?
args=detail:<Boolean_value>,fullvalues:<Boolean_value>,ntimes:<Double_value>,logfile:<String_value

Use this query-parameter to get appfw resources based on additional properties.

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "appfw": [ {

"appfirewallredirectsrate":<Double_value>,

"appfirewallviolcookierate":<Double_value>,

"appfirewallsignaturelogs":<Double_value>,

"appfirewalllogdenyurl":<Double_value>,

"appfirewalllogsqlrate":<Double_value>,

"appfirewallviolfieldconsistency":<Double_value>,

"appfirewallviolbufferoverflowrate":<Double_value>,

"appfirewalllogtransformsql":<Double_value>,

"appfirewallviolxssrate":<Double_value>,

"appfirewalllogtransformsqlrate":<Double_value>,

"appfirewalltotalviol":<Double_value>,

"appfirewallwellformednesslogs":<Double_value>,

"appfirewallviolsafeobjectrate":<Double_value>,
```

```
"appfirewallresponsesrate":<Double_value>,

"appfirewallviolxmlxssviolations":<Double_value>,

"appfirewallresponses":<Double_value>,

"appfirewallviolsignature":<Double_value>,

"appfirewalllogbufferoverflow":<Double_value>,

"appfirewallviolxmlgenviolationsrate":<Double_value>,

"appfirewallviolrate":<Double_value>,

"appfirewallabortsrate":<Double_value>,

"appfirewallviolxmlsoapfaultviolationsrate":<Double_value>,

"appfirewallrequestsrate":<Double_value>,

"appfirewallwsilogs":<Double_value>,

"appfirewallredirects":<Double_value>,

"appfirewallsignaturelogsrate":<Double_value>,

"appfirewallxdoslogsrate":<Double_value>,

"appfirewalllogtransformxss":<Double_value>,

"appfirewallxmlattachmentlogsrate":<Double_value>,

"appfirewalllogrefererheaderrate":<Double_value>,

"appfirewalllogcookie":<Double_value>,

"appfirewallviolcreditcardrate":<Double_value>,

"appfirewallret5xx":<Double_value>,

"appfirewallviolmsgvalviolations":<Double_value>,

"appfirewalllongavgresptime":<Double_value>,
```

```
"appfirewalllogstarturl":<Double_value>,

"appfirewallviolxmlgenviolations":<Double_value>,

"appfirewallviolxmlsoapfaultviolations":<Double_value>,

"appfirewallviolsafeobject":<Double_value>,

"appfirewalllogxssrate":<Double_value>,

"appfirewallviolsql":<Double_value>,

"appfirewallviolxdosviolationsrate":<Double_value>,

"appfirewalllogcreditcardrate":<Double_value>,

"appfirewalllogsql":<Double_value>,

"appfirewalllogdenyurlrate":<Double_value>,

"appfirewallxmlsoapfaultlogs":<Double_value>,

"appfirewalllogcontenttyperate":<Double_value>,

"appfirewallviolxmlattachmentviolationsrate":<Double_value>,

"appfirewalllogbufferoverflowrate":<Double_value>,

"appfirewallret5xxrate":<Double_value>,

"appfirewalltotallog":<Double_value>,

"appfirewalllogcookierate":<Double_value>,

"appfirewalllogcsrftag":<Double_value>,

"appfirewallviolxss":<Double_value>,

"appfirewallviolwellformednessviolations":<Double_value>,

"appfirewallreqbytes":<Double_value>,

"appfirewallxmlattachmentlogs":<Double_value>,
```

```
"appfirewalllogcsrftagrate":<Double_value>,

"appfirewallviolmsgvalviolationsrate":<Double_value>,

"appfirewallviolcontenttyperate":<Double_value>,

"appfirewalllograte":<Double_value>,

"appfirewallxmlsqllogsrate":<Double_value>,

"appfirewallxmlgenlogs":<Double_value>,

"appfirewallreqbytesrate":<Double_value>,

"appfirewalllogsafeobject":<Double_value>,

"appfirewallviolcreditcard":<Double_value>,

"appfirewalllogrefererheader":<Double_value>,

"appfirewallviolxmlsqlviolationsrate":<Double_value>,

"appfirewallaborts":<Double_value>,

"appfirewallviolcsrftagrate":<Double_value>,

"appfirewalllogstarturlrate":<Double_value>,

"appfirewallviolcookie":<Double_value>,

"appfirewallxmlxsslogs":<Double_value>,

"appfirewallmsgvallogs":<Double_value>,

"appfirewallviolwsiviolations":<Double_value>,

"appfirewallviolcsrftag":<Double_value>,

"appfirewallrequests":<Double_value>,

"appfirewallresbytes":<Double_value>,

"appfirewallviolbufferoverflow":<Double_value>,
```

```
"appfirewallviolsignaturerate":<Double_value>,

"appfirewallviolxmlxssviolationsrate":<Double_value>,

"appfirewallviolxdosviolations":<Double_value>,

"appfirewalllogsafeobjectrate":<Double_value>,

"appfirewallxdoslogs":<Double_value>,

"appfirewallviolcontenttype":<Double_value>,

"appfirewallviolfieldconsistencyrate":<Double_value>,

"appfirewalllogtransformxssrate":<Double_value>,

"appfirewalllogfieldformatrate":<Double_value>,

"appfirewallwellformednesslogsrate":<Double_value>,

"appfirewallviolwsiviolationsrate":<Double_value>,

"appfirewalllogcreditcard":<Double_value>,

"appfirewallviolwellformednessviolationsrate":<Double_value>,

"appfirewallviolfieldformatrate":<Double_value>,

"appfirewalllogfieldconsistencyrate":<Double_value>,

"appfirewallvioldenyurl":<Double_value>,

"appfirewallviolxmlattachmentviolations":<Double_value>,

"appfirewallviolsqlrate":<Double_value>,

"appfirewallviolstarturlrate":<Double_value>,

"appfirewallviolxmlsqlviolations":<Double_value>,

"appfirewallshortavgresptime":<Double_value>,

"appfirewallxmlgenlogsrate":<Double_value>,
```

```
"appfirewalllogfieldformat":<Double_value>,

"appfirewallxmlxsslogsrate":<Double_value>,

"appfirewallresbytesrate":<Double_value>,

"appfirewallviolrefererheader":<Double_value>,

"appfirewallviolrefererheaderrate":<Double_value>,

"appfirewalllogxss":<Double_value>,

"appfirewalltrapsdropped":<Double_value>,

"appfirewallviolstarturl":<Double_value>,

"appfirewallret4xxrate":<Double_value>,

"appfirewallxmlsqllogs":<Double_value>,

"appfirewallwsilogsrate":<Double_value>,

"appfirewallxmlsoapfaultlogsrate":<Double_value>,

"appfirewallvioldenyurlrate":<Double_value>,

"appfirewalllogcontenttype":<Double_value>,

"appfirewallmsgvallogsrate":<Double_value>,

"appfirewallret4xx":<Double_value>,

"appfirewallviolfieldformat":<Double_value>,

"appfirewalllogfieldconsistency":<Double_value>

}]}
```

appfwpolicy

Statistics for application firewall policy resource.

Properties

(click to see Operations)

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
name	<String>	Read-write	Name of the application firewall policy. Minimum length = 1
clearstats	<String>	Read-write	Clear the statsistics / counters. Possible values = basic, full
pipolicyhits	<Double>	Read-only	Number of hits on the policy
pipolicyhitsrate	<Double>	Read-only	Rate (/s) counter for pipolicyhits
pipolicyundefhits	<Double>	Read-only	Number of undef hits on the policy
pipolicyundefhitsrate	<Double>	Read-only	Rate (/s) counter for pipolicyundefhits

Operations

(click to see Properties)

GET (ALL)| GET

Some options that you can use for each operations:

- **Getting warnings in response:**NITRO allows you to get warnings in an operation by specifying the "warning" query parameter as "yes". For example, to get warnings while connecting to the Citrix ADC appliance, the URL is as follows:

  http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/login?warning=yes

  If any, the warnings are displayed in the response payload with the HTTP code "209 X-NITRO-WARNING".

- **Authenticated access for individual NITRO operations:**NITRO allows you to logon to the Citrix ADC appliance to perform individual operations. You can use this option instead of creating a NITRO session (using the login object) and then using that session to perform all operations,

  To do this, you must specify the username and password in the request header of the NITRO request as follows:

  X-NITRO-USER:<username>

  X-NITRO-PASS:<password>

  Note: In such cases, make sure that the request header DOES not include the following:

  Cookie:NITRO_AUTH_TOKEN=<tokenvalue>

`*Note:`*

Mandatory parameters are marked in redand placeholder content is marked in <green>.

get (all)

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/stat/appfwpolicy

Query-parameters:

args

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/stat/appfwpolicy?
args=name:<String_value>,detail:<Boolean_value>,fullvalues:<Boolean_value>,ntimes:<Double_value>

Use this query-parameter to get appfwpolicy resources based on additional properties.

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "appfwpolicy": [ {

"name":<String_value>,

"pipolicyundefhitsrate":<Double_value>,

"pipolicyundefhits":<Double_value>,

"pipolicyhitsrate":<Double_value>,

"pipolicyhits":<Double_value>

}]}
```

get

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/stat/appfwpolicy/name_value><String>

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "appfwpolicy": [ {

"name":<String_value>,
```

```
"pipolicyundefhitsrate":<Double_value>,

"pipolicyundefhits":<Double_value>,

"pipolicyhitsrate":<Double_value>,

"pipolicyhits":<Double_value>

}]}
```

appfwpolicylabel

Statistics for application firewall policy label resource.

Properties

(click to see Operations)

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
labelname	<String>	Read-write	Name of the application firewall policy label. Minimum length = 1
clearstats	<String>	Read-write	Clear the statsistics / counters. Possible values = basic, full
pipolicylabelhits	<Double>	Read-only	Number of times policy label was invoked.
pipolicylabelhitsrate	<Double>	Read-only	Rate (/s) counter for pipolicylabelhits

Operations

(click to see Properties)

GET (ALL)| GET

Some options that you can use for each operations:

- **Getting warnings in response:**NITRO allows you to get warnings in an operation by specifying the "warning" query parameter as "yes". For example, to get warnings while connecting to the Citrix ADC appliance, the URL is as follows:

  http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/login?warning=yes

  If any, the warnings are displayed in the response payload with the HTTP code "209 X-NITRO-WARNING".

- **Authenticated access for individual NITRO operations:**NITRO allows you to logon to the Citrix ADC appliance to perform individual operations. You can use this option instead of creating a NITRO session (using the login object) and then using that session to perform all operations,

  To do this, you must specify the username and password in the request header of the NITRO request as follows:

  X-NITRO-USER:<username>

  X-NITRO-PASS:<password>

  Note: In such cases, make sure that the request header DOES not include the following:

  Cookie:NITRO_AUTH_TOKEN=<tokenvalue>

`*Note:`*

Mandatory parameters are marked in redand placeholder content is marked in <green>.

get (all)

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/stat/appfwpolicylabel

Query-parameters:

args

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/stat/appfwpolicylabel?
args=labelname:<String_value>,detail:<Boolean_value>,fullvalues:<Boolean_value>,ntimes:<Double_va

Use this query-parameter to get appfwpolicylabel resources based on additional properties.

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "appfwpolicylabel": [ {

"labelname":<String_value>,

"pipolicylabelhitsrate":<Double_value>,

"pipolicylabelhits":<Double_value>

}]}
```

get

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/stat/appfwpolicylabel/labelname_value><String>

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "appfwpolicylabel": [ {

"labelname":<String_value>,

"pipolicylabelhitsrate":<Double_value>,

"pipolicylabelhits":<Double_value>

}]}
```

appfwprofile

Statistics for application firewall profile resource.

Properties

(click to see Operations)

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
name	<String>	Read-write	Name of the application firewall profile. Minimum length = 1
clearstats	<String>	Read-write	Clear the statsistics / counters. Possible values = basic, full

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
appfirewallrequestsperprofile	<Double>	Read-only	HTTP/HTTPS requests sent to your protected web servers via the Application Firewall.
appfirewallrequestsperprofilerate	<Double>	Read-only	Rate (/s) counter for appfirewallrequestsperprofile
appfirewallreqbytesperprofile	<Double>	Read-only	Number of bytes transfered for requests
appfirewallreqbytesperprofilerate	<Double>	Read-only	Rate (/s) counter for appfirewallreqbytesperprofile
appfirewallresponsesperprofile	<Double>	Read-only	HTTP/HTTPS responses sent by your protected web servers via the Application Firewall.
appfirewallresponsesperprofilerate	<Double>	Read-only	Rate (/s) counter for appfirewallresponsesperprofile
appfirewallresbytesperprofile	<Double>	Read-only	Number of bytes transfered for responses
appfirewallresbytesperprofilerate	<Double>	Read-only	Rate (/s) counter for appfirewallresbytesperprofile
appfirewallabortsperprofile	<Double>	Read-only	Incomplete HTTP/HTTPS requests aborted by the client before the Application Firewall

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
			could finish processing them.
appfirewallabortsperprofilerate	<Double>	Read-only	Rate (/s) counter for appfirewallabortsperprofile
appfirewallredirectsperprofile	<Double>	Read-only	HTTP/HTTPS requests redirected by the Application Firewall to a different Web page or web server. (HTTP 302)
appfirewallredirectsperprofilerate	<Double>	Read-only	Rate (/s) counter for appfirewallredirectsperprofile
appfirewalllongavgresptimeperprofile	<Double>	Read-only	Average backend response time in milliseconds since reboot
appfirewallshortavgresptimeperprofile	<Double>	Read-only	Average backend response time in milliseconds over the last 7 seconds
appfirewallviolstarturlperprofile	<Double>	Read-only	Number of Start URL security check violations seen by the Application Firewall.
appfirewallviolstarturlperprofilerate	<Double>	Read-only	Rate (/s) counter for appfirewallviolstarturlperprofile
appfirewallvioldenyurlperprofile	<Double>	Read-only	Number of Deny URL security check violations seen by

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
			the Application Firewall.
appfirewallvioldenyurlperprofilerate	<Double>	Read-only	Rate (/s) counter for appfirewallvioldenyurlperprofile
appfirewallviolreferer headerperprofile	<Double>	Read-only	Number of Referer Header security check violations seen by the Application Firewall.
appfirewallviolreferer headerperprofilerate	<Double>	Read-only	Rate (/s) counter for appfirewallviolreferer headerperprofile
appfirewallviolbuffero verflowperprofile	<Double>	Read-only	Number of Buffer Overflow security check violations seen by the Application Firewall.
appfirewallviolbuffero verflowperprofilerate	<Double>	Read-only	Rate (/s) counter for appfirewallviolbuffero verflowperprofile
appfirewallviolcookie perprofile	<Double>	Read-only	Number of Cookie Consistency security check violations seen by the Application Firewall.
appfirewallviolcookie perprofilerate	<Double>	Read-only	Rate (/s) counter for appfirewallviolcookie perprofile
appfirewallviolcsrftag perprofile	<Double>	Read-only	Number of Cross Site Request Forgery form tag security check violations

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
			seen by the Application Firewall.
appfirewallviolcsrftagperprofilerate	<Double>	Read-only	Rate (/s) counter for appfirewallviolcsrftagperprofile
appfirewallviolxsspperprofile	<Double>	Read-only	Number of HTML Cross-Site Scripting security check violations seen by the Application Firewall.
appfirewallviolxsspperprofilerate	<Double>	Read-only	Rate (/s) counter for appfirewallviolxsspperprofile
appfirewallviolsqlperprofile	<Double>	Read-only	Number of HTML SQL Injection security check violations seen by the Application Firewall.
appfirewallviolsqlperprofilerate	<Double>	Read-only	Rate (/s) counter for appfirewallviolsqlperprofile
appfirewallviolfieldformatperprofile	<Double>	Read-only	Number of Field Format security check violations seen by the Application Firewall.
appfirewallviolfieldformatperprofilerate	<Double>	Read-only	Rate (/s) counter for appfirewallviolfieldformatperprofile
appfirewallviolfieldconsistencyperprofile	<Double>	Read-only	Number of Field Consistency security check violations

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
			seen by the Application Firewall.
appfirewallviolfieldconsistencyperprofilerate	<Double>	Read-only	Rate (/s) counter for appfirewallviolfieldconsistencyperprofile
appfirewallviolcreditcardperprofile	<Double>	Read-only	Number of Credit Card security check violations seen by the Application Firewall.
appfirewallviolcreditcardperprofilerate	<Double>	Read-only	Rate (/s) counter for appfirewallviolcreditcardperprofile
appfirewallviolsafeobjectperprofile	<Double>	Read-only	Number of Safe Object security check violations seen by the Application Firewall.
appfirewallviolsafeobjectperprofilerate	<Double>	Read-only	Rate (/s) counter for appfirewallviolsafeobjectperprofile
appfirewallviolsignatureperprofile	<Double>	Read-only	Number of Signature violations seen by the Application Firewall.
appfirewallviolsignatureperprofilerate	<Double>	Read-only	Rate (/s) counter for appfirewallviolsignatureperprofile
appfirewallviolcontenttypeperprofile	<Double>	Read-only	Number of Content Type security check violations seen by the Application Firewall.

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
appfirewallviolcontenttypeperprofilerate	<Double>	Read-only	Rate (/s) counter for appfirewallviolcontenttypeperprofile
appfirewallviolwellformednessviolationsperprofile	<Double>	Read-only	Number of XML Format security check violations seen by the Application Firewall.
appfirewallviolwellformednessviolationsperprofilerate	<Double>	Read-only	Rate (/s) counter for appfirewallviolwellformednessviolationsperprofile
appfirewallviolxdosviolationsperprofile	<Double>	Read-only	Number of XML Denial-of-Service security check violations seen by the Application Firewall.
appfirewallviolxdosviolationsperprofilerate	<Double>	Read-only	Rate (/s) counter for appfirewallviolxdosviolationsperprofile
appfirewallviolmsgvalviolationsperprofile	<Double>	Read-only	Number of XML Message Validation security check violations seen by the Application Firewall.
appfirewallviolmsgvalviolationsperprofilerate	<Double>	Read-only	Rate (/s) counter for appfirewallviolmsgvalviolationsperprofile
appfirewallviolwsiviolationsperprofile	<Double>	Read-only	Number of Web Services Interoperability (WS-I) security check

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
			violations seen by the Application Firewall.
appfirewallviolwsiviolationsperprofilerate	<Double>	Read-only	Rate (/s) counter for appfirewallviolwsiviolationsperprofile
appfirewallviolxmlsqlviolationsperprofile	<Double>	Read-only	Number of XML SQL Injection security check violations seen by the Application Firewall.
appfirewallviolxmlsqlviolationsperprofilerate	<Double>	Read-only	Rate (/s) counter for appfirewallviolxmlsqlviolationsperprofile
appfirewallviolxmlxssviolationsperprofile	<Double>	Read-only	Number of XML Cross-Site Scripting (XSS) security check violations seen by the Application Firewall.
appfirewallviolxmlxssviolationsperprofilerate	<Double>	Read-only	Rate (/s) counter for appfirewallviolxmlxssviolationsperprofile
appfirewallviolxmlattachmentviolationsperprofile	<Double>	Read-only	Number of XML Attachment security check violations seen by the Application Firewall.
appfirewallviolxmlattachmentviolationsperprofilerate	<Double>	Read-only	Rate (/s) counter for appfirewallviolxmlattachmentviolationsperprofile

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
appfirewallviolxmlsoapfaultviolationsperprofile	<Double>	Read-only	Number of requests returning soap:fault from the backend server
appfirewallviolxmlsoapfaultviolationsperprofilerate	<Double>	Read-only	Rate (/s) counter for appfirewallviolxmlsoapfaultviolationsperprofile
appfirewallviolxmlgenericviolationsperprofile	<Double>	Read-only	Number of requests returning XML generic violation from the backend server
appfirewallviolxmlgenericviolationsperprofilerate	<Double>	Read-only	Rate (/s) counter for appfirewallviolxmlgenericviolationsperprofile
appfirewalltotalviolperprofile	<Double>	Read-only	Number of violations seen by the application firewall on per profile basis
appfirewallviolperprofilerate	<Double>	Read-only	Rate (/s) counter for appfirewalltotalviolperprofile
appfirewalllogstarturlperprofile	<Double>	Read-only	Number of Start URL security check log messages generated by the Application Firewall.
appfirewalllogstarturlperprofilerate	<Double>	Read-only	Rate (/s) counter for appfirewalllogstarturlperprofile

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
appfirewalllogdenyurl perprofile	<Double>	Read-only	Number of Deny URL security check log messages generated by the Application Firewall.
appfirewalllogdenyurl perprofilerate	<Double>	Read-only	Rate (/s) counter for appfirewalllogdenyurl perprofile
appfirewalllogreferer headerperprofile	<Double>	Read-only	Number of Referer Header security check log messages generated by the Application Firewall.
appfirewalllogreferer headerperprofilerate	<Double>	Read-only	Rate (/s) counter for appfirewalllogreferer headerperprofile
appfirewalllogbuffero verflowperprofile	<Double>	Read-only	Number of Buffer Overflow security check log messages generated by the Application Firewall.
appfirewalllogbuffero verflowperprofilerate	<Double>	Read-only	Rate (/s) counter for appfirewalllogbuffero verflowperprofile
appfirewalllogcookie perprofile	<Double>	Read-only	Number of Cookie Consistency security check log messages generated by the Application Firewall.
appfirewalllogcookie perprofilerate	<Double>	Read-only	Rate (/s) counter for appfirewalllogcookie perprofile

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
appfirewalllogcsrftagperprofile	<Double>	Read-only	Number of Cross Site Request Forgery form tag security check log messages generated by the Application Firewall.
appfirewalllogcsrftagperprofilerate	<Double>	Read-only	Rate (/s) counter for appfirewalllogcsrftagperprofile
appfirewalllogxsspperprofile	<Double>	Read-only	Number of HTML Cross-Site Scripting security check log messages generated by the Application Firewall.
appfirewalllogxsspperprofilerate	<Double>	Read-only	Rate (/s) counter for appfirewalllogxsspperprofile
appfirewalllogxformxssperprofile	<Double>	Read-only	Number of HTML Cross-Site Scripting security check transform log messages generated by the Application Firewall.
appfirewalllogxformxssperprofilerate	<Double>	Read-only	Rate (/s) counter for appfirewalllogxformxssperprofile
appfirewalllogsqlperprofile	<Double>	Read-only	Number of HTML SQL Injection security check log messages generated by the Application Firewall.

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
appfirewalllogsqlperprofilerate	<Double>	Read-only	Rate (/s) counter for appfirewalllogsqlperprofile
appfirewalllogxformsqlperprofile	<Double>	Read-only	Number of HTML SQL Injection security check transform log messages generated by the Application Firewall.
appfirewalllogxformsqlperprofilerate	<Double>	Read-only	Rate (/s) counter for appfirewalllogxformsqlperprofile
appfirewalllogfieldformatperprofile	<Double>	Read-only	Number of Field Format security check log messages generated by the Application Firewall.
appfirewalllogfieldformatperprofilerate	<Double>	Read-only	Rate (/s) counter for appfirewalllogfieldformatperprofile
appfirewalllogfieldconsistencyperprofile	<Double>	Read-only	Number of Field Consistency security check log messages generated by the Application Firewall.
appfirewalllogfieldconsistencyperprofilerate	<Double>	Read-only	Rate (/s) counter for appfirewalllogfieldconsistencyperprofile
appfirewalllogcreditcardperprofile	<Double>	Read-only	Number of Credit Card security check log messages generated by the Application Firewall.

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
appfirewalllogcreditcardperprofilerate	<Double>	Read-only	Rate (/s) counter for appfirewalllogcreditcardperprofile
appfirewallxformlogcreditcardperprofile	<Double>	Read-only	Number of Credit Card security check transform log messages generated by the Application Firewall.
appfirewallxformlogcreditcardperprofilerate	<Double>	Read-only	Rate (/s) counter for appfirewallxformlogcreditcardperprofile
appfirewalllogsafeobjectperprofile	<Double>	Read-only	Number of Safe Object security check log messages generated by the Application Firewall.
appfirewalllogsafeobjectperprofilerate	<Double>	Read-only	Rate (/s) counter for appfirewalllogsafeobjectperprofile
appfirewalllogcontenttypeperprofile	<Double>	Read-only	Number of Content type security check log messages generated by the Application Firewall.
appfirewalllogcontenttypeperprofilerate	<Double>	Read-only	Rate (/s) counter for appfirewalllogcontenttypeperprofile
appfirewallwellformednesslogsperprofile	<Double>	Read-only	Number of XML Format security check log messages generated by the Application Firewall.

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
appfirewallwellformednesslogsperprofilerate	<Double>	Read-only	Rate (/s) counter for appfirewallwellformednesslogsperprofile
appfirewallxdoslogsperprofile	<Double>	Read-only	Number of XML Denial-of-Service security check log messages generated by the Application Firewall.
appfirewallxdoslogsperprofilerate	<Double>	Read-only	Rate (/s) counter for appfirewallxdoslogsperprofile
appfirewallmsgvallogsperprofile	<Double>	Read-only	Number of XML Message Validation security check log messages generated by the Application Firewall.
appfirewallmsgvallogsperprofilerate	<Double>	Read-only	Rate (/s) counter for appfirewallmsgvallogsperprofile
appfirewallwsilogsperprofile	<Double>	Read-only	Number of Web Services Interoperability (WS-I) security check log messages generated by the Application Firewall.
appfirewallwsilogsperprofilerate	<Double>	Read-only	Rate (/s) counter for appfirewallwsilogsperprofile
appfirewallxmlsqllogsperprofile	<Double>	Read-only	Number of XML SQL Injection security check log messages

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
			generated by the Application Firewall.
appfirewallxmlsqllogsperprofilerate	<Double>	Read-only	Rate (/s) counter for appfirewallxmlsqllogsperprofile
appfirewallxmlxsslogsperprofile	<Double>	Read-only	Number of XML Cross-Site Scripting (XSS) security check log messages generated by the Application Firewall.
appfirewallxmlxsslogsperprofilerate	<Double>	Read-only	Rate (/s) counter for appfirewallxmlxsslogsperprofile
appfirewallxmlattachmentlogsperprofile	<Double>	Read-only	Number of XML Attachment security check log messages generated by the Application Firewall.
appfirewallxmlattachmentlogsperprofilerate	<Double>	Read-only	Rate (/s) counter for appfirewallxmlattachmentlogsperprofile
appfirewallxmlsoapfaultlogsperprofile	<Double>	Read-only	Number of requests generating soap:fault log messages
appfirewallxmlsoapfaultlogsperprofilerate	<Double>	Read-only	Rate (/s) counter for appfirewallxmlsoapfaultlogsperprofile
appfirewallxmlgenericlogsperprofile	<Double>	Read-only	Number of requests generating XML Generic log messages

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
appfirewallxmlgenericlogsperprofilerate	<Double>	Read-only	Rate (/s) counter for appfirewallxmlgenericlogsperprofile
appfirewalltotallogperprofile	<Double>	Read-only	Number of log messages generated by the application firewall on per profile basis
appfirewalllogperprofilerate	<Double>	Read-only	Rate (/s) counter for appfirewalltotallogperprofile
appfirewallret4xxperprofile	<Double>	Read-only	Number of requests returning HTTP 4xx from the backend server
appfirewallret4xxperprofilerate	<Double>	Read-only	Rate (/s) counter for appfirewallret4xxperprofile
appfirewallret5xxperprofile	<Double>	Read-only	Number of requests returning HTTP 5xx from the backend server
appfirewallret5xxperprofilerate	<Double>	Read-only	Rate (/s) counter for appfirewallret5xxperprofile

Operations

(click to see Properties)

GET (ALL)| GET

Some options that you can use for each operations:

- **Getting warnings in response:**NITRO allows you to get warnings in an operation by specifying the "warning" query parameter as "yes". For example, to get warnings while connecting to the Citrix ADC appliance, the URL is as follows:

  http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/login?warning=yes

  If any, the warnings are displayed in the response payload with the HTTP code "209 X-NITRO-WARNING".

- **Authenticated access for individual NITRO operations:**NITRO allows you to logon to the Citrix ADC appliance to perform individual operations. You can use this option instead of creating a NITRO session (using the login object) and then using that session to perform all operations,

  To do this, you must specify the username and password in the request header of the NITRO request as follows:

  X-NITRO-USER:<username>

  X-NITRO-PASS:<password>

  Note: In such cases, make sure that the request header DOES not include the following:

  Cookie:NITRO_AUTH_TOKEN=<tokenvalue>

`*Note:`*

Mandatory parameters are marked in redand placeholder content is marked in <green>.

get (all)

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/stat/appfwprofile

Query-parameters:

args

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/stat/appfwprofile?
args=name:<String_value>,detail:<Boolean_value>,fullvalues:<Boolean_value>,ntimes:<Double_value>

Use this query-parameter to get appfwprofile resources based on additional properties.

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "appfwprofile": [ {

"name":<String_value>,

"appfirewallxmlxsslogsperprofilerate":<Double_value>,

"appfirewallviolmsgvalviolationsperprofile":<Double_value>,

"appfirewalllogcontenttypeperprofile":<Double_value>,

"appfirewallviolperprofilerate":<Double_value>,

"appfirewallviolstarturlperprofile":<Double_value>,

"appfirewalltotalviolperprofile":<Double_value>,

"appfirewallret5xxperprofile":<Double_value>,

"appfirewallviolcontenttypeperprofile":<Double_value>,

"appfirewallredirectsperprofile":<Double_value>,

"appfirewallresponsesperprofile":<Double_value>,

"appfirewallxmlsqllogsperprofile":<Double_value>,

"appfirewallwsilogsperprofile":<Double_value>,

"appfirewallrequestsperprofilerate":<Double_value>,
```

```
"appfirewallviolxmlsoapfaultviolationsperprofile":<Double_value>,

"appfirewalllogbufferoverflowperprofile":<Double_value>,

"appfirewallviolwsiviolationsperprofilerate":<Double_value>,

"appfirewallviolwellformednessviolationsperprofile":<Double_value>,

"appfirewallviolfieldformatperprofile":<Double_value>,

"appfirewallviolcookieperprofilerate":<Double_value>,

"appfirewallresbytesperprofilerate":<Double_value>,

"appfirewalllogxformxssperprofile":<Double_value>,

"appfirewalllogcookieperprofile":<Double_value>,

"appfirewallviolcreditcardperprofilerate":<Double_value>,

"appfirewallxmlsoapfaultlogsperprofilerate":<Double_value>,

"appfirewalllogfieldconsistencyperprofilerate":<Double_value>,

"appfirewalllogdenyurlperprofile":<Double_value>,

"appfirewallresbytesperprofile":<Double_value>,

"appfirewallviolxssperprofile":<Double_value>,

"appfirewallmsgvallogsperprofilerate":<Double_value>,

"appfirewallviolxmlsoapfaultviolationsperprofilerate":<Double_value>,

"appfirewalllogcsrftagperprofile":<Double_value>,

"appfirewallviolcontenttypeperprofilerate":<Double_value>,

"appfirewallxmlsoapfaultlogsperprofile":<Double_value>,

"appfirewallxformlogcreditcardperprofilerate":<Double_value>,

"appfirewallreqbytesperprofilerate":<Double_value>,
```

```
"appfirewallviolmsgvalviolationsperprofilerate":<Double_value>,

"appfirewallxdoslogsperprofile":<Double_value>,

"appfirewallviolfieldconsistencyperprofilerate":<Double_value>,

"appfirewalllogrefererheaderperprofilerate":<Double_value>,

"appfirewallxmlattachmentlogsperprofilerate":<Double_value>,

"appfirewallviolxmlxssviolationsperprofilerate":<Double_value>,

"appfirewallxdoslogsperprofilerate":<Double_value>,

"appfirewallxmlxsslogsperprofile":<Double_value>,

"appfirewallxmlsqllogsperprofilerate":<Double_value>,

"appfirewallrequestsperprofile":<Double_value>,

"appfirewallviolsignatureperprofile":<Double_value>,

"appfirewallviolxmlattachmentviolationsperprofilerate":<Double_value>,

"appfirewallviolsqlperprofilerate":<Double_value>,

"appfirewallviolcsrftagperprofilerate":<Double_value>,

"appfirewallviolstarturlperprofilerate":<Double_value>,

"appfirewalllogxformxssperprofilerate":<Double_value>,

"appfirewallviolxmlsqlviolationsperprofile":<Double_value>,

"appfirewalllogsqlperprofile":<Double_value>,

"appfirewallvioldenyurlperprofile":<Double_value>,

"appfirewallredirectsperprofilerate":<Double_value>,

"appfirewalllogstarturlperprofilerate":<Double_value>,

"appfirewallviolcsrftagperprofile":<Double_value>,
```

```
"appfirewallwellformednesslogsperprofile":<Double_value>,

"appfirewalllogdenyurlperprofilerate":<Double_value>,

"appfirewallviolwsiviolationsperprofile":<Double_value>,

"appfirewallshortavgresptimeperprofile":<Double_value>,

"appfirewallviolsafeobjectperprofile":<Double_value>,

"appfirewallresponsesperprofilerate":<Double_value>,

"appfirewallviolrefererheaderperprofilerate":<Double_value>,

"appfirewallvioldenyurlperprofilerate":<Double_value>,

"appfirewallabortsperprofile":<Double_value>,

"appfirewallviolcookieperprofile":<Double_value>,

"appfirewalllogcreditcardperprofile":<Double_value>,

"appfirewalllogbufferoverflowperprofilerate":<Double_value>,

"appfirewallxmlgenericlogsperprofilerate":<Double_value>,

"appfirewallret4xxperprofile":<Double_value>,

"appfirewallviolfieldformatperprofilerate":<Double_value>,

"appfirewalllogxformsqlperprofile":<Double_value>,

"appfirewallviolbufferoverflowperprofilerate":<Double_value>,

"appfirewallmsgvallogsperprofile":<Double_value>,

"appfirewallwsilogsperprofilerate":<Double_value>,

"appfirewallviolsqlperprofile":<Double_value>,

"appfirewallviolxssperprofilerate":<Double_value>,

"appfirewallxmlgenericlogsperprofile":<Double_value>,
```

```
"appfirewalllogperprofilerate":<Double_value>,

"appfirewallviolcreditcardperprofile":<Double_value>,

"appfirewallxmlattachmentlogsperprofile":<Double_value>,

"appfirewallviolfieldconsistencyperprofile":<Double_value>,

"appfirewalllogcsrftagperprofilerate":<Double_value>,

"appfirewallret4xxperprofilerate":<Double_value>,

"appfirewalllogfieldformatperprofilerate":<Double_value>,

"appfirewallviolsafeobjectperprofilerate":<Double_value>,

"appfirewalllogxssperprofile":<Double_value>,

"appfirewallviolxmlgenericviolationsperprofile":<Double_value>,

"appfirewallviolsignatureperprofilerate":<Double_value>,

"appfirewalllongavgresptimeperprofile":<Double_value>,

"appfirewalllogfieldformatperprofile":<Double_value>,

"appfirewallabortsperprofilerate":<Double_value>,

"appfirewalllogcreditcardperprofilerate":<Double_value>,

"appfirewalllogsqlperprofilerate":<Double_value>,

"appfirewallviolbufferoverflowperprofile":<Double_value>,

"appfirewalllogsafeobjectperprofilerate":<Double_value>,

"appfirewalllogfieldconsistencyperprofile":<Double_value>,

"appfirewalllogstarturlperprofile":<Double_value>,

"appfirewallviolxmlsqlviolationsperprofilerate":<Double_value>,

"appfirewalllogxssperprofilerate":<Double_value>,
```

```
"appfirewalllogcontenttypeperprofilerate":<Double_value>,

"appfirewallwellformednesslogsperprofilerate":<Double_value>,

"appfirewallreqbytesperprofile":<Double_value>,

"appfirewalllogsafeobjectperprofile":<Double_value>,

"appfirewallviolxmlgenericviolationsperprofilerate":<Double_value>,

"appfirewallviolxmlattachmentviolationsperprofile":<Double_value>,

"appfirewalllogrefererheaderperprofile":<Double_value>,

"appfirewallviolxdosviolationsperprofilerate":<Double_value>,

"appfirewallviolwellformednessviolationsperprofilerate":<Double_value>,

"appfirewallviolxdosviolationsperprofile":<Double_value>,

"appfirewalllogxformsqlperprofilerate":<Double_value>,

"appfirewallviolxmlxssviolationsperprofile":<Double_value>,

"appfirewalltotallogperprofile":<Double_value>,

"appfirewalllogcookieperprofilerate":<Double_value>,

"appfirewallret5xxperprofilerate":<Double_value>,

"appfirewallviolrefererheaderperprofile":<Double_value>,

"appfirewallxformlogcreditcardperprofile":<Double_value>

}]}
```

get

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/stat/appfwprofile/name_value><String>

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "appfwprofile": [ {

"name":<String_value>,

"appfirewallxmlxsslogsperprofilerate":<Double_value>,

"appfirewallviolmsgvalviolationsperprofile":<Double_value>,

"appfirewalllogcontenttypeperprofile":<Double_value>,

"appfirewallviolperprofilerate":<Double_value>,

"appfirewallviolstarturlperprofile":<Double_value>,

"appfirewalltotalviolperprofile":<Double_value>,

"appfirewallret5xxperprofile":<Double_value>,

"appfirewallviolcontenttypeperprofile":<Double_value>,

"appfirewallredirectsperprofile":<Double_value>,

"appfirewallresponsesperprofile":<Double_value>,

"appfirewallxmlsqllogsperprofile":<Double_value>,

"appfirewallwsilogsperprofile":<Double_value>,
```

```
"appfirewallrequestsperprofilerate":<Double_value>,

"appfirewallviolxmlsoapfaultviolationsperprofile":<Double_value>,

"appfirewalllogbufferoverflowperprofile":<Double_value>,

"appfirewallviolwsiviolationsperprofilerate":<Double_value>,

"appfirewallviolwellformednessviolationsperprofile":<Double_value>,

"appfirewallviolfieldformatperprofile":<Double_value>,

"appfirewallviolcookieperprofilerate":<Double_value>,

"appfirewallresbytesperprofilerate":<Double_value>,

"appfirewalllogxformxssperprofile":<Double_value>,

"appfirewalllogcookieperprofile":<Double_value>,

"appfirewallviolcreditcardperprofilerate":<Double_value>,

"appfirewallxmlsoapfaultlogsperprofilerate":<Double_value>,

"appfirewalllogfieldconsistencyperprofilerate":<Double_value>,

"appfirewalllogdenyurlperprofile":<Double_value>,

"appfirewallresbytesperprofile":<Double_value>,

"appfirewallviolxssperprofile":<Double_value>,

"appfirewallmsgvallogsperprofilerate":<Double_value>,

"appfirewallviolxmlsoapfaultviolationsperprofilerate":<Double_value>,

"appfirewalllogcsrftagperprofile":<Double_value>,

"appfirewallviolcontenttypeperprofilerate":<Double_value>,

"appfirewallxmlsoapfaultlogsperprofile":<Double_value>,

"appfirewallxformlogcreditcardperprofilerate":<Double_value>,
```

```
"appfirewallreqbytesperprofilerate":<Double_value>,

"appfirewallviolmsgvalviolationsperprofilerate":<Double_value>,

"appfirewallxdoslogsperprofile":<Double_value>,

"appfirewallviolfieldconsistencyperprofilerate":<Double_value>,

"appfirewalllogrefererheaderperprofilerate":<Double_value>,

"appfirewallxmlattachmentlogsperprofilerate":<Double_value>,

"appfirewallviolxmlxssviolationsperprofilerate":<Double_value>,

"appfirewallxdoslogsperprofilerate":<Double_value>,

"appfirewallxmlxsslogsperprofile":<Double_value>,

"appfirewallxmlsqllogsperprofilerate":<Double_value>,

"appfirewallrequestsperprofile":<Double_value>,

"appfirewallviolsignatureperprofile":<Double_value>,

"appfirewallviolxmlattachmentviolationsperprofilerate":<Double_value>,

"appfirewallviolsqlperprofilerate":<Double_value>,

"appfirewallviolcsrftagperprofilerate":<Double_value>,

"appfirewallviolstarturlperprofilerate":<Double_value>,

"appfirewalllogxformxssperprofilerate":<Double_value>,

"appfirewallviolxmlsqlviolationsperprofile":<Double_value>,

"appfirewalllogsqlperprofile":<Double_value>,

"appfirewallvioldenyurlperprofile":<Double_value>,

"appfirewallredirectsperprofilerate":<Double_value>,

"appfirewalllogstarturlperprofilerate":<Double_value>,
```

```
"appfirewallviolcsrftagperprofile":<Double_value>,

"appfirewallwellformednesslogsperprofile":<Double_value>,

"appfirewalllogdenyurlperprofilerate":<Double_value>,

"appfirewallviolwsiviolationsperprofile":<Double_value>,

"appfirewallshortavgresptimeperprofile":<Double_value>,

"appfirewallviolsafeobjectperprofile":<Double_value>,

"appfirewallresponsesperprofilerate":<Double_value>,

"appfirewallviolrefererheaderperprofilerate":<Double_value>,

"appfirewallvioldenyurlperprofilerate":<Double_value>,

"appfirewallabortsperprofile":<Double_value>,

"appfirewallviolcookieperprofile":<Double_value>,

"appfirewalllogcreditcardperprofile":<Double_value>,

"appfirewalllogbufferoverflowperprofilerate":<Double_value>,

"appfirewallxmlgenericlogsperprofilerate":<Double_value>,

"appfirewallret4xxperprofile":<Double_value>,

"appfirewallviolfieldformatperprofilerate":<Double_value>,

"appfirewalllogxformsqlperprofile":<Double_value>,

"appfirewallviolbufferoverflowperprofilerate":<Double_value>,

"appfirewallmsgvallogsperprofile":<Double_value>,

"appfirewallwsilogsperprofilerate":<Double_value>,

"appfirewallviolsqlperprofile":<Double_value>,

"appfirewallviolxssperprofilerate":<Double_value>,
```

```
"appfirewallxmlgenericlogsperprofile":<Double_value>,

"appfirewalllogperprofilerate":<Double_value>,

"appfirewallviolcreditcardperprofile":<Double_value>,

"appfirewallxmlattachmentlogsperprofile":<Double_value>,

"appfirewallviolfieldconsistencyperprofile":<Double_value>,

"appfirewalllogcsrftagperprofilerate":<Double_value>,

"appfirewallret4xxperprofilerate":<Double_value>,

"appfirewalllogfieldformatperprofilerate":<Double_value>,

"appfirewallviolsafeobjectperprofilerate":<Double_value>,

"appfirewalllogxssperprofile":<Double_value>,

"appfirewallviolxmlgenericviolationsperprofile":<Double_value>,

"appfirewallviolsignatureperprofilerate":<Double_value>,

"appfirewalllongavgresptimeperprofile":<Double_value>,

"appfirewalllogfieldformatperprofile":<Double_value>,

"appfirewallabortsperprofilerate":<Double_value>,

"appfirewalllogcreditcardperprofilerate":<Double_value>,

"appfirewalllogsqlperprofilerate":<Double_value>,

"appfirewallviolbufferoverflowperprofile":<Double_value>,

"appfirewalllogsafeobjectperprofilerate":<Double_value>,

"appfirewalllogfieldconsistencyperprofile":<Double_value>,

"appfirewalllogstarturlperprofile":<Double_value>,

"appfirewallviolxmlsqlviolationsperprofilerate":<Double_value>,
```

```
"appfirewalllogxssperprofilerate":<Double_value>,

"appfirewalllogcontenttypeperprofilerate":<Double_value>,

"appfirewallwellformednesslogsperprofilerate":<Double_value>,

"appfirewallreqbytesperprofile":<Double_value>,

"appfirewalllogsafeobjectperprofile":<Double_value>,

"appfirewallviolxmlgenericviolationsperprofilerate":<Double_value>,

"appfirewallviolxmlattachmentviolationsperprofile":<Double_value>,

"appfirewalllogrefererheaderperprofile":<Double_value>,

"appfirewallviolxdosviolationsperprofilerate":<Double_value>,

"appfirewallviolwellformednessviolationsperprofilerate":<Double_value>,

"appfirewallviolxdosviolationsperprofile":<Double_value>,

"appfirewalllogxformsqlperprofilerate":<Double_value>,

"appfirewallviolxmlxssviolationsperprofile":<Double_value>,

"appfirewalltotallogperprofile":<Double_value>,

"appfirewalllogcookieperprofilerate":<Double_value>,

"appfirewallret5xxperprofilerate":<Double_value>,

"appfirewallviolrefererheaderperprofile":<Double_value>,

"appfirewallxformlogcreditcardperprofile":<Double_value>

}]}
```

appqoe

Statistics for appqoe.

Properties

(click to see Operations)

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
clearstats	<String>	Read-write	Clear the statsistics / counters. Possible values = basic, full
totinmemrsp	<Double>	Read-only	Total in-memory responses sent from NS
inmemrsprate	<Double>	Read-only	Rate (/s) counter for totinmemrsp
totfaultycookies	<Double>	Read-only	Total faulty cookies received
faultycookiesrate	<Double>	Read-only	Rate (/s) counter for totfaultycookies
totvalidcookies	<Double>	Read-only	Total valid cookies received
validcookiesrate	<Double>	Read-only	Rate (/s) counter for totvalidcookies
tothighprireq	<Double>	Read-only	Total Requests served from higher priority queue
highprireqrate	<Double>	Read-only	Rate (/s) counter for tothighprireq
totmediumprireq	<Double>	Read-only	Total Requests served from medium priority queue
mediumprireqrate	<Double>	Read-only	Rate (/s) counter for totmediumprireq

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
totlowprireq	<Double>	Read-only	Total Requests served from low priority queue
lowprireqrate	<Double>	Read-only	Rate (/s) counter for totlowprireq
totlowestprireq	<Double>	Read-only	Total Requests served from surge priority queue
lowestprireqrate	<Double>	Read-only	Rate (/s) counter for totlowestprireq
totaltsvrsubfailed	<Double>	Read-only	Total number of times alternate server substitution failed
tsvrsubfailedrate	<Double>	Read-only	Rate (/s) counter for totaltsvrsubfailed
totdostrig	<Double>	Read-only	Total number of times HDOS condition triggered
dostrigrate	<Double>	Read-only	Rate (/s) counter for totdostrig
totdosqvalidcookies	<Double>	Read-only	Total DOSQ valid cookies received
dosqvalidcookiesrate	<Double>	Read-only	Rate (/s) counter for totdosqvalidcookies
totdoshvalidcookies	<Double>	Read-only	Total DOSH valid cookies received
doshvalidcookiesrate	<Double>	Read-only	Rate (/s) counter for totdoshvalidcookies

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
totsidvalidcookies	<Double>	Read-only	Total SID valid cookies received
sidvalidcookiesrate	<Double>	Read-only	Rate (/s) counter for totsidvalidcookies
totonhvalidcookies	<Double>	Read-only	Total ONH valid cookies received
onhvalidcookiesrate	<Double>	Read-only	Rate (/s) counter for totonhvalidcookies
totpriqvalidcookies	<Double>	Read-only	Total PRIQ valid cookies received
priqvalidcookiesrate	<Double>	Read-only	Rate (/s) counter for totpriqvalidcookies
totdosqfaultycookies	<Double>	Read-only	Total DOSQ faulty cookies received
dosqfaultycookiesrate	<Double>	Read-only	Rate (/s) counter for totdosqfaultycookies
totdoshfaultycookies	<Double>	Read-only	Total DOSH faulty cookies received
doshfaultycookiesrate	<Double>	Read-only	Rate (/s) counter for totdoshfaultycookies
totsidfaultycookies	<Double>	Read-only	Total SID faulty cookies received
sidfaultycookiesrate	<Double>	Read-only	Rate (/s) counter for totsidfaultycookies
totonhfaultycookies	<Double>	Read-only	Total ONH faulty cookies received
onhfaultycookiesrate	<Double>	Read-only	Rate (/s) counter for totonhfaultycookies

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
totpriqfaultycookies	<Double>	Read-only	Total PRIQ faulty cookies received
priqfaultycookiesrate	<Double>	Read-only	Rate (/s) counter for totpriqfaultycookies
totpriembedlinks	<Double>	Read-only	Total requests for valid embedded links
priembedlinksrate	<Double>	Read-only	Rate (/s) counter for totpriembedlinks
totsessreq	<Double>	Read-only	Total valid SIDQ requests within session
sessreqrate	<Double>	Read-only	Rate (/s) counter for totsessreq
totaltcntreq	<Double>	Read-only	Total requests for alternate contents
tcntreqrate	<Double>	Read-only	Rate (/s) counter for totaltcntreq
totgetinmemrsp	<Double>	Read-only	Total in-memory GET responses sent from NS
getinmemrsprate	<Double>	Read-only	Rate (/s) counter for totgetinmemrsp
totpostinmemrsp	<Double>	Read-only	Total in-memory POST responses sent from NS
postinmemrsprate	<Double>	Read-only	Rate (/s) counter for totpostinmemrsp

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
totpostinmemrspbytes	<Double>	Read-only	Total in-memory response bytes sent from NS
postinmemrspbytesrate	<Double>	Read-only	Rate (/s) counter for totpostinmemrspbytes

Operations

(click to see Properties)

GET (ALL)

Some options that you can use for each operations:

- **Getting warnings in response:**NITRO allows you to get warnings in an operation by specifying the "warning" query parameter as "yes". For example, to get warnings while connecting to the Citrix ADC appliance, the URL is as follows:

  http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/login?warning=yes

  If any, the warnings are displayed in the response payload with the HTTP code "209 X-NITRO-WARNING".

- **Authenticated access for individual NITRO operations:**NITRO allows you to logon to the Citrix ADC appliance to perform individual operations. You can use this option instead of creating a NITRO session (using the login object) and then using that session to perform all operations,

  To do this, you must specify the username and password in the request header of the NITRO request as follows:

  X-NITRO-USER:<username>

  X-NITRO-PASS:<password>

  Note: In such cases, make sure that the request header DOES not include the following:

  Cookie:NITRO_AUTH_TOKEN=<tokenvalue>

`*Note:`*

Mandatory parameters are marked in redand placeholder content is marked in <green>.

get (all)

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/stat/appqoe

Query-parameters:

args

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/stat/appqoe?
args=detail:<Boolean_value>,fullvalues:<Boolean_value>,ntimes:<Double_value>,logfile:<String_value

Use this query-parameter to get appqoe resources based on additional properties.

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "appqoe": [ {

"totpriqvalidcookies":<Double_value>,

"totonhvalidcookies":<Double_value>,

"dosqvalidcookiesrate":<Double_value>,

"sidvalidcookiesrate":<Double_value>,

"onhvalidcookiesrate":<Double_value>,
```

```
"totdosqvalidcookies":<Double_value>,

"dostrigrate":<Double_value>,

"totgetinmemrsp":<Double_value>,

"totlowprireq":<Double_value>,

"totdoshfaultycookies":<Double_value>,

"doshfaultycookiesrate":<Double_value>,

"sessreqrate":<Double_value>,

"totpostinmemrspbytes":<Double_value>,

"totpriqfaultycookies":<Double_value>,

"totpostinmemrsp":<Double_value>,

"priembedlinksrate":<Double_value>,

"mediumprireqrate":<Double_value>,

"totdostrig":<Double_value>,

"totvalidcookies":<Double_value>,

"totinmemrsp":<Double_value>,

"totsidfaultycookies":<Double_value>,

"totsidvalidcookies":<Double_value>,

"totdosqfaultycookies":<Double_value>,

"tsvrsubfailedrate":<Double_value>,

"postinmemrsprate":<Double_value>,

"totpriembedlinks":<Double_value>,

"totmediumprireq":<Double_value>,
```

```
"priqvalidcookiesrate":<Double_value>,

"totfaultycookies":<Double_value>,

"doshvalidcookiesrate":<Double_value>,

"sidfaultycookiesrate":<Double_value>,

"highprireqrate":<Double_value>,

"dosqfaultycookiesrate":<Double_value>,

"inmemrsprate":<Double_value>,

"getinmemrsprate":<Double_value>,

"onhfaultycookiesrate":<Double_value>,

"validcookiesrate":<Double_value>,

"totaltcntreq":<Double_value>,

"totsessreq":<Double_value>,

"priqfaultycookiesrate":<Double_value>,

"tcntreqrate":<Double_value>,

"totdoshvalidcookies":<Double_value>,

"lowprireqrate":<Double_value>,

"tothighprireq":<Double_value>,

"totaltsvrsubfailed":<Double_value>,

"totlowestprireq":<Double_value>,

"lowestprireqrate":<Double_value>,

"totonhfaultycookies":<Double_value>,

"faultycookiesrate":<Double_value>,
```

```
"postinmemrspbytesrate":<Double_value>

}]}
```

appqoepolicy

Statistics for AppQoS policy resource.

Properties

(click to see Operations)

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
name	<String>	Read-write	policyName. Minimum length = 1
clearstats	<String>	Read-write	Clear the statsistics / counters. Possible values = basic, full
qosavgserverttfb	<Double>	Read-only	Average Server Time-To-First-Byte in milliseconds calculated for this AppQoE policy.
qosavgserverttfbrate	<Double>	Read-only	Rate (/s) counter for qosavgserverttfb
qosavgserverttlb	<Double>	Read-only	Average Server Time-To-Last-Byte in milliseconds calculated for this AppQoE policy.
qosavgserverttlbrate	<Double>	Read-only	Rate (/s) counter for qosavgserverttlb

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
qosavgclientttlb	<Double>	Read-only	Average Client Time-To-Last-Byte in milliseconds calculated for this AppQoE policy.
qosavgclientttlbrate	<Double>	Read-only	Rate (/s) counter for qosavgclientttlb
totthroughput	<Double>	Read-only	Throughput in Kbps calculated on this AppQoE policy
throughputrate	<Double>	Read-only	Rate (/s) counter for totthroughput
totsvrlinkedto	<Double>	Read-only	Total number of server connections that were established through this AppQoE Policy
svrlinkedtorate	<Double>	Read-only	Rate (/s) counter for totsvrlinkedto
totcltrequests	<Double>	Read-only	Total number of client connections that were requested through this AppQoE Policy
cltrequestsrate	<Double>	Read-only	Rate (/s) counter for totcltrequests
totrequests	<Double>	Read-only	Total number of requests that were requested through this AppQoE policy
requestsrate	<Double>	Read-only	Rate (/s) counter for totrequests

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
totrequestbytes	<Double>	Read-only	Total number of requests bytes that were requested through this AppQoE policy
requestbytesrate	<Double>	Read-only	Rate (/s) counter for totrequestbytes
totresponse	<Double>	Read-only	Total number of responses received by this AppQoE policy
responserate	<Double>	Read-only	Rate (/s) counter for totresponse
totresponsebytes	<Double>	Read-only	Total number of response bytes received by this AppQoE policy
responsebytesrate	<Double>	Read-only	Rate (/s) counter for totresponsebytes
totjssent	<Double>	Read-only	Total number of in-memory responses sent instead of expected responses through this AppQoE policy
jssentrate	<Double>	Read-only	Rate (/s) counter for totjssent
totjsbytessent	<Double>	Read-only	Total bytes of in-memory responses sent through this AppQoE policy

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
jsbytessentrate	<Double>	Read-only	Rate (/s) counter for totjsbytessent
pipolicyhits	<Double>	Read-only	Number of hits on the policy
pipolicyhitsrate	<Double>	Read-only	Rate (/s) counter for pipolicyhits

Operations

(click to see Properties)

GET (ALL)| GET

Some options that you can use for each operations:

- **Getting warnings in response:**NITRO allows you to get warnings in an operation by specifying the "warning" query parameter as "yes". For example, to get warnings while connecting to the Citrix ADC appliance, the URL is as follows:

  http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/login?warning=yes

  If any, the warnings are displayed in the response payload with the HTTP code "209 X-NITRO-WARNING".

- **Authenticated access for individual NITRO operations:**NITRO allows you to logon to the Citrix ADC appliance to perform individual operations. You can use this option instead of creating a NITRO session (using the login object) and then using that session to perform all operations,

  To do this, you must specify the username and password in the request header of the NITRO request as follows:

  X-NITRO-USER:<username>

  X-NITRO-PASS:<password>

  Note: In such cases, make sure that the request header DOES not include the following:

  Cookie:NITRO_AUTH_TOKEN=<tokenvalue>

`*Note:`*

Mandatory parameters are marked in redand placeholder content is marked in <green>.

get (all)

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/stat/appqoepolicy

Query-parameters:

args

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/stat/appqoepolicy?
args=name:<String_value>,detail:<Boolean_value>,fullvalues:<Boolean_value>,ntimes:<Double_value>

Use this query-parameter to get appqoepolicy resources based on additional properties.

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "appqoepolicy": [ {

"name":<String_value>,

"qosavgclientttlbrate":<Double_value>,

"totjssent":<Double_value>,

"responsebytesrate":<Double_value>,

"totthroughput":<Double_value>,
```

```
"requestbytesrate":<Double_value>,

"totjsbytessent":<Double_value>,

"svrlinkedtorate":<Double_value>,

"totresponse":<Double_value>,

"totresponsebytes":<Double_value>,

"jssentrate":<Double_value>,

"qosavgserverttfbrate":<Double_value>,

"pipolicyhits":<Double_value>,

"qosavgclientttlb":<Double_value>,

"totcltrequests":<Double_value>,

"totrequests":<Double_value>,

"qosavgserverttlbrate":<Double_value>,

"totrequestbytes":<Double_value>,

"cltrequestsrate":<Double_value>,

"throughputrate":<Double_value>,

"totsvrlinkedto":<Double_value>,

"qosavgserverttfb":<Double_value>,

"pipolicyhitsrate":<Double_value>,

"jsbytessentrate":<Double_value>,

"responserate":<Double_value>,

"requestsrate":<Double_value>,

"qosavgserverttlb":<Double_value>
```

```
}]}
```

get

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/stat/appqoepolicy/name_value><String>

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "appqoepolicy": [ {

"name":<String_value>,

"qosavgclientttlbrate":<Double_value>,

"totjssent":<Double_value>,

"responsebytesrate":<Double_value>,

"totthroughput":<Double_value>,

"requestbytesrate":<Double_value>,

"totjsbytessent":<Double_value>,

"svrlinkedtorate":<Double_value>,
```

```
"totresponse":<Double_value>,

"totresponsebytes":<Double_value>,

"jssentrate":<Double_value>,

"qosavgserverttfbrate":<Double_value>,

"pipolicyhits":<Double_value>,

"qosavgclientttlb":<Double_value>,

"totcltrequests":<Double_value>,

"totrequests":<Double_value>,

"qosavgserverttlbrate":<Double_value>,

"totrequestbytes":<Double_value>,

"cltrequestsrate":<Double_value>,

"throughputrate":<Double_value>,

"totsvrlinkedto":<Double_value>,

"qosavgserverttfb":<Double_value>,

"pipolicyhitsrate":<Double_value>,

"jsbytessentrate":<Double_value>,

"responserate":<Double_value>,

"requestsrate":<Double_value>,

"qosavgserverttlb":<Double_value>

}]}
```

audit

Statistics for audit.

Properties

(click to see Operations)

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
clearstats	<String>	Read-write	Clear the statsistics / counters. Possible values = basic, full
auditsyslogmsgsent	<Double>	Read-only	Syslog messages sent to the syslog server(s) over UDP.
auditsyslogmsgsentrate	<Double>	Read-only	Rate (/s) counter for auditsyslogmsgsent
auditsyslogmsggen	<Double>	Read-only	Syslog messages about to be sent to the syslog server.
auditsyslogmsggenrate	<Double>	Read-only	Rate (/s) counter for auditsyslogmsggen
auditsyslogmsgsenttcp	<Double>	Read-only	Syslog messages sent to the syslog server(s) over TCP.
auditsyslogmsgsenttcprate	<Double>	Read-only	Rate (/s) counter for auditsyslogmsgsenttcp
auditnsballocfail	<Double>	Read-only	NAT allocation failed.
auditnsballocfailrate	<Double>	Read-only	Rate (/s) counter for auditnsballocfail

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
auditlog32errsyslogallocnsbfail	<Double>	Read-only	Nsb allocation failed.
auditlog32errsyslogallocnsbfailrate	<Double>	Read-only	Rate (/s) counter for auditlog32errsyslogallocnsbfail
auditmemallocfail	<Double>	Read-only	Failures in allocation of Access Gateway context structure. When an Access Gateway session is established, the Citrix ADC creates an internal context structure , which identifies the user and the IP address from which the user has logged in.
auditmemallocfailrate	<Double>	Read-only	Rate (/s) counter for auditmemallocfail
auditportallocfail	<Double>	Read-only	Number of times the Citrix ADC failed to allocate a port when sending a syslog message to the syslog server(s).
auditportallocfailrate	<Double>	Read-only	Rate (/s) counter for auditportallocfail
auditcontextnotfound	<Double>	Read-only	Failures in finding the context structure for an Access Gateway session during attempts to send session-specific audit

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
			messages. During an Access Gateway session, audit messages related to the session are queued up in the auditlog buffer for transmission to the audit log server(s). If the session is killed before the messages are sent, the context structure allocated at session creation is removed. This structure is needed for sending the queued auditlog messages. If it is not found, this counter is incremented.
auditcontextnotfound rate	<Double>	Read-only	Rate (/s) counter for auditcontextnotfound
nsbchainallocfail	<Double>	Read-only	Nsb Chain allocation failed.
nsbchainallocfailrate	<Double>	Read-only	Rate (/s) counter for nsbchainallocfail
clientconnfail	<Double>	Read-only	Failures in establishment of a connection between the Citrix ADC and the auditserver tool (the Citrix ADC's custom logging tool).
clientconnfailrate	<Double>	Read-only	Rate (/s) counter for clientconnfail

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
flushcmdcnt	<Double>	Read-only	Auditlog buffer flushes. In a multiprocessor Citrix ADC, both the main processor and the co-processor can generate auditlog messages and fill up the auditlog buffers. But only the primary processor can free up the buffers by sending auditlog messages to the auditlog server(s). The number of auditlog buffers is fixed. If the co-processor detects that all the auditlog buffers are full, it issues a flush command to the main processor.
flushcmdcntrate	<Double>	Read-only	Rate (/s) counter for flushcmdcnt
systcpconnfail	<Double>	Read-only	Failures in establishment of a connection between the Citrix ADC and the syslog server.
systcpconnfailrate	<Double>	Read-only	Rate (/s) counter for systcpconnfail
logunsentlbsys	<Double>	Read-only	Total auditlog messages which are not delivered to load

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
			balanced syslog servers
logunsentlbsysrate	<Double>	Read-only	Rate (/s) counter for logunsentlbsys
logsdropped	<Double>	Read-only	Total number of log messages dropped by Citrix ADC after max hold limit is reached
logsdroppedrate	<Double>	Read-only	Rate (/s) counter for logsdropped
logsdroppedtxminnsbs	<Double>	Read-only	Total number of log messages dropped by Citrix ADC when NSBQ length is less than TX min NSBs
logsdroppedtxminnsbsrate	<Double>	Read-only	Rate (/s) counter for logsdroppedtxminnsbs

Operations

(click to see Properties)

GET (ALL)

Some options that you can use for each operations:

- **Getting warnings in response:**NITRO allows you to get warnings in an operation by specifying the "warning" query parameter as "yes". For example, to get warnings while connecting to the Citrix ADC appliance, the URL is as follows:

  http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/login?warning=yes

  If any, the warnings are displayed in the response payload with the HTTP code "209 X-NITRO-WARNING".

- **Authenticated access for individual NITRO operations:**NITRO allows you to logon to the Citrix ADC appliance to perform individual operations. You can use this option instead of creating a NITRO session (using the login object) and then using that session to perform all operations,

  To do this, you must specify the username and password in the request header of the NITRO request as follows:

  X-NITRO-USER:<username>

  X-NITRO-PASS:<password>

  Note: In such cases, make sure that the request header DOES not include the following:

  Cookie:NITRO_AUTH_TOKEN=<tokenvalue>

`*Note:`*

Mandatory parameters are marked in redand placeholder content is marked in <green>.

get (all)

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/stat/audit

Query-parameters:

args

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/stat/audit?
args=detail:<Boolean_value>,fullvalues:<Boolean_value>,ntimes:<Double_value>,logfile:<String_value

Use this query-parameter to get audit resources based on additional properties.

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:


```
{ "audit": [ {

"auditportallocfailrate":<Double_value>,

"auditnsballocfailrate":<Double_value>,

"auditsyslogmsgsenttcprate":<Double_value>,

"auditcontextnotfoundrate":<Double_value>,

"logsdroppedtxminnsbs":<Double_value>,

"clientconnfail":<Double_value>,

"flushcmdcntrate":<Double_value>,

"auditlog32errsyslogallocnsbfailrate":<Double_value>,

"auditnsballocfail":<Double_value>,

"auditsyslogmsgsentrate":<Double_value>,

"clientconnfailrate":<Double_value>,

"auditmemallocfailrate":<Double_value>,

"auditlog32errsyslogallocnsbfail":<Double_value>,

"logsdroppedrate":<Double_value>,

"auditportallocfail":<Double_value>,

"nsbchainallocfail":<Double_value>,

"logunsentlbsys":<Double_value>,

"systcpconnfailrate":<Double_value>,

"systcpconnfail":<Double_value>,
```

```
"auditmemallocfail":<Double_value>,

"logunsentlbsysrate":<Double_value>,

"auditsyslogmsggenrate":<Double_value>,

"logsdropped":<Double_value>,

"auditsyslogmsgsent":<Double_value>,

"auditcontextnotfound":<Double_value>,

"flushcmdcnt":<Double_value>,

"logsdroppedtxminnsbsrate":<Double_value>,

"auditsyslogmsggen":<Double_value>,

"nsbchainallocfailrate":<Double_value>,

"auditsyslogmsgsenttcp":<Double_value>

}]}
```

Authentication

Authentication statistics.

NAME	DESCRIPTION
authenticationloginschemapolicy	
authenticationoauthidppolicy	Statistics for AAA OAuth IdentityProvider (IdP) policy resource.
authenticationpolicy	Statistics for Authentication Policy resource.
authenticationpolicylabel	Statistics for authentication policy label resource.

NAME	DESCRIPTION
authenticationsamlidppolicy	Statistics for AAA Saml IdentityProvider (IdP) policy resource.
authenticationvserver	Statistics for authentication virtual server resource.

authenticationloginschemapolicy

Read/write properties

Properties

(click to see Operations)

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
name	<String>	Read-write	The name of the LoginSchema policy for which statistics will be displayed. If not given statistics are shown for all policies. Minimum length = 1
clearstats	<String>	Read-write	Clear the statsistics / counters. Possible values = basic, full
pipolicyhits	<Double>	Read-only	Number of hits on the policy
pipolicyhitsrate	<Double>	Read-only	Rate (/s) counter for pipolicyhits
pipolicyundefhits	<Double>	Read-only	Number of undef hits on the policy
pipolicyundefhitsrate	<Double>	Read-only	Rate (/s) counter for pipolicyundefhits

Operations

(click to see Properties)

GET (ALL)| GET

Some options that you can use for each operations:

- **Getting warnings in response:**NITRO allows you to get warnings in an operation by specifying the "warning" query parameter as "yes". For example, to get warnings while connecting to the Citrix ADC appliance, the URL is as follows:

  http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/login?warning=yes

  If any, the warnings are displayed in the response payload with the HTTP code "209 X-NITRO-WARNING".

- **Authenticated access for individual NITRO operations:**NITRO allows you to logon to the Citrix ADC appliance to perform individual operations. You can use this option instead of creating a NITRO session (using the login object) and then using that session to perform all operations,

  To do this, you must specify the username and password in the request header of the NITRO request as follows:

  X-NITRO-USER:<username>

  X-NITRO-PASS:<password>

  Note: In such cases, make sure that the request header DOES not include the following:

  Cookie:NITRO_AUTH_TOKEN=<tokenvalue>

`*Note:`*

Mandatory parameters are marked in redand placeholder content is marked in <green>.

get (all)

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/stat/authenticationloginschemapolicy

Query-parameters:

args

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/stat/authenticationloginschemapolicy?**args=name:<String_value>,detail:<Boolean_value>,fullvalues:<Boolean_value>,ntimes:<Double_value>**

Use this query-parameter to get authenticationloginschemapolicy resources based on additional properties.

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "authenticationloginschemapolicy": [ {

"name":<String_value>,

"pipolicyundefhitsrate":<Double_value>,

"pipolicyundefhits":<Double_value>,

"pipolicyhitsrate":<Double_value>,

"pipolicyhits":<Double_value>

}]}
```

get

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/stat/authenticationloginschemapolicy/name_value><String>

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "authenticationloginschemapolicy": [ {

"name":<String_value>,

"pipolicyundefhitsrate":<Double_value>,

"pipolicyundefhits":<Double_value>,

"pipolicyhitsrate":<Double_value>,

"pipolicyhits":<Double_value>

}]}
```

authenticationoauthidppolicy

Statistics for AAA OAuth IdentityProvider (IdP) policy resource.

Properties

(click to see Operations)

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
name	<String>	Read-write	The name of the OAuth Identity Provider (IdP) policy for which statistics will be displayed. If not given statistics are shown for all policies. Minimum length = 1
clearstats	<String>	Read-write	Clear the statsistics / counters. Possible values = basic, full
pipolicyhits	<Double>	Read-only	Number of hits on the policy
pipolicyhitsrate	<Double>	Read-only	Rate (/s) counter for pipolicyhits
pipolicyundefhits	<Double>	Read-only	Number of undef hits on the policy
pipolicyundefhitsrate	<Double>	Read-only	Rate (/s) counter for pipolicyundefhits

Operations

(click to see Properties)

GET (ALL)| GET

Some options that you can use for each operations:

- **Getting warnings in response:**NITRO allows you to get warnings in an operation by specifying the "warning" query parameter as "yes". For example, to get warnings while connecting to the Citrix ADC appliance, the URL is as follows:

  http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/login?warning=yes

If any, the warnings are displayed in the response payload with the HTTP code "209 X-NITRO-WARNING".

- **Authenticated access for individual NITRO operations:**NITRO allows you to logon to the Citrix ADC appliance to perform individual operations. You can use this option instead of creating a NITRO session (using the login object) and then using that session to perform all operations,

  To do this, you must specify the username and password in the request header of the NITRO request as follows:

  X-NITRO-USER:<username>

  X-NITRO-PASS:<password>

  Note: In such cases, make sure that the request header DOES not include the following:

  Cookie:NITRO_AUTH_TOKEN=<tokenvalue>

`*Note:`*

Mandatory parameters are marked in redand placeholder content is marked in <green>.

get (all)

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/stat/authenticationoauthidppolicy

Query-parameters:

args

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/stat/authenticationoauthidppolicy?
args=name:<String_value>,detail:<Boolean_value>,fullvalues:<Boolean_value>,ntimes:<Double_value>

Use this query-parameter to get authenticationoauthidppolicy resources based on additional properties.

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "authenticationoauthidppolicy": [ {

"name":<String_value>,

"pipolicyundefhitsrate":<Double_value>,

"pipolicyundefhits":<Double_value>,

"pipolicyhitsrate":<Double_value>,

"pipolicyhits":<Double_value>

}]}
```

get

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/stat/authenticationoauthidppolicy/name_value><String>

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "authenticationoauthidppolicy": [ {

"name":<String_value>,

"pipolicyundefhitsrate":<Double_value>,

"pipolicyundefhits":<Double_value>,

"pipolicyhitsrate":<Double_value>,

"pipolicyhits":<Double_value>

}]}
```

authenticationpolicy

Statistics for Authentication Policy resource.

Properties

(click to see Operations)

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
name	<String>	Read-write	Name of the advanced authentication policy for which to display statistics. If no name is specified, statistics for all advanced authentication polices are shown. Minimum length = 1
clearstats	<String>	Read-write	Clear the statsistics / counters.

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
			Possible values = basic, full
pipolicyhits	<Double>	Read-only	Number of hits on the policy
pipolicyhitsrate	<Double>	Read-only	Rate (/s) counter for pipolicyhits
pipolicyundefhits	<Double>	Read-only	Number of undef hits on the policy
pipolicyundefhitsrate	<Double>	Read-only	Rate (/s) counter for pipolicyundefhits

Operations

(click to see Properties)

GET (ALL)| GET

Some options that you can use for each operations:

- **Getting warnings in response:**NITRO allows you to get warnings in an operation by specifying the "warning" query parameter as "yes". For example, to get warnings while connecting to the Citrix ADC appliance, the URL is as follows:

  http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/login?warning=yes

  If any, the warnings are displayed in the response payload with the HTTP code "209 X-NITRO-WARNING".

- **Authenticated access for individual NITRO operations:**NITRO allows you to logon to the Citrix ADC appliance to perform individual operations. You can use this option instead of creating a NITRO session (using the login object) and then using that session to perform all operations,

  To do this, you must specify the username and password in the request header of the NITRO request as follows:

  X-NITRO-USER:<username>

X-NITRO-PASS:<password>

Note: In such cases, make sure that the request header DOES not include the following:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue>

`*Note:`*

Mandatory parameters are marked in redand placeholder content is marked in <green>.

get (all)

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/stat/authenticationpolicy

Query-parameters:

args

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/stat/authenticationpolicy?
args=name:<String_value>,detail:<Boolean_value>,fullvalues:<Boolean_value>,ntimes:<Double_value>

Use this query-parameter to get authenticationpolicy resources based on additional properties.

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "authenticationpolicy": [ {

"name":<String_value>,

"pipolicyundefhitsrate":<Double_value>,
```

```
"pipolicyundefhits":<Double_value>,

"pipolicyhitsrate":<Double_value>,

"pipolicyhits":<Double_value>

}]}
```

get

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/stat/authenticationpolicy/name_value><String>

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "authenticationpolicy": [ {

"name":<String_value>,

"pipolicyundefhitsrate":<Double_value>,

"pipolicyundefhits":<Double_value>,

"pipolicyhitsrate":<Double_value>,

"pipolicyhits":<Double_value>
```

```
}]}
```

authenticationpolicylabel

Statistics for authentication policy label resource.

Properties

(click to see Operations)

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
labelname	<String>	Read-write	Name of the authentication policy label. Minimum length = 1
clearstats	<String>	Read-write	Clear the statsistics / counters. Possible values = basic, full
pipolicylabelhits	<Double>	Read-only	Number of times policy label was invoked.
pipolicylabelhitsrate	<Double>	Read-only	Rate (/s) counter for pipolicylabelhits

Operations

(click to see Properties)

GET (ALL)| GET

Some options that you can use for each operations:

- **Getting warnings in response:**NITRO allows you to get warnings in an operation by specifying the "warning" query parameter as "yes". For example, to get warnings while connecting to the Citrix ADC appliance, the URL is as follows:

  http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/login?warning=yes

If any, the warnings are displayed in the response payload with the HTTP code "209 X-NITRO-WARNING".

- **Authenticated access for individual NITRO operations:**NITRO allows you to logon to the Citrix ADC appliance to perform individual operations. You can use this option instead of creating a NITRO session (using the login object) and then using that session to perform all operations,

  To do this, you must specify the username and password in the request header of the NITRO request as follows:

  X-NITRO-USER:<username>

  X-NITRO-PASS:<password>

  Note: In such cases, make sure that the request header DOES not include the following:

  Cookie:NITRO_AUTH_TOKEN=<tokenvalue>

`*Note:`*

Mandatory parameters are marked in redand placeholder content is marked in <green>.

get (all)

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/stat/authenticationpolicylabel

Query-parameters:

args

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/stat/authenticationpolicylabel?
args=labelname:<String_value>,detail:<Boolean_value>,fullvalues:<Boolean_value>,ntimes:<Double_va

Use this query-parameter to get authenticationpolicylabel resources based on additional properties.

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "authenticationpolicylabel": [ {

"labelname":<String_value>,

"pipolicylabelhitsrate":<Double_value>,

"pipolicylabelhits":<Double_value>

}]}
```

get

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/stat/authenticationpolicylabel/labelname_value><String>

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "authenticationpolicylabel": [ {

"labelname":<String_value>,

"pipolicylabelhitsrate":<Double_value>,

"pipolicylabelhits":<Double_value>

}]}
```

authenticationsamlidppolicy

Statistics for AAA Saml IdentityProvider (IdP) policy resource.

Properties

(click to see Operations)

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
name	<String>	Read-write	The name of the SAML Identity Provider (IdP) policy for which statistics will be displayed. If not given statistics are shown for all policies. Minimum length = 1
clearstats	<String>	Read-write	Clear the statsistics / counters. Possible values = basic, full
pipolicyhits	<Double>	Read-only	Number of hits on the policy

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
pipolicyhitsrate	<Double>	Read-only	Rate (/s) counter for pipolicyhits

Operations

(click to see Properties)

GET (ALL)| GET

Some options that you can use for each operations:

- **Getting warnings in response:**NITRO allows you to get warnings in an operation by specifying the "warning" query parameter as "yes". For example, to get warnings while connecting to the Citrix ADC appliance, the URL is as follows:

  http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/login?warning=yes

  If any, the warnings are displayed in the response payload with the HTTP code "209 X-NITRO-WARNING".
- **Authenticated access for individual NITRO operations:**NITRO allows you to logon to the Citrix ADC appliance to perform individual operations. You can use this option instead of creating a NITRO session (using the login object) and then using that session to perform all operations,

  To do this, you must specify the username and password in the request header of the NITRO request as follows:

  X-NITRO-USER:<username>

  X-NITRO-PASS:<password>

  Note: In such cases, make sure that the request header DOES not include the following:

  Cookie:NITRO_AUTH_TOKEN=<tokenvalue>

`*Note:`*

Mandatory parameters are marked in redand placeholder content is marked in <green>.

get (all)

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/stat/authenticationsamlidppolicy

Query-parameters:

args

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/stat/authenticationsamlidppolicy?
args=name:<String_value>,detail:<Boolean_value>,fullvalues:<Boolean_value>,ntimes:<Double_value>

Use this query-parameter to get authenticationsamlidppolicy resources based on additional properties.

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "authenticationsamlidppolicy": [ {

"name":<String_value>,

"pipolicyhitsrate":<Double_value>,

"pipolicyhits":<Double_value>

}]}
```

get

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/stat/authenticationsamlidppolicy/name_value><String>

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "authenticationsamlidppolicy": [ {

"name":<String_value>,

"pipolicyhitsrate":<Double_value>,

"pipolicyhits":<Double_value>

}]}
```

authenticationvserver

Statistics for authentication virtual server resource.

Properties

(click to see Operations)

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
name	<String>	Read-write	Name of the authentication virtual server. Minimum length = 1

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
clearstats	<String>	Read-write	Clear the statsistics / counters. Possible values = basic, full
primaryipaddress	<String>	Read-only	The IP address on which the service is running.
primaryport	<Integer>	Read-only	The port on which the service is running.
type	<String>	Read-only	Protocol associated with the vserver
state	<String>	Read-only	Current state of the server. There are seven possible values: UP(7), DOWN(1), UNKNOWN(2), BUSY(3), OFS(Out of Service)(4), TROFS(Transition Out of Service)(5), TROFS_DOWN(Down When going Out of Service)(8)
totalrequests	<Double>	Read-only	Total number of requests received on this service or virtual server. (This applies to HTTP/SSL services and servers.)
requestsrate	<Double>	Read-only	Rate (/s) counter for totalrequests

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
totalresponses	<Double>	Read-only	Number of responses received on this service or virtual server. (This applies to HTTP/SSL services and servers.)
responsesrate	<Double>	Read-only	Rate (/s) counter for totalresponses
totalrequestbytes	<Double>	Read-only	Total number of request bytes received on this service or virtual server.
requestbytesrate	<Double>	Read-only	Rate (/s) counter for totalrequestbytes
totalresponsebytes	<Double>	Read-only	Number of response bytes received by this service or virtual server.
responsebytesrate	<Double>	Read-only	Rate (/s) counter for totalresponsebytes

Operations

(click to see Properties)

GET (ALL)| GET

Some options that you can use for each operations:

- **Getting warnings in response:**NITRO allows you to get warnings in an operation by specifying the "warning" query parameter as "yes". For example, to get warnings while connecting to the Citrix ADC appliance, the URL is as follows:

  http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/login?warning=yes

If any, the warnings are displayed in the response payload with the HTTP code "209 X-NITRO-WARNING".

- **Authenticated access for individual NITRO operations:**NITRO allows you to logon to the Citrix ADC appliance to perform individual operations. You can use this option instead of creating a NITRO session (using the login object) and then using that session to perform all operations,

  To do this, you must specify the username and password in the request header of the NITRO request as follows:

  X-NITRO-USER:<username>

  X-NITRO-PASS:<password>

  Note: In such cases, make sure that the request header DOES not include the following:

  Cookie:NITRO_AUTH_TOKEN=<tokenvalue>

`*Note:`*

Mandatory parameters are marked in redand placeholder content is marked in <green>.

get (all)

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/stat/authenticationvserver

Query-parameters:

args

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/stat/authenticationvserver?
args=name:<String_value>,detail:<Boolean_value>,fullvalues:<Boolean_value>,ntimes:<Double_value>

Use this query-parameter to get authenticationvserver resources based on additional properties.

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "authenticationvserver": [ {

"name":<String_value>,

"totalrequests":<Double_value>,

"responsebytesrate":<Double_value>,

"requestbytesrate":<Double_value>,

"type":<String_value>,

"primaryipaddress":<String_value>,

"responsesrate":<Double_value>,

"totalrequestbytes":<Double_value>,

"state":<String_value>,

"totalresponses":<Double_value>,

"totalresponsebytes":<Double_value>,

"primaryport":<Integer_value>,

"requestsrate":<Double_value>

}]}
```

get

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/stat/authenticationvserver/name_value><String>

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "authenticationvserver": [ {

"name":<String_value>,

"totalrequests":<Double_value>,

"responsebytesrate":<Double_value>,

"requestbytesrate":<Double_value>,

"type":<String_value>,

"primaryipaddress":<String_value>,

"responsesrate":<Double_value>,

"totalrequestbytes":<Double_value>,

"state":<String_value>,

"totalresponses":<Double_value>,
```

```
"totalresponsebytes":<Double_value>,

"primaryport":<Integer_value>,

"requestsrate":<Double_value>

}]}
```

Authorization

Authorization statistics. Authorization services check which resources users are authorized to access, and grant permissions accordingly.

NAME	DESCRIPTION
authorizationpolicylabel	Statistics for authorization policy label resource.

authorizationpolicylabel

Statistics for authorization policy label resource.

Properties

(click to see Operations)

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
labelname	<String>	Read-write	Name of the authorization policy label. Minimum length = 1
clearstats	<String>	Read-write	Clear the statsistics / counters. Possible values = basic, full

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
pipolicylabelhits	<Double>	Read-only	Number of times policy label was invoked.
pipolicylabelhitsrate	<Double>	Read-only	Rate (/s) counter for pipolicylabelhits

Operations

(click to see Properties)

GET (ALL)| GET

Some options that you can use for each operations:

- **Getting warnings in response:**NITRO allows you to get warnings in an operation by specifying the "warning" query parameter as "yes". For example, to get warnings while connecting to the Citrix ADC appliance, the URL is as follows:

  http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/login?warning=yes

  If any, the warnings are displayed in the response payload with the HTTP code "209 X-NITRO-WARNING".

- **Authenticated access for individual NITRO operations:**NITRO allows you to logon to the Citrix ADC appliance to perform individual operations. You can use this option instead of creating a NITRO session (using the login object) and then using that session to perform all operations,

  To do this, you must specify the username and password in the request header of the NITRO request as follows:

  X-NITRO-USER:<username>

  X-NITRO-PASS:<password>

  Note: In such cases, make sure that the request header DOES not include the following:

  Cookie:NITRO_AUTH_TOKEN=<tokenvalue>

`*Note:`*

Mandatory parameters are marked in redand placeholder content is marked in <green>.

get (all)

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/stat/authorizationpolicylabel

Query-parameters:

args

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/stat/authorizationpolicylabel?
args=labelname:<String_value>,detail:<Boolean_value>,fullvalues:<Boolean_value>,ntimes:<Double_va

Use this query-parameter to get authorizationpolicylabel resources based on additional properties.

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "authorizationpolicylabel": [ {

"labelname":<String_value>,

"pipolicylabelhitsrate":<Double_value>,

"pipolicylabelhits":<Double_value>

}]}
```

get

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/stat/authorizationpolicylabel/labelname_value><String>

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "authorizationpolicylabel": [ {

"labelname":<String_value>,

"pipolicylabelhitsrate":<Double_value>,

"pipolicylabelhits":<Double_value>

}]}
```

Autoscale

AutoScale statistics.

NAME	DESCRIPTION
autoscalepolicy	Statistics for Autoscale policy resource.

autoscalepolicy

Statistics for Autoscale policy resource.

Properties

(click to see Operations)

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
name	<String>	Read-write	The name of the autoscale policy for which statistics will be displayed. If not given statistics are shown for all autoscale policies. Minimum length = 1
clearstats	<String>	Read-write	Clear the statsistics / counters. Possible values = basic, full
pipolicyhits	<Double>	Read-only	Number of hits on the policy
pipolicyhitsrate	<Double>	Read-only	Rate (/s) counter for pipolicyhits
pipolicyundefhits	<Double>	Read-only	Number of undef hits on the policy
pipolicyundefhitsrate	<Double>	Read-only	Rate (/s) counter for pipolicyundefhits

Operations

(click to see Properties)

GET (ALL)| GET

Some options that you can use for each operations:

- **Getting warnings in response:**NITRO allows you to get warnings in an operation by specifying the "warning" query parameter as "yes". For example, to get warnings while connecting to the Citrix ADC appliance, the URL is as follows:

  http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/login?warning=yes

  If any, the warnings are displayed in the response payload with the HTTP code "209 X-NITRO-WARNING".

- **Authenticated access for individual NITRO operations:**NITRO allows you to logon to the Citrix ADC appliance to perform individual operations. You can use this option instead of creating a NITRO session (using the login object) and then using that session to perform all operations,

  To do this, you must specify the username and password in the request header of the NITRO request as follows:

  X-NITRO-USER:<username>

  X-NITRO-PASS:<password>

  Note: In such cases, make sure that the request header DOES not include the following:

  Cookie:NITRO_AUTH_TOKEN=<tokenvalue>

`*Note:`*

Mandatory parameters are marked in redand placeholder content is marked in <green>.

get (all)

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/stat/autoscalepolicy

Query-parameters:

args

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/stat/autoscalepolicy?
args=name:<String_value>,detail:<Boolean_value>,fullvalues:<Boolean_value>,ntimes:<Double_value>

Use this query-parameter to get autoscalepolicy resources based on additional properties.

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "autoscalepolicy": [ {

"name":<String_value>,

"pipolicyundefhitsrate":<Double_value>,

"pipolicyundefhits":<Double_value>,

"pipolicyhitsrate":<Double_value>,

"pipolicyhits":<Double_value>

}]}
```

get

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/stat/autoscalepolicy/name_value><String>

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "autoscalepolicy": [ {

"name":<String_value>,

"pipolicyundefhitsrate":<Double_value>,

"pipolicyundefhits":<Double_value>,

"pipolicyhitsrate":<Double_value>,

"pipolicyhits":<Double_value>

}]}
```

Basic

Basic system statistics.

NAME	DESCRIPTION
service	Statistics for service resource.
servicegroup	Statistics for service group resource.
servicegroupmember	Statistics for service group entity resource.

service

Statistics for service resource.

Properties

(click to see Operations)

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
name	<String>	Read-write	Name of the service. Minimum length = 1
clearstats	<String>	Read-write	Clear the statsistics / counters. Possible values = basic, full
throughput	<Double>	Read-only	Number of bytes received or sent by this service (Mbps).
throughputrate	<Double>	Read-only	Rate (/s) counter for throughput
avgsvrttfb	<Double>	Read-only	Average TTFB between the Citrix ADC and the server. TTFB is the time interval between sending the request packet to a service and receiving the first response from the service
primaryipaddress	<String>	Read-only	The IP address on which the service is running.
primaryport	<Integer>	Read-only	The port on which the service is running.
servicetype	<String>	Read-only	The service type of this service.Possible values are ADNS, DNS, MYSQL, RTSP, SSL_DIAMETER, ADNS_TCP,

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
			DNS_TCP, NNTP, SIP_UDP, SSL_TCP, ANY, FTP, RADIUS, SNMP, TCP, DHCPRA, HTTP, RDP, SSL, TFTP, DIAMETER, MSSQL, RPCSVR, SSL_BRIDGE, UDP
state	<String>	Read-only	Current state of the server. There are seven possible values: UP(7), DOWN(1), UNKNOWN(2), BUSY(3), OFS(Out of Service)(4), TROFS(Transition Out of Service)(5), TROFS_DOWN(Down When going Out of Service)(8)
totalrequests	<Double>	Read-only	Total number of requests received on this service or virtual server. (This applies to HTTP/SSL services and servers.)
requestsrate	<Double>	Read-only	Rate (/s) counter for totalrequests
totalresponses	<Double>	Read-only	Number of responses received on this service or virtual server. (This applies to HTTP/SSL

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
			services and servers.)
responsesrate	<Double>	Read-only	Rate (/s) counter for totalresponses
totalrequestbytes	<Double>	Read-only	Total number of request bytes received on this service or virtual server.
requestbytesrate	<Double>	Read-only	Rate (/s) counter for totalrequestbytes
totalresponsebytes	<Double>	Read-only	Number of response bytes received by this service or virtual server.
responsebytesrate	<Double>	Read-only	Rate (/s) counter for totalresponsebytes
curclntconnections	<Double>	Read-only	Number of current client connections.
surgecount	<Double>	Read-only	Number of requests in the surge queue.
cursrvrconnections	<Double>	Read-only	Number of current connections to the actual servers behind the virtual server.
svrestablishedconn	<Double>	Read-only	Number of server connections in ESTABLISHED state.

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
curreusepool	<Double>	Read-only	Number of requests in the idle queue/ reuse pool.
maxclients	<Double>	Read-only	Maximum open connections allowed on this service.
curload	<Double>	Read-only	Load on the service that is calculated from the bound load based monitor.
totalconnreassembly queue75	<Double>	Read-only	Total no of connections with 75% TCP reassembly queue
totalconnreassembly queueflush	<Double>	Read-only	Total no of connections incurred TCP reassembly queue flush
curtflags	<Double>	Read-only	Current flags on the service for internal use in display handlers.
totsvrttlbtransactions	<Double>	Read-only	Total transactions where server TTLB is calculated.
toleratingttlbtransactions	<Double>	Read-only	Tolerable transactions based on APDEX threshold (>T ;; <4T).
frustratingttlbtransactions	<Double>	Read-only	Frustrating transactions based on APDEX threshold (>4T).

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
vsvrservicehits	<Double>	Read-only	Number of times that the service has been provided.
vsvrservicehitsrate	<Double>	Read-only	Rate (/s) counter for vsvrservicehits
activetransactions	<Double>	Read-only	Number of active transactions handled by this service. (Including those in the surge queue.) Active Transaction means number of transactions currently served by the server including those waiting in the SurgeQ

Operations

(click to see Properties)

GET (ALL)| GET

Some options that you can use for each operations:

- **Getting warnings in response:**NITRO allows you to get warnings in an operation by specifying the "warning" query parameter as "yes". For example, to get warnings while connecting to the Citrix ADC appliance, the URL is as follows:

  http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/login?warning=yes

  If any, the warnings are displayed in the response payload with the HTTP code "209 X-NITRO-WARNING".

- **Authenticated access for individual NITRO operations:**NITRO allows you to logon to the Citrix ADC appliance to perform individual operations. You can use this option instead of creating a NITRO session (using the login object) and then using that session to perform all operations,

To do this, you must specify the username and password in the request header of the NITRO request as follows:

X-NITRO-USER:<username>

X-NITRO-PASS:<password>

Note: In such cases, make sure that the request header DOES not include the following:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue>

`*Note:`*

Mandatory parameters are marked in redand placeholder content is marked in <green>.

get (all)

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/stat/service

Query-parameters:

args

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/stat/service?
args=name:<String_value>,detail:<Boolean_value>,fullvalues:<Boolean_value>,ntimes:<Double_value>

Use this query-parameter to get service resources based on additional properties.

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "service": [ {
"name":<String_value>,
"svrestablishedconn":<Double_value>,
"curclntconnections":<Double_value>,
"servicetype":<String_value>,
"totalrequests":<Double_value>,
"surgecount":<Double_value>,
"responsebytesrate":<Double_value>,
"totalresponses":<Double_value>,
"requestbytesrate":<Double_value>,
"throughput":<Double_value>,
"throughputrate":<Double_value>,
"totalconnreassemblyqueue75":<Double_value>,
"curtflags":<Double_value>,
"cursrvrconnections":<Double_value>,
"totalconnreassemblyqueueflush":<Double_value>,
"primaryipaddress":<String_value>,
"activetransactions":<Double_value>,
"toleratingttlbtransactions":<Double_value>,
"responsesrate":<Double_value>,
"maxclients":<Double_value>,
```

```
"avgsvrttfb":<Double_value>,

"curload":<Double_value>,

"totalrequestbytes":<Double_value>,

"curreusepool":<Double_value>,

"state":<String_value>,

"vsvrservicehits":<Double_value>,

"totsvrttlbtransactions":<Double_value>,

"frustratingttlbtransactions":<Double_value>,

"totalresponsebytes":<Double_value>,

"primaryport":<Integer_value>,

"requestsrate":<Double_value>,

"vsvrservicehitsrate":<Double_value>

}]}
```

get

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/stat/service/name_value><String>

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "service": [ {

"name":<String_value>,

"svrestablishedconn":<Double_value>,

"curclntconnections":<Double_value>,

"servicetype":<String_value>,

"totalrequests":<Double_value>,

"surgecount":<Double_value>,

"responsebytesrate":<Double_value>,

"totalresponses":<Double_value>,

"requestbytesrate":<Double_value>,

"throughput":<Double_value>,

"throughputrate":<Double_value>,

"totalconnreassemblyqueue75":<Double_value>,

"curtflags":<Double_value>,

"cursrvrconnections":<Double_value>,

"totalconnreassemblyqueueflush":<Double_value>,

"primaryipaddress":<String_value>,

"activetransactions":<Double_value>,

"toleratingttlbtransactions":<Double_value>,
```

```
"responsesrate":<Double_value>,

"maxclients":<Double_value>,

"avgsvrttfb":<Double_value>,

"curload":<Double_value>,

"totalrequestbytes":<Double_value>,

"curreusepool":<Double_value>,

"state":<String_value>,

"vsvrservicehits":<Double_value>,

"totsvrttlbtransactions":<Double_value>,

"frustratingttlbtransactions":<Double_value>,

"totalresponsebytes":<Double_value>,

"primaryport":<Integer_value>,

"requestsrate":<Double_value>,

"vsvrservicehitsrate":<Double_value>

}]}
```

servicegroup

Statistics for service group resource.

Properties

(click to see Operations)

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
servicegroupname	<String>	Read-write	Name of the service group for which to display settings. Minimum length = 1
clearstats	<String>	Read-write	Clear the statsistics / counters. Possible values = basic, full
state	<String>	Read-only	Current state of the server. There are seven possible values: UP(7), DOWN(1), UNKNOWN(2), BUSY(3), OFS(Out of Service)(4), TROFS(Transition Out of Service)(5), TROFS_DOWN(Down When going Out of Service)(8)
servicetype	<String>	Read-only	The service type of this service.Possible values are ADNS, DNS, MYSQL, RTSP, SSL_DIAMETER, ADNS_TCP, DNS_TCP, NNTP, SIP_UDP, SSL_TCP, ANY, FTP, RADIUS, SNMP, TCP, DHCPRA, HTTP, RDP, SSL, TFTP, DIAMETER, MSSQL, RPCSVR, SSL_BRIDGE, UDP

Operations

(click to see Properties)

GET (ALL)| GET

Some options that you can use for each operations:

- **Getting warnings in response:**NITRO allows you to get warnings in an operation by specifying the "warning" query parameter as "yes". For example, to get warnings while connecting to the Citrix ADC appliance, the URL is as follows:

  http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/login?warning=yes

  If any, the warnings are displayed in the response payload with the HTTP code "209 X-NITRO-WARNING".

- **Authenticated access for individual NITRO operations:**NITRO allows you to logon to the Citrix ADC appliance to perform individual operations. You can use this option instead of creating a NITRO session (using the login object) and then using that session to perform all operations,

  To do this, you must specify the username and password in the request header of the NITRO request as follows:

  X-NITRO-USER:<username>

  X-NITRO-PASS:<password>

  Note: In such cases, make sure that the request header DOES not include the following:

  Cookie:NITRO_AUTH_TOKEN=<tokenvalue>

`*Note:`*

Mandatory parameters are marked in redand placeholder content is marked in <green>.

get (all)

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/stat/servicegroup

Query-parameters:

args

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/stat/servicegroup?
args=servicegroupname:<String_value>,detail:<Boolean_value>,fullvalues:<Boolean_value>,ntimes:<D

Use this query-parameter to get servicegroup resources based on additional properties.

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "servicegroup": [ {

"servicegroupname":<String_value>,

"servicetype":<String_value>,

"state":<String_value>

}]}
```

get

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/stat/servicegroup/servicegroupname_value><String>

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "servicegroup": [ {

"servicegroupname":<String_value>,

"servicetype":<String_value>,

"state":<String_value>

}]}
```

servicegroupmember

Statistics for service group entity resource.

Properties

(click to see Operations)

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
servicegroupname	<String>	Read-write	Displays statistics for the specified service group.Name of the service group. Must begin with an ASCII alphanumeric or underscore (_) character, and must contain only ASCII alphanumeric,

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
			underscore, hash (#), period (.), space, colon (:), at sign (@), equal sign (=), and hyphen (-) characters. CLI Users: If the name includes one or more spaces, enclose the name in double or single quotation marks (for example, "my servicegroup" or 'my servicegroup'). Minimum length = 1
ip	<String>	Read-write	IP address of the service group. Mutually exclusive with the server name parameter.
servername	<String>	Read-write	Name of the server. Mutually exclusive with the IP address parameter. Minimum length = 1
port	<Integer>	Read-write	Port number of the service group member. Range 1 - 65535 * in CLI is represented as 65535 in NITRO API

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
clearstats	<String>	Read-write	Clear the statsistics / counters. Possible values = basic, full
avgsvrttfb	<Double>	Read-only	Average TTFB between the Citrix ADC and the server. TTFB is the time interval between sending the request packet to a service and receiving the first response from the service
primaryipaddress	<String>	Read-only	The IP address on which the service is running.
primaryport	<Integer>	Read-only	The port on which the service is running.
servicetype	<String>	Read-only	The service type of this service.Possible values are ADNS, DNS, MYSQL, RTSP, SSL_DIAMETER, ADNS_TCP, DNS_TCP, NNTP, SIP_UDP, SSL_TCP, ANY, FTP, RADIUS, SNMP, TCP, DHCPRA, HTTP, RDP, SSL, TFTP, DIAMETER, MSSQL, RPCSVR, SSL_BRIDGE, UDP

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
state	<String>	Read-only	Current state of the server. There are seven possible values: UP(7), DOWN(1), UNKNOWN(2), BUSY(3), OFS(Out of Service)(4), TROFS(Transition Out of Service)(5), TROFS_DOWN(Down When going Out of Service)(8)
totalrequests	<Double>	Read-only	Total number of requests received on this service or virtual server. (This applies to HTTP/SSL services and servers.)
requestsrate	<Double>	Read-only	Rate (/s) counter for totalrequests
totalresponses	<Double>	Read-only	Number of responses received on this service or virtual server. (This applies to HTTP/SSL services and servers.)
responsesrate	<Double>	Read-only	Rate (/s) counter for totalresponses
totalrequestbytes	<Double>	Read-only	Total number of request bytes received on this service or virtual server.

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
requestbytesrate	<Double>	Read-only	Rate (/s) counter for totalrequestbytes
totalresponsebytes	<Double>	Read-only	Number of response bytes received by this service or virtual server.
responsebytesrate	<Double>	Read-only	Rate (/s) counter for totalresponsebytes
curclntconnections	<Double>	Read-only	Number of current client connections.
surgecount	<Double>	Read-only	Number of requests in the surge queue.
cursrvrconnections	<Double>	Read-only	Number of current connections to the actual servers behind the virtual server.
svrestablishedconn	<Double>	Read-only	Number of server connections in ESTABLISHED state.
curreusepool	<Double>	Read-only	Number of requests in the idle queue/ reuse pool.
maxclients	<Double>	Read-only	Maximum open connections allowed on this service.
totalconnreassembly queue75	<Double>	Read-only	Total no of connections with 75% TCP reassembly queue

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
totalconnreassemblyqueueflush	<Double>	Read-only	Total no of connections incurred TCP reassembly queue flush
totsvrttlbtransactions	<Double>	Read-only	Total transactions where server TTLB is calculated.
toleratingttlbtransactions	<Double>	Read-only	Tolerable transactions based on APDEX threshold (>T ;; <4T).
frustratingttlbtransactions	<Double>	Read-only	Frustrating transactions based on APDEX threshold (>4T).

Operations

(click to see Properties)

GET (ALL)| GET

Some options that you can use for each operations:

- **Getting warnings in response:**NITRO allows you to get warnings in an operation by specifying the "warning" query parameter as "yes". For example, to get warnings while connecting to the Citrix ADC appliance, the URL is as follows:

  http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/login?warning=yes

  If any, the warnings are displayed in the response payload with the HTTP code "209 X-NITRO-WARNING".
- **Authenticated access for individual NITRO operations:**NITRO allows you to logon to the Citrix ADC appliance to perform individual operations. You can use this option instead of creating a NITRO session (using the login object) and then using that session to perform all operations,

To do this, you must specify the username and password in the request header of the NITRO request as follows:

X-NITRO-USER:<username>

X-NITRO-PASS:<password>

Note: In such cases, make sure that the request header DOES not include the following:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue>

`*Note:`*

Mandatory parameters are marked in redand placeholder content is marked in <green>.

get (all)

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/stat/servicegroupmember

Query-parameters:

args

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/stat/servicegroupmember?
args=servicegroupname:<String_value>,ip:<String_value>,servername:<String_value>,port:<Integer_va

Use this query-parameter to get servicegroupmember resources based on additional properties.

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "servicegroupmember": [ {
"servicegroupname":<String_value>,
"svrestablishedconn":<Double_value>,
"curclntconnections":<Double_value>,
"servicetype":<String_value>,
"totalrequests":<Double_value>,
"surgecount":<Double_value>,
"responsebytesrate":<Double_value>,
"totalresponses":<Double_value>,
"requestbytesrate":<Double_value>,
"totalconnreassemblyqueue75":<Double_value>,
"cursrvrconnections":<Double_value>,
"totalconnreassemblyqueueflush":<Double_value>,
"primaryipaddress":<String_value>,
"toleratingttlbtransactions":<Double_value>,
"responsesrate":<Double_value>,
"maxclients":<Double_value>,
"avgsvrttfb":<Double_value>,
"totalrequestbytes":<Double_value>,
"curreusepool":<Double_value>,
"state":<String_value>,
```

```
"totsvrttlbtransactions":<Double_value>,

"frustratingttlbtransactions":<Double_value>,

"totalresponsebytes":<Double_value>,

"primaryport":<Integer_value>,

"requestsrate":<Double_value>

}]}
```

get

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/stat/servicegroupmember/servicegroupname_value><String>

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "servicegroupmember": [ {

"servicegroupname":<String_value>,

"svrestablishedconn":<Double_value>,

"curclntconnections":<Double_value>,
```

```
"servicetype":<String_value>,

"totalrequests":<Double_value>,

"surgecount":<Double_value>,

"responsebytesrate":<Double_value>,

"totalresponses":<Double_value>,

"requestbytesrate":<Double_value>,

"totalconnreassemblyqueue75":<Double_value>,

"cursrvrconnections":<Double_value>,

"totalconnreassemblyqueueflush":<Double_value>,

"primaryipaddress":<String_value>,

"toleratingttlbtransactions":<Double_value>,

"responsesrate":<Double_value>,

"maxclients":<Double_value>,

"avgsvrttfb":<Double_value>,

"totalrequestbytes":<Double_value>,

"curreusepool":<Double_value>,

"state":<String_value>,

"totsvrttlbtransactions":<Double_value>,

"frustratingttlbtransactions":<Double_value>,

"totalresponsebytes":<Double_value>,

"primaryport":<Integer_value>,

"requestsrate":<Double_value>
```

```
}]}
```

ca

Statistics for ca.

Properties

(click to see Operations)

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
clearstats	<String>	Read-write	Clear the statsistics / counters. Possible values = basic, full
caaunindentifiedbyteshr	<Double>	Read-only	This tells the hit ratio of unindentified bytes
caiosbyteshr	<Double>	Read-only	This tells the hit ratio of IOS bytes
caotherbyteshr	<Double>	Read-only	This tells the hit ratio of Other device
calaptpdsktpbyteshr	<Double>	Read-only	This tells the hit ratio of laptop_desktop bytes
caandroidbyteshr	<Double>	Read-only	This tells the hit ratio of 3GP bytes
ca3gpvidcachebytes hr	<Double>	Read-only	This tells the hit ratio of 3GP bytes
camp4vidcachebytes hr	<Double>	Read-only	This tells the hit ratio of MP4 bytes

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
caflvvidcachebyteshr	<Double>	Read-only	This tells the hit ratio of FLV bytes
caaacaudiocachebyteshr	<Double>	Read-only	This tells the hit ratio AAC bytes served
caadtsaudiocachebyteshr	<Double>	Read-only	This tells the byte hit ratio of ADTS bytes
caaplelivestrmngplvidcachebyteshr	<Double>	Read-only	This tells the byte hit ratio of AppleLive Playlist bytes
caapplelivestrmngvidcachebyteshr	<Double>	Read-only	This tells the total number of AppleLive bytes
camsftsmthstrmngplvidcabyteshr	<Double>	Read-only	This tells the byte hit ratio of MicrosoftSmoothStreaming Playlist bytes
camsftsmthstrmngvidcabyteshr	<Double>	Read-only	This tells the Bytes hit ratio of MicrosoftSmoothStreaming bytes .
caunidentifiedhr	<Double>	Read-only	This tells the hit ratio of android requests
caioshr	<Double>	Read-only	This tells the hit ratio of ios requests
caotherhr	<Double>	Read-only	This tells the hit ratio of other mobile device requests
calaptopdesktphr	<Double>	Read-only	This tells the hit ratio of laptop/desktop requests

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
caandroidhr	<Double>	Read-only	This tells the hit ratio of android requests
ca3gpvidhr	<Double>	Read-only	This tells the total number of 3GP requests
cacmp4vidhr	<Double>	Read-only	This tells the hit ratio of MP4 requests
cacflvvidhr	<Double>	Read-only	This tells the hit ratio of FLV requests
cacaacaudiohr	<Double>	Read-only	This tells the hit ratio of AAC requests
cacadtsaudiohr	<Double>	Read-only	This tells the Hit Ratio of ADTS requests
cacapplelivestreamingplaylistvidhr	<Double>	Read-only	This tells the hit ratio of AppleLive Playlist requests
cacapplelivestreamingvidhr	<Double>	Read-only	This tells the hit ratio of AppleLive requests
cacmsftsmthstrmplvidhr	<Double>	Read-only	This tells the hit ratio of MicrosoftSmoothStreaming Playlist requests
camsftsmthstrmvidhr	<Double>	Read-only	This tells the hit ratio of MicrosoftSmoothStreaming requests
catotother	<Double>	Read-only	

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
catotvideo	<Double>	Read-only	
catotaudio	<Double>	Read-only	
catotobjstored	<Double>	Read-only	
calookuphitrate	<Double>	Read-only	Rate (/s) counter for catotlookuphit
catotgetobjreq	<Double>	Read-only	
catotrequests	<Double>	Read-only	
camsftsmthstrmvid	<Double>	Read-only	This tells the total number of MicrosoftSmoothStreaming requests served by Citrix ADC
camsftsmthstrmvidrate	<Double>	Read-only	Rate (/s) counter for camsftsmthstrmvid
cacmsftsmthstrmvid	<Double>	Read-only	This tells the total number of MicrosoftSmoothStreaming requests served from cache
cacmsftsmthstrmvidrate	<Double>	Read-only	Rate (/s) counter for cacmsftsmthstrmvid
camsftsmthstrvid	<Double>	Read-only	This tells the total number of MicrosoftSmoothStreaming Playlist requests served by Citrix ADC
camsftsmthstrvidrate	<Double>	Read-only	Rate (/s) counter for camsftsmthstrvid

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
cacmsftsmthstrmplvid	<Double>	Read-only	This tells the total number of MicrosoftSmoothStreaming Playlist requests served from cache
cacmsftsmthstrmplvidrate	<Double>	Read-only	Rate (/s) counter for cacmsftsmthstrmplvid
cacaplelivestrmngvid	<Double>	Read-only	This tells the total number of AppleLive requests served by Citrix ADC
cacaplelivestrmngvidrate	<Double>	Read-only	Rate (/s) counter for cacaplelivestrmngvid
cacapplelivestreamingvid	<Double>	Read-only	This tells the total number of AppleLive requests served from cache
cacapplelivestreamingvidrate	<Double>	Read-only	Rate (/s) counter for cacapplelivestreamingvid
cacapplelivestrmngvid	<Double>	Read-only	This tells the total number of AppleLive Playlist requests served by Citrix ADC
cacapplelivestrmngvidrate	<Double>	Read-only	Rate (/s) counter for cacapplelivestrmngvid
cacapplelivestreamingplaylistvid	<Double>	Read-only	This tells the total number of AppleLive Playlist requests served from cache

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
cacapplelivestreamingplaylistvidrate	<Double>	Read-only	Rate (/s) counter for cacapplelivestreamingplaylistvid
caadtsaudio	<Double>	Read-only	This tells the total number of ADTS requests served by Citrix ADC
caadtsaudiorate	<Double>	Read-only	Rate (/s) counter for caadtsaudio
cacadtsaudio	<Double>	Read-only	This tells the total number of ADTS requests served from cache
cacadtsaudiorate	<Double>	Read-only	Rate (/s) counter for cacadtsaudio
caaacaudio	<Double>	Read-only	This tells the total number of AAC requests served by Citrix ADC
caaacaudiorate	<Double>	Read-only	Rate (/s) counter for caaacaudio
cacaacaudio	<Double>	Read-only	This tells the total number of AAC requests served from cache
cacaacaudiorate	<Double>	Read-only	Rate (/s) counter for cacaacaudio
caflvvid	<Double>	Read-only	This tells the total number of FLV requests served by Citrix ADC

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
caflvvidrate	<Double>	Read-only	Rate (/s) counter for caflvvid
cacflvvid	<Double>	Read-only	This tells the total number of FLV requests served from cache
cacflvvidrate	<Double>	Read-only	Rate (/s) counter for cacflvvid
camp4vid	<Double>	Read-only	This tells the total number of MP4 requests served by Citrix ADC
camp4vidrate	<Double>	Read-only	Rate (/s) counter for camp4vid
cacmp4vid	<Double>	Read-only	This tells the total number of MP4 requests served from cache
cacmp4vidrate	<Double>	Read-only	Rate (/s) counter for cacmp4vid
ca3pvid	<Double>	Read-only	This tells the total number of 3GP requests served by Citrix ADC
ca3pvidrate	<Double>	Read-only	Rate (/s) counter for ca3pvid
ca3gpvid	<Double>	Read-only	This tells the hit ratio of 3GP requests served from cache
ca3gpvidrate	<Double>	Read-only	Rate (/s) counter for ca3gpvid

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
camsftsmthstrmvidbytes	<Double>	Read-only	This tells the total number of MicrosoftSmoothStreaming bytes served by Citrix ADC
camsftsmthstrmvidbytesrate	<Double>	Read-only	Rate (/s) counter for camsftsmthstrmvidbytes
camicrosoftsmoothstreamingvidcachebytes	<Double>	Read-only	This tells the total number of MicrosoftSmoothStreaming bytes served from cache.
camicrosoftsmoothstreamingvidcachebytesrate	<Double>	Read-only	Rate (/s) counter for camicrosoftsmoothstreamingvidcachebytes
camsftsmthstrmplvidbytes	<Double>	Read-only	This tells the total number of MicrosoftSmoothStreaming Playlist bytes served by Citrix ADC
camsftsmthstrmplvidbytesrate	<Double>	Read-only	Rate (/s) counter for camsftsmthstrmplvidbytes
camicrosoftsmoothstreamingplaylistvidcachebytes	<Double>	Read-only	This tells the total number of MicrosoftSmoothStreaming Playlist bytes served from cache
camicrosoftsmoothstreamingplaylistvidcachebytesrate	<Double>	Read-only	Rate (/s) counter for camicrosoftsmoothstreamingplaylistvidcachebytes

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
caapplelivestreamingvidbytes	<Double>	Read-only	This tells the total number of AppleLive bytes served by Citrix ADC
caapplelivestreamingvidbytesrate	<Double>	Read-only	Rate (/s) counter for caapplelivestreamingvidbytes
caapplelivestreamingvidcachebytes	<Double>	Read-only	This tells the total number of AppleLive bytes served from cache
caapplelivestreamingvidcachebytesrate	<Double>	Read-only	Rate (/s) counter for caapplelivestreamingvidcachebytes
caapplelivestreamingplaylistvidbytes	<Double>	Read-only	This tells the total number of AppleLive Playlist bytes served by Citrix ADC
caapplelivestreamingplaylistvidbytesrate	<Double>	Read-only	Rate (/s) counter for caapplelivestreamingplaylistvidbytes
caapplelivestreamingplaylistvidcachebytes	<Double>	Read-only	This tells the total number of AppleLive Playlist bytes served from cache
caapplelivestreamingplaylistvidcachebytesrate	<Double>	Read-only	Rate (/s) counter for caapplelivestreamingplaylistvidcachebytes
caadtsaudiobytes	<Double>	Read-only	This tells the total number of ADTS bytes served by Citrix ADC

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
caadtsaudiobytesrate	<Double>	Read-only	Rate (/s) counter for caadtsaudiobytes
caadtsaudiocachebytes	<Double>	Read-only	This tells the total number of ADTS bytes served from cache
caadtsaudiocachebytesrate	<Double>	Read-only	Rate (/s) counter for caadtsaudiocachebytes
caaacaudiobytes	<Double>	Read-only	This tells the total number of AAC bytes served by Citrix ADC
caaacaudiobytesrate	<Double>	Read-only	Rate (/s) counter for caaacaudiobytes
caaacaudiocachebytes	<Double>	Read-only	This tells the total number of AAC bytes served from cache
caaacaudiocachebytesrate	<Double>	Read-only	Rate (/s) counter for caaacaudiocachebytes
caflvvidbytes	<Double>	Read-only	This tells the total number of FLV bytes served by Citrix ADC
caflvvidbytesrate	<Double>	Read-only	Rate (/s) counter for caflvvidbytes
caflvvidcachebytes	<Double>	Read-only	This tells the total number of FLV bytes served from cache
caflvvidcachebytesrate	<Double>	Read-only	Rate (/s) counter for caflvvidcachebytes

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
camp4vidbytes	<Double>	Read-only	This tells the total number of MP4 bytes served by Citrix ADC
camp4vidbytesrate	<Double>	Read-only	Rate (/s) counter for camp4vidbytes
camp4vidcachebytes	<Double>	Read-only	This tells the total number of MP4 bytes served from cache
camp4vidcachebytes rate	<Double>	Read-only	Rate (/s) counter for camp4vidcachebytes
ca3gpvidbytes	<Double>	Read-only	This tells the total number of 3GP bytes served by Citrix ADC
ca3gpvidbytesrate	<Double>	Read-only	Rate (/s) counter for ca3gpvidbytes
ca3gpvidcachebytes	<Double>	Read-only	This tells the total number of 3GP bytes served from cache
ca3gpvidcachebytesr ate	<Double>	Read-only	Rate (/s) counter for ca3gpvidcachebytes
caandroid	<Double>	Read-only	Total number of android requests to Citrix ADC
caandroidrate	<Double>	Read-only	Rate (/s) counter for caandroid
calaptopdesktp	<Double>	Read-only	Total number of laptop/desktop requests to Citrix ADC

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
calaptopdesktprate	<Double>	Read-only	Rate (/s) counter for calaptopdesktp
caios	<Double>	Read-only	Total number of iOs requests to Citrix ADC
caiosrate	<Double>	Read-only	Rate (/s) counter for caios
caother	<Double>	Read-only	Total number of other mobile device requests to Citrix ADC
caotherrate	<Double>	Read-only	Rate (/s) counter for caother
caunidentified	<Double>	Read-only	Total number of unidentified requests to Citrix ADC
caunidentifiedrate	<Double>	Read-only	Rate (/s) counter for caunidentified
caandroidcache	<Double>	Read-only	This tells android requests served from cache
caandroidcacherate	<Double>	Read-only	Rate (/s) counter for caandroidcache
caioscache	<Double>	Read-only	This tells iOS requests served from cache
caioscacherate	<Double>	Read-only	Rate (/s) counter for caioscache

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
caothercache	<Double>	Read-only	This tells Other device requests served from cache
caothercacherate	<Double>	Read-only	Rate (/s) counter for caothercache
calaptopdesktpcache	<Double>	Read-only	This tells laptop/ desktop requests served from cache
calaptopdesktpcache rate	<Double>	Read-only	Rate (/s) counter for calaptopdesktpcache
caunidentifiedcache	<Double>	Read-only	This tells unidentified device requests served from cache
caunidentifiedcacher ate	<Double>	Read-only	Rate (/s) counter for caunidentifiedcache
caandroidbytes	<Double>	Read-only	This tells the total number of Android bytes served by Citrix ADC
caandroidbytesrate	<Double>	Read-only	Rate (/s) counter for caandroidbytes
caiosbytes	<Double>	Read-only	This tells the total number of IOS bytes served by Citrix ADC
caiosbytesrate	<Double>	Read-only	Rate (/s) counter for caiosbytes
caotherbytes	<Double>	Read-only	This tells the total number of Other mobile device bytes served by Citrix ADC

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
caotherbytesrate	<Double>	Read-only	Rate (/s) counter for caotherbytes
caalptopdsktpbytes	<Double>	Read-only	This tells the total number of Laptop/ desktop bytes served by Citrix ADC
caalptopdsktpbytesrate	<Double>	Read-only	Rate (/s) counter for caalptopdsktpbytes
caunidentifiedbytes	<Double>	Read-only	This tells the total number of unidentified device bytes served by Citrix ADC
caunidentifiedbytesrate	<Double>	Read-only	Rate (/s) counter for caunidentifiedbytes
caandroididcachebytes	<Double>	Read-only	This tells the total number of Android bytes served from cache
caandroididcachebytesrate	<Double>	Read-only	Rate (/s) counter for caandroididcachebytes
caiosidcachebytes	<Double>	Read-only	This tells the total number of IOS bytes served from cache
caiosidcachebytesrate	<Double>	Read-only	Rate (/s) counter for caiosidcachebytes
caotherididcachebytes	<Double>	Read-only	This tells the total number of other device bytes served from cache

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
caotherididcachebytesrate	<Double>	Read-only	Rate (/s) counter for caotherididcachebytes
calaptpdektpbytes	<Double>	Read-only	This tells the total number of Laptop/desktop bytes served from cache
calaptpdektpbytesrate	<Double>	Read-only	Rate (/s) counter for calaptpdektpbytes
caaunindentifiedbytes	<Double>	Read-only	This tells the total number of unindentified device bytes served from cache
caaunindentifiedbytesrate	<Double>	Read-only	Rate (/s) counter for caaunindentifiedbytes

Operations

(click to see Properties)

GET (ALL)

Some options that you can use for each operations:

- **Getting warnings in response:**NITRO allows you to get warnings in an operation by specifying the "warning" query parameter as "yes". For example, to get warnings while connecting to the Citrix ADC appliance, the URL is as follows:

  http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/login?warning=yes

  If any, the warnings are displayed in the response payload with the HTTP code "209 X-NITRO-WARNING".
- **Authenticated access for individual NITRO operations:**NITRO allows you to logon to the Citrix ADC appliance to perform individual operations. You can use this option instead of

creating a NITRO session (using the login object) and then using that session to perform all operations,

To do this, you must specify the username and password in the request header of the NITRO request as follows:

X-NITRO-USER:<username>

X-NITRO-PASS:<password>

Note: In such cases, make sure that the request header DOES not include the following:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue>

`*Note:`*

Mandatory parameters are marked in redand placeholder content is marked in <green>.

get (all)

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/stat/ca

Query-parameters:

args

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/stat/ca?
args=detail:<Boolean_value>,fullvalues:<Boolean_value>,ntimes:<Double_value>,logfile:<String_value

Use this query-parameter to get ca resources based on additional properties.

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "ca": [ {
"caaacaudiocachebytes":<Double_value>,
"camsftsmthstrmvidhr":<Double_value>,
"caflvvidrate":<Double_value>,
"cacmsftsmthstrmplvid":<Double_value>,
"camp4vidbytesrate":<Double_value>,
"camsftsmthstrmngplvidcabyteshr":<Double_value>,
"caapplelivestreamingplaylistvidbytesrate":<Double_value>,
"catotvideo":<Double_value>,
"caaacaudiobytes":<Double_value>,
"calookuphitrate":<Double_value>,
"cacapplelivestreamingplaylistvid":<Double_value>,
"cacmp4vidhr":<Double_value>,
"cacapplelivestreamingvidhr":<Double_value>,
"caflvvidbytesrate":<Double_value>,
"catotobjstored":<Double_value>,
"ca3gpvidcachebytesrate":<Double_value>,
"caaacaudiocachebyteshr":<Double_value>,
"caadtsaudiobytes":<Double_value>,
"cacadtsaudio":<Double_value>,
"caaudiorate":<Double_value>,
```

```
"cacapplelivestreamingplaylistvidrate":<Double_value>,

"caaplelivestrmngplvidcachebyteshr":<Double_value>,

"caotherididcachebytes":<Double_value>,

"calaptpdsktpbyteshr":<Double_value>,

"caflvvidcachebytes":<Double_value>,

"caalptopdsktpbytesrate":<Double_value>,

"cacflvvidrate":<Double_value>,

"caaunindentifiedbyteshr":<Double_value>,

"caapplelivestreamingplaylistvidcachebytes":<Double_value>,

"catotrequests":<Double_value>,

"cacflvvidhr":<Double_value>,

"caadtsaudiocachebyteshr":<Double_value>,

"caiosbytesrate":<Double_value>,

"cacmp4vidrate":<Double_value>,

"catotlookuphit":<Double_value>,

"caadtsaudiobytesrate":<Double_value>,

"ca3pvid":<Double_value>,

"calaptopdesktpcache":<Double_value>,

"camsftsmthstrmplvidbytes":<Double_value>,

"caiosidcachebytesrate":<Double_value>,

"ca3gpvidbytesrate":<Double_value>,

"caunidentifiedcache":<Double_value>,
```

```
"caapplelivestreamingvidcachebytesrate":<Double_value>,

"caapplelivestreamingvidbytes":<Double_value>,

"caflvvidcachebyteshr":<Double_value>,

"caaunindentifiedbytesrate":<Double_value>,

"caios":<Double_value>,

"caiosrate":<Double_value>,

"caandroidbytes":<Double_value>,

"caotherrate":<Double_value>,

"caflvvidbytes":<Double_value>,

"calaptpdektpbytes":<Double_value>,

"caapplelivestreamingplaylistvidcachebytesrate":<Double_value>,

"camsftsmthstrmplvidbytesrate":<Double_value>,

"caaacaudiorate":<Double_value>,

"ca3gpvidhr":<Double_value>,

"calaptopdesktprate":<Double_value>,

"camp4vidrate":<Double_value>,

"cacadtsaudiohr":<Double_value>,

"cacapplelivestrmngvid":<Double_value>,

"caaacaudiocachebytesrate":<Double_value>,

"caapplelivestreamingvidbytesrate":<Double_value>,

"cacaplelivestrmngvidrate":<Double_value>,

"caunidentifiedhr":<Double_value>,
```

```
"cagetobjreqrate":<Double_value>,

"camp4vidcachebyteshr":<Double_value>,

"calaptopdesktphr":<Double_value>,

"camsftsmthstrmvidbytesrate":<Double_value>,

"camsftsmthstrmngvidcabyteshr":<Double_value>,

"camicrosoftsmoothstreamingplaylistvidcachebytes":<Double_value>,

"caiosbytes":<Double_value>,

"calaptopdesktp":<Double_value>,

"caothercacherate":<Double_value>,

"camp4vidcachebytesrate":<Double_value>,

"cacaacaudiohr":<Double_value>,

"caotherididcachebytesrate":<Double_value>,

"camsftsmthstrmvidbytes":<Double_value>,

"cacmsftsmthstrmplvidhr":<Double_value>,

"calaptopdesktpcacherate":<Double_value>,

"ca3gpvidbytes":<Double_value>,

"caotherbytes":<Double_value>,

"carequestsrate":<Double_value>,

"caunidentifiedbytesrate":<Double_value>,

"ca3gpvidcachebyteshr":<Double_value>,

"caadtsaudiocachebytes":<Double_value>,

"caadtsaudio":<Double_value>,
```

```
"caadtsaudiocachebytesrate":<Double_value>,

"ca3gpvidrate":<Double_value>,

"caioscache":<Double_value>,

"caaunindentifiedbytes":<Double_value>,

"cacapplelivestreamingplaylistvidhr":<Double_value>,

"caunidentified":<Double_value>,

"caandroid":<Double_value>,

"caapplelivestreamingplaylistvidbytes":<Double_value>,

"camp4vidbytes":<Double_value>,

"caunidentifiedrate":<Double_value>,

"cacaacaudio":<Double_value>,

"cacflvvid":<Double_value>,

"catotgetobjreq":<Double_value>,

"caotherbyteshr":<Double_value>,

"caioshr":<Double_value>,

"calaptpdektpbytesrate":<Double_value>,

"caandroididcachebytes":<Double_value>,

"caflvvid":<Double_value>,

"cacadtsaudiorate":<Double_value>,

"caandroidbyteshr":<Double_value>,

"caandroididcachebytesrate":<Double_value>,

"cacmp4vid":<Double_value>,
```

```
"caandroidbytesrate":<Double_value>,

"cacmsftsmthstrmplvidrate":<Double_value>,

"caadtsaudiorate":<Double_value>,

"camp4vid":<Double_value>,

"caunidentifiedcacherate":<Double_value>,

"caandroidcacherate":<Double_value>,

"cavideorate":<Double_value>,

"camicrosoftsmoothstreamingplaylistvidcachebytesrate":<Double_value>,

"caaacaudio":<Double_value>,

"catotother":<Double_value>,

"cacapplelivestreamingvidrate":<Double_value>,

"cacapplelivestreamingvid":<Double_value>,

"caandroidhr":<Double_value>,

"cacmsftsmthstrmvidrate":<Double_value>,

"caioscacherate":<Double_value>,

"caunidentifiedbytes":<Double_value>,

"caandroidrate":<Double_value>,

"cacaacaudiorate":<Double_value>,

"cacmsftsmthstrmvid":<Double_value>,

"camicrosoftsmoothstreamingvidcachebytes":<Double_value>,

"caother":<Double_value>,

"caalptopdsktpbytes":<Double_value>,
```

```
"caothercache":<Double_value>,

"camsftsmthstrvid":<Double_value>,

"camp4vidcachebytes":<Double_value>,

"ca3gpvid":<Double_value>,

"cacapplelivestrmngvidrate":<Double_value>,

"camicrosoftsmoothstreamingvidcachebytesrate":<Double_value>,

"catoherrate":<Double_value>,

"caotherhr":<Double_value>,

"camsftsmthstrmvidrate":<Double_value>,

"camsftsmthstrvidrate":<Double_value>,

"ca3gpvidcachebytes":<Double_value>,

"caaacaudiobytesrate":<Double_value>,

"camsftsmthstrmvid":<Double_value>,

"cacaplelivestrmngvid":<Double_value>,

"caapplelivestrmngvidcachebyteshr":<Double_value>,

"ca3pvidrate":<Double_value>,

"caandroidcache":<Double_value>,

"caflvvidcachebytesrate":<Double_value>,

"caiosidcachebytes":<Double_value>,

"caotherbytesrate":<Double_value>,

"caapplelivestreamingvidcachebytes":<Double_value>,

"caiosbyteshr":<Double_value>,
```

```
"catotaudio":<Double_value>

}]}
```

Cache Redirection

Cache redirection statistics. The cache redirection feature can transparently redirect cacheable HTTP requests to a cache and send non-cacheable or dynamic HTTP requests to the origin server. A cache stores or caches frequently requested web content and serves such web content to a client on behalf of the origin servers, alleviating the load on the origin server farm.

NAME	DESCRIPTION
crvserver	Statistics for CR virtual server resource.

crvserver

Statistics for CR virtual server resource.

Properties

(click to see Operations)

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
name	<String>	Read-write	Name of a specific cache redirection virtual server. Minimum length = 1
clearstats	<String>	Read-write	Clear the statsistics / counters. Possible values = basic, full
primaryipaddress	<String>	Read-only	The IP address on which the service is running.

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
primaryport	<Integer>	Read-only	The port on which the service is running.
type	<String>	Read-only	Protocol associated with the vserver
state	<String>	Read-only	Current state of the server. There are seven possible values: UP(7), DOWN(1), UNKNOWN(2), BUSY(3), OFS(Out of Service)(4), TROFS(Transition Out of Service)(5), TROFS_DOWN(Down When going Out of Service)(8)
tothits	<Double>	Read-only	Total vserver hits
hitsrate	<Double>	Read-only	Rate (/s) counter for tothits
totalrequests	<Double>	Read-only	Total number of requests received on this service or virtual server. (This applies to HTTP/SSL services and servers.)
requestsrate	<Double>	Read-only	Rate (/s) counter for totalrequests
totalresponses	<Double>	Read-only	Number of responses received on this service or virtual server. (This

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
			applies to HTTP/SSL services and servers.)
responsesrate	<Double>	Read-only	Rate (/s) counter for totalresponses
totalrequestbytes	<Double>	Read-only	Total number of request bytes received on this service or virtual server.
requestbytesrate	<Double>	Read-only	Rate (/s) counter for totalrequestbytes
totalresponsebytes	<Double>	Read-only	Number of response bytes received by this service or virtual server.
responsebytesrate	<Double>	Read-only	Rate (/s) counter for totalresponsebytes
totalpktsrecvd	<Double>	Read-only	Total number of packets received by this service or virtual server.
pktsrecvdrate	<Double>	Read-only	Rate (/s) counter for totalpktsrecvd
totalpktssent	<Double>	Read-only	Total number of packets sent.
pktssentrate	<Double>	Read-only	Rate (/s) counter for totalpktssent
invalidrequestrespon se	<Double>	Read-only	Number invalid requests/responses on this vserver

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
invalidrequestrespon sedropped	<Double>	Read-only	Number invalid requests/responses dropped on this vserver

Operations

(click to see Properties)

GET (ALL)| GET

Some options that you can use for each operations:

- **Getting warnings in response:**NITRO allows you to get warnings in an operation by specifying the "warning" query parameter as "yes". For example, to get warnings while connecting to the Citrix ADC appliance, the URL is as follows:

  http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/login?warning=yes

  If any, the warnings are displayed in the response payload with the HTTP code "209 X-NITRO-WARNING".
- **Authenticated access for individual NITRO operations:**NITRO allows you to logon to the Citrix ADC appliance to perform individual operations. You can use this option instead of creating a NITRO session (using the login object) and then using that session to perform all operations,

  To do this, you must specify the username and password in the request header of the NITRO request as follows:

  X-NITRO-USER:<username>

  X-NITRO-PASS:<password>

  Note: In such cases, make sure that the request header DOES not include the following:

  Cookie:NITRO_AUTH_TOKEN=<tokenvalue>

`*Note:`*

Mandatory parameters are marked in redand placeholder content is marked in <green>.

get (all)

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/stat/crvserver

Query-parameters:

args

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/stat/crvserver?
args=name:<String_value>,detail:<Boolean_value>,fullvalues:<Boolean_value>,ntimes:<Double_value>

Use this query-parameter to get crvserver resources based on additional properties.

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "crvserver": [ {

"name":<String_value>,

"totalpktssent":<Double_value>,

"tothits":<Double_value>,

"totalrequests":<Double_value>,

"responsebytesrate":<Double_value>,

"invalidrequestresponsedropped":<Double_value>,
```

```
"totalresponses":<Double_value>,

"requestbytesrate":<Double_value>,

"type":<String_value>,

"hitsrate":<Double_value>,

"pktsrecvdrate":<Double_value>,

"primaryipaddress":<String_value>,

"responsesrate":<Double_value>,

"totalrequestbytes":<Double_value>,

"invalidrequestresponse":<Double_value>,

"state":<String_value>,

"totalpktsrecvd":<Double_value>,

"pktssentrate":<Double_value>,

"totalresponsebytes":<Double_value>,

"primaryport":<Integer_value>,

"requestsrate":<Double_value>

}]}
```

get

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/stat/crvserver/name_value><String>

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "crvserver": [ {

"name":<String_value>,

"totalpktssent":<Double_value>,

"tothits":<Double_value>,

"totalrequests":<Double_value>,

"responsebytesrate":<Double_value>,

"invalidrequestresponsedropped":<Double_value>,

"totalresponses":<Double_value>,

"requestbytesrate":<Double_value>,

"type":<String_value>,

"hitsrate":<Double_value>,

"pktsrecvdrate":<Double_value>,

"primaryipaddress":<String_value>,

"responsesrate":<Double_value>,

"totalrequestbytes":<Double_value>,

"invalidrequestresponse":<Double_value>,
```

```
"state":<String_value>,

"totalpktsrecvd":<Double_value>,

"pktssentrate":<Double_value>,

"totalresponsebytes":<Double_value>,

"primaryport":<Integer_value>,

"requestsrate":<Double_value>

}]}
```

CI

Content Inspection

NAME	DESCRIPTION
contentinspectionpolicy	Statistics for ContentInspection policy resource.
contentinspectionpolicylabel	Statistics for ContentInspection policy label resource.

contentinspectionpolicy

Statistics for ContentInspection policy resource.

Properties

(click to see Operations)

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
name	<String>	Read-write	Name of the contentInspection policy for which to

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
			show detailed statistics. Minimum length = 1
clearstats	<String>	Read-write	Clear the statsistics / counters. Possible values = basic, full
pipolicyhits	<Double>	Read-only	Number of hits on the policy
pipolicyhitsrate	<Double>	Read-only	Rate (/s) counter for pipolicyhits
pipolicyundefhits	<Double>	Read-only	Number of undef hits on the policy
pipolicyundefhitsrate	<Double>	Read-only	Rate (/s) counter for pipolicyundefhits

Operations

(click to see Properties)

GET (ALL)| GET

Some options that you can use for each operations:

- **Getting warnings in response:**NITRO allows you to get warnings in an operation by specifying the "warning" query parameter as "yes". For example, to get warnings while connecting to the Citrix ADC appliance, the URL is as follows:

  http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/login?warning=yes

  If any, the warnings are displayed in the response payload with the HTTP code "209 X-NITRO-WARNING".

- **Authenticated access for individual NITRO operations:**NITRO allows you to logon to the Citrix ADC appliance to perform individual operations. You can use this option instead of

creating a NITRO session (using the login object) and then using that session to perform all operations,

To do this, you must specify the username and password in the request header of the NITRO request as follows:

X-NITRO-USER:<username>

X-NITRO-PASS:<password>

Note: In such cases, make sure that the request header DOES not include the following:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue>

`*Note:`*

Mandatory parameters are marked in redand placeholder content is marked in <green>.

get (all)

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/stat/contentinspectionpolicy

Query-parameters:

args

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/stat/contentinspectionpolicy?**args=name:<String_value>,detail:<Boolean_value>,fullvalues:<Boolean_value>,ntimes:<Double_value>**

Use this query-parameter to get contentinspectionpolicy resources based on additional properties.

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "contentinspectionpolicy": [ {

"name":<String_value>,

"pipolicyundefhitsrate":<Double_value>,

"pipolicyundefhits":<Double_value>,

"pipolicyhitsrate":<Double_value>,

"pipolicyhits":<Double_value>

}]}
```

get

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/stat/contentinspectionpolicy/name_value><String>

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "contentinspectionpolicy": [ {

"name":<String_value>,
```

```
"pipolicyundefhitsrate":<Double_value>,

"pipolicyundefhits":<Double_value>,

"pipolicyhitsrate":<Double_value>,

"pipolicyhits":<Double_value>

}]}
```

contentinspectionpolicylabel

Statistics for ContentInspection policy label resource.

Properties

(click to see Operations)

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
labelname	<String>	Read-write	Name of the contentInspection policy label. Minimum length = 1
clearstats	<String>	Read-write	Clear the statsistics / counters. Possible values = basic, full
pipolicylabelhits	<Double>	Read-only	Number of times policy label was invoked.
pipolicylabelhitsrate	<Double>	Read-only	Rate (/s) counter for pipolicylabelhits

Operations

(click to see Properties)

GET (ALL)| GET

Some options that you can use for each operations:

- **Getting warnings in response:**NITRO allows you to get warnings in an operation by specifying the "warning" query parameter as "yes". For example, to get warnings while connecting to the Citrix ADC appliance, the URL is as follows:

  http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/login?warning=yes

  If any, the warnings are displayed in the response payload with the HTTP code "209 X-NITRO-WARNING".

- **Authenticated access for individual NITRO operations:**NITRO allows you to logon to the Citrix ADC appliance to perform individual operations. You can use this option instead of creating a NITRO session (using the login object) and then using that session to perform all operations,

  To do this, you must specify the username and password in the request header of the NITRO request as follows:

  X-NITRO-USER:<username>

  X-NITRO-PASS:<password>

  Note: In such cases, make sure that the request header DOES not include the following:

  Cookie:NITRO_AUTH_TOKEN=<tokenvalue>

`*Note:`*

Mandatory parameters are marked in redand placeholder content is marked in <green>.

get (all)

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/stat/contentinspectionpolicylabel

Query-parameters:

args

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/stat/contentinspectionpolicylabel?
args=labelname:<String_value>,detail:<Boolean_value>,fullvalues:<Boolean_value>,ntimes:<Double_va

Use this query-parameter to get contentinspectionpolicylabel resources based on additional properties.

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "contentinspectionpolicylabel": [ {

"labelname":<String_value>,

"pipolicylabelhitsrate":<Double_value>,

"pipolicylabelhits":<Double_value>

}]}
```

get

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/stat/contentinspectionpolicylabel/labelname_value><String>

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "contentinspectionpolicylabel": [ {

"labelname":<String_value>,

"pipolicylabelhitsrate":<Double_value>,

"pipolicylabelhits":<Double_value>

}]}
```

Cluster

Cluster commands.

NAME	DESCRIPTION
clusterinstance	Statistics for cluster instance resource.
clusternode	Statistics for cluster node resource.

clusterinstance

Statistics for cluster instance resource.

Properties

(click to see Operations)

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
clid	<Double>	Read-write	ID of the cluster instance for which to display statistics. Minimum value = 1 Maximum value = 16
clearstats	<String>	Read-write	Clear the statsistics / counters. Possible values = basic, full
clbkplanetx	<Double>	Read-only	Traffic transmitted from backplane (in mbits)
clbkplanetxrate	<Double>	Read-only	Rate (/s) counter for clbkplanetx
clbkplanerx	<Double>	Read-only	Traffic received on backplane (in mbits)
clbkplanerxrate	<Double>	Read-only	Rate (/s) counter for clbkplanerx
clnumnodes	<Double>	Read-only	Number of nodes in the cluster.
clcurstatus	<String>	Read-only	State of the cluster.
clviewleader	<String>	Read-only	NSIP address of the Configuration Coordinator of the cluster.
totsteeredpkts	<Double>	Read-only	Total number of packets steered on the cluster backplane.

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
numdfddroppkts	<Double>	Read-only	Number of steered packets that are dropped.
totpropagationtimeout	<Double>	Read-only	Number of times the update to the client timed-out.

Operations

(click to see Properties)

GET (ALL)| GET

Some options that you can use for each operations:

- **Getting warnings in response:**NITRO allows you to get warnings in an operation by specifying the "warning" query parameter as "yes". For example, to get warnings while connecting to the Citrix ADC appliance, the URL is as follows:

  http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/login?warning=yes

  If any, the warnings are displayed in the response payload with the HTTP code "209 X-NITRO-WARNING".

- **Authenticated access for individual NITRO operations:**NITRO allows you to logon to the Citrix ADC appliance to perform individual operations. You can use this option instead of creating a NITRO session (using the login object) and then using that session to perform all operations,

  To do this, you must specify the username and password in the request header of the NITRO request as follows:

  X-NITRO-USER:<username>

  X-NITRO-PASS:<password>

  Note: In such cases, make sure that the request header DOES not include the following:

  Cookie:NITRO_AUTH_TOKEN=<tokenvalue>

`*Note:`*

Mandatory parameters are marked in redand placeholder content is marked in <green>.

get (all)

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/stat/clusterinstance

Query-parameters:

args

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/stat/clusterinstance?
args=clid:<Double_value>,detail:<Boolean_value>,fullvalues:<Boolean_value>,ntimes:<Double_value>,

Use this query-parameter to get clusterinstance resources based on additional properties.

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "clusterinstance": [ {

"clid":<Double_value>,

"totpropagationtimeout":<Double_value>,

"clbkplanetx":<Double_value>,

"clbkplanetxrate":<Double_value>,

"totsteeredpkts":<Double_value>,
```

```
"numdfddroppkts":<Double_value>,

"clviewleader":<String_value>,

"clbkplanerx":<Double_value>,

"clcurstatus":<String_value>,

"clnumnodes":<Double_value>,

"clbkplanerxrate":<Double_value>

}]}
```

get

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/stat/clusterinstance/clid_value><Double>

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "clusterinstance": [ {

"clid":<Double_value>,

"totpropagationtimeout":<Double_value>,
```

```
"clbkplanetx":<Double_value>,

"clbkplanetxrate":<Double_value>,

"totsteeredpkts":<Double_value>,

"numdfddroppkts":<Double_value>,

"clviewleader":<String_value>,

"clbkplanerx":<Double_value>,

"clcurstatus":<String_value>,

"clnumnodes":<Double_value>,

"clbkplanerxrate":<Double_value>

}]}
```

clusternode

Statistics for cluster node resource.

Properties

(click to see Operations)

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
nodeid	<Double>	Read-write	ID of the cluster node for which to display statistics. If an ID is not provided, statistics are shown for all nodes. Minimum value = 0 Maximum value = 31

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
clearstats	<String>	Read-write	Clear the statsistics / counters. Possible values = basic, full
clsyncstate	<String>	Read-only	Sync state of the cluster node.
clnodeeffectivehealth	<String>	Read-only	Health of the cluster node.
clnodeip	<String>	Read-only	NSIP address of the cluster node.
clmasterstate	<String>	Read-only	Operational state of the cluster node.
cltothbtx	<Double>	Read-only	Number of heartbeats sent. When executed from the NSIP address, shows the statistics for local node only. For remote node it shows a value of 0. When executed from the cluster IP address, shows all the statistics.
cltothbrx	<Double>	Read-only	Number of heartbeats received. When executed from the NSIP address, shows the statistics for local node only. For remote node it shows a value of 0. When executed from the cluster IP

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
			address, shows all the statistics.
nnmcurconn	<Double>	Read-only	Number of connections open for node-to-node communication.
nnmtotconntx	<Double>	Read-only	Number of node-to-node messages sent. When executed from the NSIP address, shows the statistics for local node only. For remote node it shows a value of 0. When executed from the cluster IP address, shows all the statistics.
nnmtotconnrx	<Double>	Read-only	Number of node-to-node messages received. When executed from the NSIP address, shows the statistics for local node only. For remote node it shows a value of 0. When executed from the cluster IP address, shows all the statistics.
clptpstate	<String>	Read-only	PTP state of the node. This state is Master for one node and Slave for the rest. When executed

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
			from the NSIP address, shows the statistics for local node only. For remote node it shows UNKNOWN. When executed from the cluster IP address, shows all the statistics.
clptptx	<Double>	Read-only	Number of PTP packets transmitted by the node. When executed from the NSIP address, shows the statistics for local node only. For remote node it shows a value of 0. When executed from the cluster IP address, shows all the statistics.
clptprx	<Double>	Read-only	Number of PTP packets received on the node. When executed from the NSIP address, shows the statistics for local node only. For remote node it shows a value of 0. When executed from the cluster IP address, shows all the statistics.
nnmerrmsend	<Double>	Read-only	Number of errors in sending node-to-

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
			node multicast/ broadcast messages. When executed from the NSIP address, shows the statistics for local node only. For remote node it shows a value of 0. When executed from the cluster IP address, shows all the statistics.

Operations

(click to see Properties)

GET (ALL)| GET

Some options that you can use for each operations:

- **Getting warnings in response:**NITRO allows you to get warnings in an operation by specifying the "warning" query parameter as "yes". For example, to get warnings while connecting to the Citrix ADC appliance, the URL is as follows:

  http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/login?warning=yes

  If any, the warnings are displayed in the response payload with the HTTP code "209 X-NITRO-WARNING".

- **Authenticated access for individual NITRO operations:**NITRO allows you to logon to the Citrix ADC appliance to perform individual operations. You can use this option instead of creating a NITRO session (using the login object) and then using that session to perform all operations,

  To do this, you must specify the username and password in the request header of the NITRO request as follows:

  X-NITRO-USER:<username>

  X-NITRO-PASS:<password>

Note: In such cases, make sure that the request header DOES not include the following:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue>

`*Note:`*

Mandatory parameters are marked in redand placeholder content is marked in <green>.

get (all)

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/stat/clusternode

Query-parameters:

args

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/stat/clusternode?
args=nodeid:<Double_value>,detail:<Boolean_value>,fullvalues:<Boolean_value>,ntimes:<Double_valu

Use this query-parameter to get clusternode resources based on additional properties.

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "clusternode": [ {

"nodeid":<Double_value>,

"clnodeip":<String_value>,

"clsyncstate":<String_value>,
```

```
"nnmcurconn":<Double_value>,

"nnmerrmsend":<Double_value>,

"clnodeeffectivehealth":<String_value>,

"nnmtotconnrx":<Double_value>,

"cltothbrx":<Double_value>,

"clptprx":<Double_value>,

"nnmtotconntx":<Double_value>,

"clmasterstate":<String_value>,

"clptptx":<Double_value>,

"clptpstate":<String_value>,

"cltothbtx":<Double_value>

}]}
```

get

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/stat/clusternode/nodeid_value><Double>

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "clusternode": [ {

"nodeid":<Double_value>,

"clnodeip":<String_value>,

"clsyncstate":<String_value>,

"nnmcurconn":<Double_value>,

"nnmerrmsend":<Double_value>,

"clnodeeffectivehealth":<String_value>,

"nnmtotconnrx":<Double_value>,

"cltothbrx":<Double_value>,

"clptprx":<Double_value>,

"nnmtotconntx":<Double_value>,

"clmasterstate":<String_value>,

"clptptx":<Double_value>,

"clptpstate":<String_value>,

"cltothbtx":<Double_value>

}]}
```

Compression

Compression statistics. The system's feature for compressing HTTP responses to compression-aware browsers.

NAME	DESCRIPTION
cmp	Statistics for cmp.
cmppolicy	Statistics for compression policy resource.
cmppolicylabel	Statistics for compression policy label resource.

cmp

Statistics for cmp.

Properties

(click to see Operations)

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
clearstats	<String>	Read-write	Clear the statsistics / counters. Possible values = basic, full
delbwsaving	<Double>	Read-only	Bandwidth saving from delta compression expressed as percentage.
delcmpratio	<Double>	Read-only	Ratio of compressible data received to compressed data transmitted.If this ratio is one (uncmp:1.0) that means compression is disabled or we are not able to compress even a single compressible packet.

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
decomptcpratio	<Double>	Read-only	Ratio of decompressed data transmitted to compressed data received.
decomptcpbandwidthsaving	<Double>	Read-only	Bandwidth saving from TCP compression expressed as percentage.
comptcpratio	<Double>	Read-only	Ratio of compressible data received to compressed data transmitted.If this ratio is one (uncmp:1.0) that means compression is disabled or we are not able to compress even a single compressible packet.
comptcpbandwidthsaving	<Double>	Read-only	Bandwidth saving from TCP compression expressed as percentage.
comptotaldatacompressionratio	<Double>	Read-only	Ratio of total HTTP data received to total HTTP data transmitted.
comphttpbandwidthsaving	<Double>	Read-only	Bandwidth saving from TCP compression expressed as percentage.

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
compratio	<Double>	Read-only	Ratio of the compressible data received from the server to the compressed data sent to the client.
comptotalrequests	<Double>	Read-only	Number of HTTP compression requests the Citrix ADC receives for which the response is successfully compressed. For example, after you enable compression and configure services, if you send HTTP requests to the Citrix ADC using a HTTP client that supports compression, and Citrix ADC compresses the corresponding response, this counter is incremented.
comprequestsrate	<Double>	Read-only	Rate (/s) counter for comptotalrequests
comptotalrxbytes	<Double>	Read-only	Number of bytes that can be compressed, which the Citrix ADC receives from the server. This gives the content length of the response that the

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
			Citrix ADC receives from server.
comprxbytesrate	<Double>	Read-only	Rate (/s) counter for comptotalrxbytes
comptotaltxbytes	<Double>	Read-only	Number of bytes the Citrix ADC sends to the client after compressing the response from the server.
comptxbytesrate	<Double>	Read-only	Rate (/s) counter for comptotaltxbytes
comptotalrxpackets	<Double>	Read-only	Number of HTTP packets that can be compressed, which the Citrix ADC receives from the server.
comprxpacketsrate	<Double>	Read-only	Rate (/s) counter for comptotalrxpackets
comptotaltxpackets	<Double>	Read-only	Number of HTTP packets that the Citrix ADC sends to the client after compressing the response from the server.
comptxpacketsrate	<Double>	Read-only	Rate (/s) counter for comptotaltxpackets
comptcptotalrxbytes	<Double>	Read-only	Number of bytes that can be compressed, which the Citrix ADC receives from the server. This gives the

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
			content length of the response that the Citrix ADC receives from server.
comptcprxbytesrate	<Double>	Read-only	Rate (/s) counter for comptcptotalrxbytes
comptcptotalrxpackets	<Double>	Read-only	Total number of compressible packets received by Citrix ADC.
comptcprxpacketsrate	<Double>	Read-only	Rate (/s) counter for comptcptotalrxpackets
comptcptotaltxbytes	<Double>	Read-only	Number of bytes that the Citrix ADC sends to the client after compressing the response from the server.
comptcptxbytesrate	<Double>	Read-only	Rate (/s) counter for comptcptotaltxbytes
comptcptotaltxpackets	<Double>	Read-only	Number of TCP packets that the Citrix ADC sends to the client after compressing the response from the server.
comptcptxpacketsrate	<Double>	Read-only	Rate (/s) counter for comptcptotaltxpackets
comptcptotalquantum	<Double>	Read-only	Number of times the Citrix ADC compresses a

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
			quantum of data. Citrix ADC buffers the data received from the server till it reaches the quantum size and then compresses the buffered data and transmits to the client.
comptcpquantumrate	<Double>	Read-only	Rate (/s) counter for comptcptotalquantum
comptcptotalpush	<Double>	Read-only	Number of times the Citrix ADC compresses data on receiving a TCP PUSH flag from the server. The PUSH flag ensures that data is compressed immediately without waiting for the buffered data size to reach the quantum size.
comptcppushrate	<Double>	Read-only	Rate (/s) counter for comptcptotalpush
comptcptotaleoi	<Double>	Read-only	Number of times the Citrix ADC compresses data on receiving End Of Input (FIN packet). When the Citrix ADC receives End Of Input (FIN packet), it compresses the

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
			buffered data immediately without waiting for the buffered data size to reach the quantum size.
comptcpeoirate	<Double>	Read-only	Rate (/s) counter for comptcptotaleoi
comptcptotaltimer	<Double>	Read-only	Number of times the Citrix ADC compresses data on expiration of data accumulation timer. The timer expires if the server response is very slow and consequently, the Citrix ADC does not receive response for a certain amount of time. Under such a condition, the Citrix ADC compresses the buffered data immediately without waiting for the buffered data size to reach the quantum size.
comptcptimerrate	<Double>	Read-only	Rate (/s) counter for comptcptotaltimer
decomptcprxbytes	<Double>	Read-only	Total number of compressed bytes received by Citrix ADC.

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
decomptcprxbytesrate	<Double>	Read-only	Rate (/s) counter for decomptcprxbytes
decomptcprxpackets	<Double>	Read-only	Total number of compressed packets received by Citrix ADC.
decomptcprxpacketsrate	<Double>	Read-only	Rate (/s) counter for decomptcprxpackets
decomptcptxbytes	<Double>	Read-only	Total number of decompressed bytes transmitted by Citrix ADC.
decomptcptxbytesrate	<Double>	Read-only	Rate (/s) counter for decomptcptxbytes
decomptcptxpackets	<Double>	Read-only	Total number of decompressed packets transmitted by Citrix ADC.
decomptcptxpacketsrate	<Double>	Read-only	Rate (/s) counter for decomptcptxpackets
decomptcperrdata	<Double>	Read-only	Number of data errors encountered while decompressing.
decomptcperrdatarate	<Double>	Read-only	Rate (/s) counter for decomptcperrdata
decomptcperrlessdata	<Double>	Read-only	Number of times Citrix ADC received less data than declared by protocol.

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
decomptcperrlessdatarate	<Double>	Read-only	Rate (/s) counter for decomptcperrlessdata
decomptcperrmoredata	<Double>	Read-only	Number of times Citrix ADC received more data than declared by protocol.
decomptcperrmoredatarate	<Double>	Read-only	Rate (/s) counter for decomptcperrmoredata
decomptcperrmemory	<Double>	Read-only	Number of times memory failures occurred while decompressing.
decomptcperrmemoryrate	<Double>	Read-only	Rate (/s) counter for decomptcperrmemory
decomptcperrunknown	<Double>	Read-only	Number of times unknown errors occurred while decompressing.
decomptcperrunknownrate	<Double>	Read-only	Rate (/s) counter for decomptcperrunknown
delcomptotalrequests	<Double>	Read-only	Total number of delta compression requests received by Citrix ADC.
delcomprequestsrate	<Double>	Read-only	Rate (/s) counter for delcomptotalrequests

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
delcompdone	<Double>	Read-only	Total number of delta compressions done by Citrix ADC.
delcompdonerate	<Double>	Read-only	Rate (/s) counter for delcompdone
delcomptcprxbytes	<Double>	Read-only	Total number of delta-compressible bytes received by Citrix ADC.
delcomptcprxbytesrate	<Double>	Read-only	Rate (/s) counter for delcomptcprxbytes
delcomptcptxbytes	<Double>	Read-only	Total number of delta-compressed bytes transmitted by Citrix ADC.
delcomptcptxbytesrate	<Double>	Read-only	Rate (/s) counter for delcomptcptxbytes
delcompfirstaccess	<Double>	Read-only	Total number of delta compression first accesses.
delcompfirstaccessrate	<Double>	Read-only	Rate (/s) counter for delcompfirstaccess
delcomptcprxpackets	<Double>	Read-only	Number of delta-compressible packets received.
delcomptcprxpacketsrate	<Double>	Read-only	Rate (/s) counter for delcomptcprxpackets
delcomptcptxpackets	<Double>	Read-only	Total number of delta-compressed packets transmitted by Citrix ADC.

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
delcomptcptxpackets rate	<Double>	Read-only	Rate (/s) counter for delcomptcptxpackets
delcompbaseserved	<Double>	Read-only	Total number of basefile requests served by Citrix ADC.
delcompbaseservedr ate	<Double>	Read-only	Rate (/s) counter for delcompbaseserved
delcompbasetcptxbyt es	<Double>	Read-only	Number of basefile bytes transmitted by Citrix ADC.
delcompbasetcptxbyt esrate	<Double>	Read-only	Rate (/s) counter for delcompbasetcptxbyt es
delcomperrbypassed	<Double>	Read-only	Number of times delta-compression bypassed by Citrix ADC.
delcomperrbypassed rate	<Double>	Read-only	Rate (/s) counter for delcomperrbypassed
delcomperrbfilewhdrf ailed	<Double>	Read-only	Number of times basefile could not be updated in Citrix ADC cache.
delcomperrbfilewhdrf ailedrate	<Double>	Read-only	Rate (/s) counter for delcomperrbfilewhdrf ailed
delcomperrnostoremi ss	<Double>	Read-only	Number of times basefile was not found in Citrix ADC cache.

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
delcomperrnostoremissrate	<Double>	Read-only	Rate (/s) counter for delcomperrnostoremiss
delcomperrreqinfotoobig	<Double>	Read-only	Number of times basefile request URL was too large.
delcomperrreqinfotoobigrate	<Double>	Read-only	Rate (/s) counter for delcomperrreqinfotoobig
delcomperrreqinfoallocfail	<Double>	Read-only	Number of times requested basefile could not be allocated.
delcomperrreqinfoallocfailrate	<Double>	Read-only	Rate (/s) counter for delcomperrreqinfoallocfail
delcomperrsessallocfail	<Double>	Read-only	Number of times delta compression session could not be allocated.
delcomperrsessallocfailrate	<Double>	Read-only	Rate (/s) counter for delcomperrsessallocfail

Operations

(click to see Properties)

GET (ALL)

Some options that you can use for each operations:

- **Getting warnings in response:**NITRO allows you to get warnings in an operation by specifying the "warning" query parameter as "yes". For example, to get warnings while connecting to the Citrix ADC appliance, the URL is as follows:

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/login?warning=yes

If any, the warnings are displayed in the response payload with the HTTP code "209 X-NITRO-WARNING".

- **Authenticated access for individual NITRO operations:**NITRO allows you to logon to the Citrix ADC appliance to perform individual operations. You can use this option instead of creating a NITRO session (using the login object) and then using that session to perform all operations,

  To do this, you must specify the username and password in the request header of the NITRO request as follows:

  X-NITRO-USER:<username>

  X-NITRO-PASS:<password>

  Note: In such cases, make sure that the request header DOES not include the following:

  Cookie:NITRO_AUTH_TOKEN=<tokenvalue>

`*Note:`*

Mandatory parameters are marked in redand placeholder content is marked in <green>.

get (all)

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/stat/cmp

Query-parameters:

args

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/stat/cmp?
args=detail:<Boolean_value>,fullvalues:<Boolean_value>,ntimes:<Double_value>,logfile:<String_value

Use this query-parameter to get cmp resources based on additional properties.

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "cmp": [ {

"comphttpbandwidthsaving":<Double_value>,

"decomptcptxpacketsrate":<Double_value>,

"delcompbasetcptxbytesrate":<Double_value>,

"comptotaltxbytes":<Double_value>,

"comptcpeoirate":<Double_value>,

"delcomperrbypassedrate":<Double_value>,

"delcmpratio":<Double_value>,

"delcomprequestsrate":<Double_value>,

"delcomptcprxpacketsrate":<Double_value>,

"decomptcperrmemory":<Double_value>,

"comptotaldatacompressionratio":<Double_value>,

"comprxbytesrate":<Double_value>,

"delcomperrsessallocfailrate":<Double_value>,

"delcomptcptxpacketsrate":<Double_value>,

"decomptcperrmemoryrate":<Double_value>,

"decomptcperrmoredata":<Double_value>,
```

```
"delcompfirstaccessrate":<Double_value>,

"comprxpacketsrate":<Double_value>,

"decomptcprxpacketsrate":<Double_value>,

"comptcpquantumrate":<Double_value>,

"comptxbytesrate":<Double_value>,

"delcompbaseservedrate":<Double_value>,

"comptcptxbytesrate":<Double_value>,

"delcomptcprxpackets":<Double_value>,

"delcomperrbfilewhdrfailed":<Double_value>,

"decomptcperrmoredatarate":<Double_value>,

"delcompbaseserved":<Double_value>,

"compratio":<Double_value>,

"decomptcperrlessdata":<Double_value>,

"comptcprxbytesrate":<Double_value>,

"comptcprxpacketsrate":<Double_value>,

"delcomptcptxbytesrate":<Double_value>,

"decomptcperrdatarate":<Double_value>,

"delcompdonerate":<Double_value>,

"decomptcpratio":<Double_value>,

"decomptcperrlessdatarate":<Double_value>,

"comptcptotaltxbytes":<Double_value>,

"delcomptotalrequests":<Double_value>,
```

```
"delcomperrreqinfoallocfailrate":<Double_value>,

"delcomperrnostoremiss":<Double_value>,

"delcompbasetcptxbytes":<Double_value>,

"comptcptotalpush":<Double_value>,

"delcompfirstaccess":<Double_value>,

"delcompdone":<Double_value>,

"comptcptotalrxpackets":<Double_value>,

"delcomperrbypassed":<Double_value>,

"delbwsaving":<Double_value>,

"comprequestsrate":<Double_value>,

"delcomptcptxbytes":<Double_value>,

"delcomperrreqinfoallocfail":<Double_value>,

"delcomperrreqinfotoobig":<Double_value>,

"decomptcprxbytes":<Double_value>,

"comptcptxpacketsrate":<Double_value>,

"comptcptotaleoi":<Double_value>,

"comptcppushrate":<Double_value>,

"decomptcperrunknownrate":<Double_value>,

"comptcpbandwidthsaving":<Double_value>,

"comptotalrxbytes":<Double_value>,

"decomptcptxbytes":<Double_value>,

"decomptcprxpackets":<Double_value>,
```

```
"comptcptotaltimer":<Double_value>,

"delcomperrnostoremissrate":<Double_value>,

"decomptcpbandwidthsaving":<Double_value>,

"delcomperrsessallocfail":<Double_value>,

"decomptcptxbytesrate":<Double_value>,

"comptxpacketsrate":<Double_value>,

"comptotaltxpackets":<Double_value>,

"delcomperrreqinfotoobigrate":<Double_value>,

"decomptcprxbytesrate":<Double_value>,

"comptotalrequests":<Double_value>,

"decomptcperrunknown":<Double_value>,

"comptotalrxpackets":<Double_value>,

"delcomptcprxbytes":<Double_value>,

"comptcptimerrate":<Double_value>,

"comptcptotalquantum":<Double_value>,

"comptcptotaltxpackets":<Double_value>,

"delcomptcptxpackets":<Double_value>,

"comptcptotalrxbytes":<Double_value>,

"delcomptcprxbytesrate":<Double_value>,

"decomptcptxpackets":<Double_value>,

"delcomperrbfilewhdrfailedrate":<Double_value>,

"comptcpratio":<Double_value>,
```

```
"decomptcperrdata":<Double_value>

}]}
```

cmppolicy

Statistics for compression policy resource.

Properties

(click to see Operations)

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
name	<String>	Read-write	Name of the advanced compression policy for which to display statistics. If no name is specified, statistics for all advanced compression polices are shown. Minimum length = 1
clearstats	<String>	Read-write	Clear the statsistics / counters. Possible values = basic, full
pipolicyhits	<Double>	Read-only	Number of hits on the policy
pipolicyhitsrate	<Double>	Read-only	Rate (/s) counter for pipolicyhits
pipolicyundefhits	<Double>	Read-only	Number of undef hits on the policy
pipolicyundefhitsrate	<Double>	Read-only	Rate (/s) counter for pipolicyundefhits

Operations

(click to see Properties)

GET (ALL)| GET

Some options that you can use for each operations:

- **Getting warnings in response:**NITRO allows you to get warnings in an operation by specifying the "warning" query parameter as "yes". For example, to get warnings while connecting to the Citrix ADC appliance, the URL is as follows:

  http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/login?warning=yes

  If any, the warnings are displayed in the response payload with the HTTP code "209 X-NITRO-WARNING".

- **Authenticated access for individual NITRO operations:**NITRO allows you to logon to the Citrix ADC appliance to perform individual operations. You can use this option instead of creating a NITRO session (using the login object) and then using that session to perform all operations,

  To do this, you must specify the username and password in the request header of the NITRO request as follows:

  X-NITRO-USER:<username>

  X-NITRO-PASS:<password>

  Note: In such cases, make sure that the request header DOES not include the following:

  Cookie:NITRO_AUTH_TOKEN=<tokenvalue>

`*Note:`*

Mandatory parameters are marked in redand placeholder content is marked in <green>.

get (all)

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/stat/cmppolicy

Query-parameters:

args

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/stat/cmppolicy?
args=name:<String_value>,detail:<Boolean_value>,fullvalues:<Boolean_value>,ntimes:<Double_value>

Use this query-parameter to get cmppolicy resources based on additional properties.

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "cmppolicy": [ {

"name":<String_value>,

"pipolicyundefhitsrate":<Double_value>,

"pipolicyundefhits":<Double_value>,

"pipolicyhitsrate":<Double_value>,

"pipolicyhits":<Double_value>

}]}
```

get

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/stat/cmppolicy/name_value><String>

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "cmppolicy": [ {

"name":<String_value>,

"pipolicyundefhitsrate":<Double_value>,

"pipolicyundefhits":<Double_value>,

"pipolicyhitsrate":<Double_value>,

"pipolicyhits":<Double_value>

}]}
```

cmppolicylabel

Statistics for compression policy label resource.

Properties

(click to see Operations)

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
labelname	<String>	Read-write	Name of the compression policy label for which to

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
			display statistics. If not specified, statistics are displayed for all compression policy labels. Minimum length = 1
clearstats	<String>	Read-write	Clear the statsistics / counters. Possible values = basic, full
pipolicylabelhits	<Double>	Read-only	Number of times policy label was invoked.
pipolicylabelhitsrate	<Double>	Read-only	Rate (/s) counter for pipolicylabelhits

Operations

(click to see Properties)

GET (ALL)| GET

Some options that you can use for each operations:

- **Getting warnings in response:**NITRO allows you to get warnings in an operation by specifying the "warning" query parameter as "yes". For example, to get warnings while connecting to the Citrix ADC appliance, the URL is as follows:

  http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/login?warning=yes

  If any, the warnings are displayed in the response payload with the HTTP code "209 X-NITRO-WARNING".

- **Authenticated access for individual NITRO operations:**NITRO allows you to logon to the Citrix ADC appliance to perform individual operations. You can use this option instead of creating a NITRO session (using the login object) and then using that session to perform all operations,

To do this, you must specify the username and password in the request header of the NITRO request as follows:

X-NITRO-USER:<username>

X-NITRO-PASS:<password>

Note: In such cases, make sure that the request header DOES not include the following:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue>

`*Note:`*

Mandatory parameters are marked in redand placeholder content is marked in <green>.

get (all)

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/stat/cmppolicylabel

Query-parameters:

args

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/stat/cmppolicylabel?
args=labelname:<String_value>,detail:<Boolean_value>,fullvalues:<Boolean_value>,ntimes:<Double_va

Use this query-parameter to get cmppolicylabel resources based on additional properties.

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "cmppolicylabel": [ {
"labelname":<String_value>,
"pipolicylabelhitsrate":<Double_value>,
"pipolicylabelhits":<Double_value>
}]}
```

get

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/stat/cmppolicylabel/labelname_value><String>

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "cmppolicylabel": [ {
"labelname":<String_value>,
"pipolicylabelhitsrate":<Double_value>,
"pipolicylabelhits":<Double_value>
}]}
```

Content Switching

Content Switching statistics. Content Switching feature that enables you to direct traffic to servers on the basis of content.

NAME	DESCRIPTION
csvserver	Statistics for CS virtual server resource.

csvserver

Statistics for CS virtual server resource.

Properties

(click to see Operations)

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
name	<String>	Read-write	Name of the content switching virtual server for which to display statistics. To display statistics for all configured Content Switching virtual servers, do not specify a value for this parameter. Minimum length = 1
clearstats	<String>	Read-write	Clear the statsistics / counters. Possible values = basic, full
avgclttlb	<Double>	Read-only	Average TTLB between the client and the server. TTLB is the time interval

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
			between sending the request packet to a service and receiving the ACK for response from client.
cltresponsetimeapdex	<Double>	Read-only	Vserver APDEX index based on client response times.
establishedconn	<Double>	Read-only	Number of client connections in ESTABLISHED state.
primaryipaddress	<String>	Read-only	The IP address on which the service is running.
primaryport	<Integer>	Read-only	The port on which the service is running.
type	<String>	Read-only	Protocol associated with the vserver
state	<String>	Read-only	Current state of the server. There are seven possible values: UP(7), DOWN(1), UNKNOWN(2), BUSY(3), OFS(Out of Service)(4), TROFS(Transition Out of Service)(5), TROFS_DOWN(Down When going Out of Service)(8)
tothits	<Double>	Read-only	Total vserver hits

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
hitsrate	<Double>	Read-only	Rate (/s) counter for tothits
totalrequests	<Double>	Read-only	Total number of requests received on this service or virtual server. (This applies to HTTP/SSL services and servers.)
requestsrate	<Double>	Read-only	Rate (/s) counter for totalrequests
totalresponses	<Double>	Read-only	Number of responses received on this service or virtual server. (This applies to HTTP/SSL services and servers.)
responsesrate	<Double>	Read-only	Rate (/s) counter for totalresponses
totalrequestbytes	<Double>	Read-only	Total number of request bytes received on this service or virtual server.
requestbytesrate	<Double>	Read-only	Rate (/s) counter for totalrequestbytes
totalresponsebytes	<Double>	Read-only	Number of response bytes received by this service or virtual server.
responsebytesrate	<Double>	Read-only	Rate (/s) counter for totalresponsebytes

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
totalpktsrecvd	<Double>	Read-only	Total number of packets received by this service or virtual server.
pktsrecvdrate	<Double>	Read-only	Rate (/s) counter for totalpktsrecvd
totalpktssent	<Double>	Read-only	Total number of packets sent.
pktssentrate	<Double>	Read-only	Rate (/s) counter for totalpktssent
curclntconnections	<Double>	Read-only	Number of current client connections.
cursrvrconnections	<Double>	Read-only	Number of current connections to the actual servers behind the virtual server.
curpersistencesessions	<Double>	Read-only	current vserver owned persistence sessions
curbackuppersistencesessions	<Double>	Read-only	current vserver owned backup persistence sessions
sothreshold	<Double>	Read-only	Spill Over Threshold set on the VServer.
totspillovers	<Double>	Read-only	Number of times vserver experienced spill over.
labelledconn	<Double>	Read-only	Number of Labeled connection on this vserver

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
pushlabel	<Double>	Read-only	Number of labels for this push vserver.
deferredreq	<Double>	Read-only	Number of deferred request on this vserver
deferredreqrate	<Double>	Read-only	Rate (/s) counter for deferredreq
invalidrequestrespon se	<Double>	Read-only	Number invalid requests/responses on this vserver
invalidrequestrespon sedropped	<Double>	Read-only	Number invalid requests/responses dropped on this vserver
totvserverdownbacku phits	<Double>	Read-only	Number of times traffic was diverted to backup vserver since primary vserver was DOWN.
curmptcpsessions	<Double>	Read-only	Current Multipath TCP sessions
cursubflowconn	<Double>	Read-only	Current Multipath TCP subflows
totclttlbtransactions	<Double>	Read-only	Total transactions where client TTLB is calculated.
toleratingttlbtransacti ons	<Double>	Read-only	Tolerable transactions based on APDEX threshold (>T ;; <4T).

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
frustratingttlbtransactions	<Double>	Read-only	Frustrating transactions based on APDEX threshold (>4T).

Operations

(click to see Properties)

GET (ALL)| GET

Some options that you can use for each operations:

- **Getting warnings in response:**NITRO allows you to get warnings in an operation by specifying the "warning" query parameter as "yes". For example, to get warnings while connecting to the Citrix ADC appliance, the URL is as follows:

  http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/login?warning=yes

  If any, the warnings are displayed in the response payload with the HTTP code "209 X-NITRO-WARNING".
- **Authenticated access for individual NITRO operations:**NITRO allows you to logon to the Citrix ADC appliance to perform individual operations. You can use this option instead of creating a NITRO session (using the login object) and then using that session to perform all operations,

  To do this, you must specify the username and password in the request header of the NITRO request as follows:

  X-NITRO-USER:<username>

  X-NITRO-PASS:<password>

  Note: In such cases, make sure that the request header DOES not include the following:

  Cookie:NITRO_AUTH_TOKEN=<tokenvalue>

`*Note:`*

Mandatory parameters are marked in redand placeholder content is marked in <green>.

get (all)

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/stat/csvserver

Query-parameters:

args

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/stat/csvserver?
args=name:<String_value>,detail:<Boolean_value>,fullvalues:<Boolean_value>,ntimes:<Double_value>

Use this query-parameter to get csvserver resources based on additional properties.

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "csvserver": [ {

"name":<String_value>,

"curclntconnections":<Double_value>,

"establishedconn":<Double_value>,

"totalpktssent":<Double_value>,

"labelledconn":<Double_value>,

"tothits":<Double_value>,
```

```
"totalrequests":<Double_value>,

"sothreshold":<Double_value>,

"cursubflowconn":<Double_value>,

"responsebytesrate":<Double_value>,

"invalidrequestresponsedropped":<Double_value>,

"totalresponses":<Double_value>,

"curpersistencesessions":<Double_value>,

"requestbytesrate":<Double_value>,

"avgclttlb":<Double_value>,

"type":<String_value>,

"hitsrate":<Double_value>,

"cursrvrconnections":<Double_value>,

"cltresponsetimeapdex":<Double_value>,

"totclttlbtransactions":<Double_value>,

"pktsrecvdrate":<Double_value>,

"primaryipaddress":<String_value>,

"pushlabel":<Double_value>,

"toleratingttlbtransactions":<Double_value>,

"responsesrate":<Double_value>,

"deferredreq":<Double_value>,

"curmptcpsessions":<Double_value>,

"totspillovers":<Double_value>,
```

```
"totalrequestbytes":<Double_value>,

"invalidrequestresponse":<Double_value>,

"state":<String_value>,

"deferredreqrate":<Double_value>,

"totalpktsrecvd":<Double_value>,

"curbackuppersistencesessions":<Double_value>,

"pktssentrate":<Double_value>,

"frustratingttlbtransactions":<Double_value>,

"totalresponsebytes":<Double_value>,

"primaryport":<Integer_value>,

"requestsrate":<Double_value>,

"totvserverdownbackuphits":<Double_value>

}]}
```

get

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/stat/csvserver/name_value><String>

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "csvserver": [ {

"name":<String_value>,

"curclntconnections":<Double_value>,

"establishedconn":<Double_value>,

"totalpktssent":<Double_value>,

"labelledconn":<Double_value>,

"tothits":<Double_value>,

"totalrequests":<Double_value>,

"sothreshold":<Double_value>,

"cursubflowconn":<Double_value>,

"responsebytesrate":<Double_value>,

"invalidrequestresponsedropped":<Double_value>,

"totalresponses":<Double_value>,

"curpersistencesessions":<Double_value>,

"requestbytesrate":<Double_value>,

"avgclttlb":<Double_value>,

"type":<String_value>,

"hitsrate":<Double_value>,

"cursrvrconnections":<Double_value>,
```

```
"cltresponsetimeapdex":<Double_value>,

"totclttlbtransactions":<Double_value>,

"pktsrecvdrate":<Double_value>,

"primaryipaddress":<String_value>,

"pushlabel":<Double_value>,

"toleratingttlbtransactions":<Double_value>,

"responsesrate":<Double_value>,

"deferredreq":<Double_value>,

"curmptcpsessions":<Double_value>,

"totspillovers":<Double_value>,

"totalrequestbytes":<Double_value>,

"invalidrequestresponse":<Double_value>,

"state":<String_value>,

"deferredreqrate":<Double_value>,

"totalpktsrecvd":<Double_value>,

"curbackuppersistencesessions":<Double_value>,

"pktssentrate":<Double_value>,

"frustratingttlbtransactions":<Double_value>,

"totalresponsebytes":<Double_value>,

"primaryport":<Integer_value>,

"requestsrate":<Double_value>,

"totvserverdownbackuphits":<Double_value>
```

```
}]}
```

Domain Name Service

Domain Name Service(DNS) statistics.

NAME	DESCRIPTION
dns	Statistics for dns.
dnspolicylabel	Statistics for dns policy label resource.
dnsrecords	

dns

Statistics for dns.

Properties

(click to see Operations)

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
clearstats	<String>	Read-write	Clear the statsistics / counters. Possible values = basic, full
dnstotqueries	<Double>	Read-only	Total number of DNS queries received.
dnsqueriesrate	<Double>	Read-only	Rate (/s) counter for dnstotqueries
dnstotmultiquery	<Double>	Read-only	Total number of Multi Query request received.

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
dnstotanswers	<Double>	Read-only	Total number of DNS responses received.
dnsanswersrate	<Double>	Read-only	Rate (/s) counter for dnstotanswers
dnstotserverresponse	<Double>	Read-only	Total number of Server responses received.
dnsserverresponserate	<Double>	Read-only	Rate (/s) counter for dnstotserverresponse
dnstotrecupdate	<Double>	Read-only	Total number of record updates.
dnscurcachesize	<Double>	Read-only	Current DNS cache size
dnscurnegcachesize	<Double>	Read-only	Current Negative DNS cache size
dnstotjumboqueries	<Double>	Read-only	Total number of Jumbo DNS queries received over UDP.
dnsjumboqueriesrate	<Double>	Read-only	Rate (/s) counter for dnstotjumboqueries
dnstotjumboanswers	<Double>	Read-only	Total number of Jumbo DNS responses sent over UDP.
dnsjumboanswersrate	<Double>	Read-only	Rate (/s) counter for dnstotjumboanswers
dnstotjumboserverresponses	<Double>	Read-only	Total number of Jumbo DNS

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
			responses received over UDP.
dnsjumboserverresponsesrate	<Double>	Read-only	Rate (/s) counter for dnstotjumboserverresponses
dnstotauthans	<Double>	Read-only	Number of queries which were authoritatively answered.
dnstotserverquery	<Double>	Read-only	Total number of Server queries sent.
dnsserverqueryrate	<Double>	Read-only	Rate (/s) counter for dnstotserverquery
dnstotcacheflush	<Double>	Read-only	Total number of times cache was flushed.
dnstotcacheentriesflush	<Double>	Read-only	Total number of cache entries flushed.
dnscurnoauthentries	<Double>	Read-only	Total number of non-authoritative entries.
dnscurauthentries	<Double>	Read-only	Total number of authoritative entries.
dnstotauthnonames	<Double>	Read-only	Number of queries for which no record was found.
dnstotunsupportedresponseclass	<Double>	Read-only	Total number of responses for which response types were unsupported.

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
dnstotinvalidqueryformat	<Double>	Read-only	Total number of queries whose format was invalid.
dnstotstrayanswer	<Double>	Read-only	Total number of stray answers.
dnstotresponsebadlen	<Double>	Read-only	Number of DNS responses received with invalid resoure data length.
dnstotreqrefusals	<Double>	Read-only	Number of DNS requests refused.
dnserrnullattack	<Double>	Read-only	Total number of queries received where all the counts are 0.
dnstotunsupportedresponsetype	<Double>	Read-only	Total number of responses for which response type requested was unsupported.
dnstotunsupportedqueryclass	<Double>	Read-only	Total number of queries for which query class was unsupported.
dnstotnonauthnodatas	<Double>	Read-only	Total number of responses for which there was a format error.
dnstotnodataresps	<Double>	Read-only	Number of DNS responses received without answer.

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
dnstotmultiquerydisableerror	<Double>	Read-only	Total number of times a multi query was disabled and received a multi query.
dnstotothererrors	<Double>	Read-only	Total number of other errors.
dns64totqueries	<Double>	Read-only	Total number of DNS64 queries recieved.
dns64queriesrate	<Double>	Read-only	Rate (/s) counter for dns64totqueries
dns64totanswers	<Double>	Read-only	Total number of DNS64 answers served.
dns64answersrate	<Double>	Read-only	Rate (/s) counter for dns64totanswers
dns64totrwanswers	<Double>	Read-only	Total number of DNS64 answers served after rewriting the response.
dns64rwanswersrate	<Double>	Read-only	Rate (/s) counter for dns64totrwanswers
dns64totresponses	<Double>	Read-only	Total number of responses recieved from backend in DNS64 context.
dns64responsesrate	<Double>	Read-only	Rate (/s) counter for dns64totresponses

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
dns64totgslbqueries	<Double>	Read-only	Total number of DNS64 queries for GSLB domain
dns64gslbqueriesrate	<Double>	Read-only	Rate (/s) counter for dns64totgslbqueries
dns64totgslbanswers	<Double>	Read-only	Total number of DNS64 queries served.
dns64gslbanswersrate	<Double>	Read-only	Rate (/s) counter for dns64totgslbanswers
dns64tottcanswers	<Double>	Read-only	Total number of Answers served with TC bit set in DNS64 context.
dns64tcanswersrate	<Double>	Read-only	Rate (/s) counter for dns64tottcanswers
dns64totsvraqueries	<Double>	Read-only	Total number of Queries sent by DNS64 module to backend.
dns64svraqueriesrate	<Double>	Read-only	Rate (/s) counter for dns64totsvraqueries
dns64totaaaabypass	<Double>	Read-only	Total number of times AAAA query has been bypassed in DNS64 trnsaction.
dns64aaaabypassrate	<Double>	Read-only	Rate (/s) counter for dns64totaaaabypass
dns64tottcpqueries	<Double>	Read-only	Total number of dns64 queries over TCP

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
dns64tcpqueriesrate	<Double>	Read-only	Rate (/s) counter for dns64tottcpqueries
dns64activepolicies	<Double>	Read-only	Total number of active dns64 policies
dns64totnodataresp	<Double>	Read-only	Total number of responses recieved from backend with ancount 0
dns64nodataresprate	<Double>	Read-only	Rate (/s) counter for dns64totnodataresp
dnstotnsrecqueries	<Double>	Read-only	Total number of NS queries received.
dnsnsrecqueriesrate	<Double>	Read-only	Rate (/s) counter for dnstotnsrecqueries
dnstotsoarecqueries	<Double>	Read-only	Total number of SOA queries received.
dnssoarecqueriesrate	<Double>	Read-only	Rate (/s) counter for dnstotsoarecqueries
dnstotptrrecqueries	<Double>	Read-only	Total number of PTR queries received.
dnsptrrecqueriesrate	<Double>	Read-only	Rate (/s) counter for dnstotptrrecqueries
dnstotsrvrecqueries	<Double>	Read-only	Total number of SRV queries received.
dnssrvrecqueriesrate	<Double>	Read-only	Rate (/s) counter for dnstotsrvrecqueries
dnstotaresponse	<Double>	Read-only	Total number of A responses received.

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
dnsaresponserate	<Double>	Read-only	Rate (/s) counter for dnstotaresponse
dnstotcnamerespons e	<Double>	Read-only	Total number of CNAME responses received.
dnscnameresponser ate	<Double>	Read-only	Rate (/s) counter for dnstotcnamerespons e
dnstotmxresponse	<Double>	Read-only	Total number of MX responses received.
dnsmxresponserate	<Double>	Read-only	Rate (/s) counter for dnstotmxresponse
dnstotanyresponse	<Double>	Read-only	Total number of ANY responses received.
dnsanyresponserate	<Double>	Read-only	Rate (/s) counter for dnstotanyresponse
dnstotnsrecupdate	<Double>	Read-only	Total number of NS record updates.
dnstotsoarecupdate	<Double>	Read-only	Total number of SOA record updates.
dnstotptrrecupdate	<Double>	Read-only	Total number of PTR record updates.
dnstotsrvrecupdate	<Double>	Read-only	Total number of SRV record updates.
dnstotaaaarecquerie s	<Double>	Read-only	Total number of AAAA queries received.

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
dnsaaaarecqueriesrate	<Double>	Read-only	Rate (/s) counter for dnstotaaaarecqueries
dnstotarecqueries	<Double>	Read-only	Total number of A queries received.
dnsarecqueriesrate	<Double>	Read-only	Rate (/s) counter for dnstotarecqueries
dnstotcnamerecqueries	<Double>	Read-only	Total number of CNAME queries received.
dnscnamerecqueriesrate	<Double>	Read-only	Rate (/s) counter for dnstotcnamerecqueries
dnstotmxrecqueries	<Double>	Read-only	Total number of MX queries received.
dnsmxrecqueriesrate	<Double>	Read-only	Rate (/s) counter for dnstotmxrecqueries
dnstotanyqueries	<Double>	Read-only	Total number of ANY queries received.
dnsanyqueriesrate	<Double>	Read-only	Rate (/s) counter for dnstotanyqueries
dnstotaaaaresponse	<Double>	Read-only	Total number of AAAA responses received.
dnsaaaaresponserate	<Double>	Read-only	Rate (/s) counter for dnstotaaaaresponse
dnstotnsresponse	<Double>	Read-only	Total number of NS responses received.

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
dnsnsresponserate	<Double>	Read-only	Rate (/s) counter for dnstotnsresponse
dnstotsoaresponse	<Double>	Read-only	Total number of SOA responses received.
dnssoaresponserate	<Double>	Read-only	Rate (/s) counter for dnstotsoaresponse
dnstotptrresponse	<Double>	Read-only	Total number of PTR responses received.
dnsptrresponserate	<Double>	Read-only	Rate (/s) counter for dnstotptrresponse
dnstotsrvresponse	<Double>	Read-only	Total number of SRV responses received.
dnssrvresponserate	<Double>	Read-only	Rate (/s) counter for dnstotsrvresponse
dnstotaaaarecupdate	<Double>	Read-only	Total number of AAAA record updates.
dnstotarecupdate	<Double>	Read-only	Total number of A record updates.
dnstotmxrecupdate	<Double>	Read-only	Total number of MX record updates.
dnstotcnamerecupdate	<Double>	Read-only	Total number of CNAME record updates.
dnscuraaaarecord	<Double>	Read-only	Total number of AAAA records.
dnscurarecord	<Double>	Read-only	Total number of A records.

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
dnscurmxrecord	<Double>	Read-only	Total number of MX records.
dnscurcnamerecord	<Double>	Read-only	Total number of CNAME records.
dnscurnsrecord	<Double>	Read-only	Total number of NS records.
dnscursoarecord	<Double>	Read-only	Total number of SOA records.
dnscurptrrecord	<Double>	Read-only	Total number of PTR records.
dnscursrvrecord	<Double>	Read-only	Total number of SRV records.
dnstotaaaarecfailed	<Double>	Read-only	Total number of times AAAA record lookup failed.
dnstotarecfailed	<Double>	Read-only	Total number of times A record lookup failed.
dnstotmxrecfailed	<Double>	Read-only	Total number of times MX record lookup failed.
dnstotptrrecfailed	<Double>	Read-only	Total number of times PTR record lookup failed.
dnstotnsrecfailed	<Double>	Read-only	Total number of times NS record lookup failed.
dnstotcnamerecfaile d	<Double>	Read-only	Total number of times CNAME record lookup failed.

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
dnstotsoarecfailed	<Double>	Read-only	Total number of times SOA record lookup failed.
dnstotsrvrecfailed	<Double>	Read-only	Total number of times SRV record lookup failed.
dnstotanyrecfailed	<Double>	Read-only	Total number of times ANY query lookup failed.
dnstotunsupportedqueries	<Double>	Read-only	Total number of requests for which query type requested was unsupported.

Operations

(click to see Properties)

GET (ALL)

Some options that you can use for each operations:

- **Getting warnings in response:**NITRO allows you to get warnings in an operation by specifying the "warning" query parameter as "yes". For example, to get warnings while connecting to the Citrix ADC appliance, the URL is as follows:

  http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/login?warning=yes

  If any, the warnings are displayed in the response payload with the HTTP code "209 X-NITRO-WARNING".

- **Authenticated access for individual NITRO operations:**NITRO allows you to logon to the Citrix ADC appliance to perform individual operations. You can use this option instead of creating a NITRO session (using the login object) and then using that session to perform all operations,

  To do this, you must specify the username and password in the request header of the NITRO request as follows:

X-NITRO-USER:<username>

X-NITRO-PASS:<password>

Note: In such cases, make sure that the request header DOES not include the following:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue>

`*Note:`*

Mandatory parameters are marked in redand placeholder content is marked in <green>.

get (all)

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/stat/dns

Query-parameters:

args

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/stat/dns?
args=detail:<Boolean_value>,fullvalues:<Boolean_value>,ntimes:<Double_value>,logfile:<String_value

Use this query-parameter to get dns resources based on additional properties.

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "dns": [ {

"dnstotcnameresponse":<Double_value>,
```

```
"dnstotjumboserverresponses":<Double_value>,

"dnsaresponserate":<Double_value>,

"dns64rwanswersrate":<Double_value>,

"dnstotaaaarecupdate":<Double_value>,

"dnstotptrrecupdate":<Double_value>,

"dns64nodatarespratе":<Double_value>,

"dnserrnullattack":<Double_value>,

"dnstotunsupportedqueries":<Double_value>,

"dnstotjumboqueries":<Double_value>,

"dnstotauthnonames":<Double_value>,

"dnstotmxrecupdate":<Double_value>,

"dnstotanyqueries":<Double_value>,

"dns64totrwanswers":<Double_value>,

"dnstotstrayanswer":<Double_value>,

"dnstotptrrecfailed":<Double_value>,

"dns64activepolicies":<Double_value>,

"dnstotcnamerecupdate":<Double_value>,

"dnscursoarecord":<Double_value>,

"dnstotcacheentriesflush":<Double_value>,

"dnsaaaaresponserate":<Double_value>,

"dnsaaaarecqueriesrate":<Double_value>,

"dnstotmxrecqueries":<Double_value>,
```

```
"dnstotsoarecqueries":<Double_value>,

"dnscurarecord":<Double_value>,

"dnstotinvalidqueryformat":<Double_value>,

"dnstotsrvrecfailed":<Double_value>,

"dnsarecqueriesrate":<Double_value>,

"dnstotsoaresponse":<Double_value>,

"dnstotserverresponse":<Double_value>,

"dnsanyqueriesrate":<Double_value>,

"dnscurauthentries":<Double_value>,

"dnstotmxresponse":<Double_value>,

"dnstotptrrecqueries":<Double_value>,

"dnstotunsupportedqueryclass":<Double_value>,

"dnsptrresponserate":<Double_value>,

"dnssoarecqueriesrate":<Double_value>,

"dns64totgslbanswers":<Double_value>,

"dns64answersrate":<Double_value>,

"dns64totqueries":<Double_value>,

"dnsptrrecqueriesrate":<Double_value>,

"dnstotsoarecfailed":<Double_value>,

"dns64tottcpqueries":<Double_value>,

"dnstotaaaarecqueries":<Double_value>,

"dns64responsesrate":<Double_value>,
```

```
"dnstotmxrecfailed":<Double_value>,

"dns64tottcanswers":<Double_value>,

"dnstotaaaarecfailed":<Double_value>,

"dnssrvresponserate":<Double_value>,

"dnsnsrecqueriesrate":<Double_value>,

"dnstotserverquery":<Double_value>,

"dnssoaresponserate":<Double_value>,

"dnstotmultiquery":<Double_value>,

"dnscuraaaarecord":<Double_value>,

"dnsqueriesrate":<Double_value>,

"dns64gslbqueriesrate":<Double_value>,

"dnsanyresponserate":<Double_value>,

"dnsanswersrate":<Double_value>,

"dnstotarecupdate":<Double_value>,

"dnscnameresponserate":<Double_value>,

"dnstotothererrors":<Double_value>,

"dnstotnsrecfailed":<Double_value>,

"dnscurcnamerecord":<Double_value>,

"dnscurnoauthentries":<Double_value>,

"dnstotresponsebadlen":<Double_value>,

"dns64totaaaabypass":<Double_value>,

"dns64tcpqueriesrate":<Double_value>,
```

```
"dnstotaaaaresponse":<Double_value>,

"dns64gslbanswersrate":<Double_value>,

"dnstotunsupportedresponsetype":<Double_value>,

"dns64totsvraqueries":<Double_value>,

"dns64totresponses":<Double_value>,

"dnstotnsrecqueries":<Double_value>,

"dns64totanswers":<Double_value>,

"dnscursrvrecord":<Double_value>,

"dnscurptrrecord":<Double_value>,

"dnstotanyresponse":<Double_value>,

"dnstotanyrecfailed":<Double_value>,

"dns64aaaabypassrate":<Double_value>,

"dnstotnsresponse":<Double_value>,

"dnscurcachesize":<Double_value>,

"dnscurnegcachesize":<Double_value>,

"dnssrvrecqueriesrate":<Double_value>,

"dnsjumboqueriesrate":<Double_value>,

"dnstotnsrecupdate":<Double_value>,

"dnstotcnamerecqueries":<Double_value>,

"dnstotmultiquerydisableerror":<Double_value>,

"dnstotarecqueries":<Double_value>,

"dnsserverresponserate":<Double_value>,
```

```
"dnsnsresponserate":<Double_value>,

"dnstotanswers":<Double_value>,

"dnsmxrecqueriesrate":<Double_value>,

"dnstotcnamerecfailed":<Double_value>,

"dnstotsrvrecqueries":<Double_value>,

"dnstotaresponse":<Double_value>,

"dnscnamerecqueriesrate":<Double_value>,

"dns64totnodataresp":<Double_value>,

"dnstotqueries":<Double_value>,

"dnstotsrvresponse":<Double_value>,

"dnstotunsupportedresponseclass":<Double_value>,

"dnstotjumboanswers":<Double_value>,

"dnstotcacheflush":<Double_value>,

"dnstotarecfailed":<Double_value>,

"dnstotsrvrecupdate":<Double_value>,

"dns64svraqueriesrate":<Double_value>,

"dnsserverqueryrate":<Double_value>,

"dnsjumboanswersrate":<Double_value>,

"dnsjumboserverresponsesrate":<Double_value>,

"dnscurnsrecord":<Double_value>,

"dnstotsoarecupdate":<Double_value>,

"dns64queriesrate":<Double_value>,
```

```
"dnstotnonauthnodatas":<Double_value>,

"dnstotauthans":<Double_value>,

"dnstotreqrefusals":<Double_value>,

"dnscurmxrecord":<Double_value>,

"dns64tcanswersrate":<Double_value>,

"dnstotrecupdate":<Double_value>,

"dns64totgslbqueries":<Double_value>,

"dnstotnodataresps":<Double_value>,

"dnstotptrresponse":<Double_value>,

"dnsmxresponserate":<Double_value>

}]}
```

dnspolicylabel

Statistics for dns policy label resource.

Properties

(click to see Operations)

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
labelname	<String>	Read-write	The name of the dns policy label for which statistics will be displayed. If not given statistics are shown for all dns policylabels. Minimum length = 1

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
clearstats	<String>	Read-write	Clear the statsistics / counters. Possible values = basic, full
pipolicylabelhits	<Double>	Read-only	Number of times policy label was invoked.
pipolicylabelhitsrate	<Double>	Read-only	Rate (/s) counter for pipolicylabelhits

Operations

(click to see Properties)

GET (ALL)| GET

Some options that you can use for each operations:

- **Getting warnings in response:**NITRO allows you to get warnings in an operation by specifying the "warning" query parameter as "yes". For example, to get warnings while connecting to the Citrix ADC appliance, the URL is as follows:

  http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/login?warning=yes

  If any, the warnings are displayed in the response payload with the HTTP code "209 X-NITRO-WARNING".

- **Authenticated access for individual NITRO operations:**NITRO allows you to logon to the Citrix ADC appliance to perform individual operations. You can use this option instead of creating a NITRO session (using the login object) and then using that session to perform all operations,

  To do this, you must specify the username and password in the request header of the NITRO request as follows:

  X-NITRO-USER:<username>

  X-NITRO-PASS:<password>

  Note: In such cases, make sure that the request header DOES not include the following:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue>

`*Note:`*

Mandatory parameters are marked in redand placeholder content is marked in <green>.

get (all)

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/stat/dnspolicylabel

Query-parameters:

args

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/stat/dnspolicylabel?
args=labelname:<String_value>,detail:<Boolean_value>,fullvalues:<Boolean_value>,ntimes:<Double_va

Use this query-parameter to get dnspolicylabel resources based on additional properties.

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "dnspolicylabel": [ {

"labelname":<String_value>,

"pipolicylabelhitsrate":<Double_value>,

"pipolicylabelhits":<Double_value>

}]}
```

get

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/stat/dnspolicylabel/labelname_value><String>

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "dnspolicylabel": [ {

"labelname":<String_value>,

"pipolicylabelhitsrate":<Double_value>,

"pipolicylabelhits":<Double_value>

}]}
```

dnsrecords

Read/write properties

Properties

(click to see Operations)

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
dnsrecordtype	<String>	Read-write	Display statistics for the specified DNS record or query type or, if a record or query type is not specified, statistics for all record types supported on the Citrix ADC.
clearstats	<String>	Read-write	Clear the statsistics / counters. Possible values = basic, full
dnstotentries	<Double>	Read-only	Total number of DNS record entries
dnstotupdates	<Double>	Read-only	Total number of DNS proactive updates
dnstotresponses	<Double>	Read-only	Total number of DNS server responses
dnstotrequests	<Double>	Read-only	Total number of DNS queries recieved
dnscurentries	<Double>	Read-only	Current number of DNS entries
dnstoterrlimits	<Double>	Read-only	Total number of times we have recieved dns record with more entries than we support
dnstoterrrespform	<Double>	Read-only	Total number of times we have recieved malformed responses from the backend

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
dnstoterraliasex	<Double>	Read-only	Total number of times we have recieved non-cname record for a domain for which an alias exists
dnstoterrnodomains	<Double>	Read-only	Total number of cache misses
dnscurrecords	<Double>	Read-only	Current number of DNS Records

Operations

(click to see Properties)

GET (ALL)| GET

Some options that you can use for each operations:

- **Getting warnings in response:**NITRO allows you to get warnings in an operation by specifying the "warning" query parameter as "yes". For example, to get warnings while connecting to the Citrix ADC appliance, the URL is as follows:

  http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/login?warning=yes

  If any, the warnings are displayed in the response payload with the HTTP code "209 X-NITRO-WARNING".

- **Authenticated access for individual NITRO operations:**NITRO allows you to logon to the Citrix ADC appliance to perform individual operations. You can use this option instead of creating a NITRO session (using the login object) and then using that session to perform all operations,

  To do this, you must specify the username and password in the request header of the NITRO request as follows:

  X-NITRO-USER:<username>

  X-NITRO-PASS:<password>

Note: In such cases, make sure that the request header DOES not include the following:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue>

`*Note:` *

Mandatory parameters are marked in redand placeholder content is marked in <green>.

get (all)

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/stat/dnsrecords

Query-parameters:

args

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/stat/dnsrecords?
args=dnsrecordtype:<String_value>,detail:<Boolean_value>,fullvalues:<Boolean_value>,ntimes:<Doub

Use this query-parameter to get dnsrecords resources based on additional properties.

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "dnsrecords": [ {

"dnsrecordtype":<String_value>,

"dnscurentries":<Double_value>,

"dnstotupdates":<Double_value>,
```

```
"dnstotrequests":<Double_value>,

"dnstoterraliasex":<Double_value>,

"dnstoterrrespform":<Double_value>,

"dnstotentries":<Double_value>,

"dnscurrecords":<Double_value>,

"dnstoterrnodomains":<Double_value>,

"dnstoterrlimits":<Double_value>,

"dnstotresponses":<Double_value>

}]}
```

get

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/stat/dnsrecords/dnsrecordtype_value><String>

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "dnsrecords": [ {
```

```
"dnsrecordtype":<String_value>,

"dnscurentries":<Double_value>,

"dnstotupdates":<Double_value>,

"dnstotrequests":<Double_value>,

"dnstoterraliasex":<Double_value>,

"dnstoterrrespform":<Double_value>,

"dnstotentries":<Double_value>,

"dnscurrecords":<Double_value>,

"dnstoterrnodomains":<Double_value>,

"dnstoterrlimits":<Double_value>,

"dnstotresponses":<Double_value>

}]}
```

Front-end-optimization

Front end optimization statistics. The system's feature to optimize Web content for performance.

NAME	DESCRIPTION
feo	Statistics for feo.

feo

Statistics for feo.

Properties

(click to see Operations)

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
clearstats	<String>	Read-write	Clear the statsistics / counters. Possible values = basic, full
optcacheobjects	<Double>	Read-only	Total number of optimized cache objects ready to be served.
optcacheobjectsrate	<Double>	Read-only	Rate (/s) counter for optcacheobjects
origcacheobjects	<Double>	Read-only	Total number of original cache objects ready to be served.
origcacheobjectsrate	<Double>	Read-only	Rate (/s) counter for origcacheobjects
totalimgsdomainsharded	<Double>	Read-only	Total no of images whose domain has been set from shards.
imgsdomainshardedrate	<Double>	Read-only	Rate (/s) counter for totalimgsdomainsharded
totalimgslazyloaded	<Double>	Read-only	Total number of images modified for lazy loading.
imgslazyloadedrate	<Double>	Read-only	Rate (/s) counter for totalimgslazyloaded
toturireplaced	<Double>	Read-only	Total number of URI replaced.

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
urireplacedrate	<Double>	Read-only	Rate (/s) counter for toturireplaced
totalimgsinlinedincss	<Double>	Read-only	Total number of images inlined in CSS.
imgsinlinedincssrate	<Double>	Read-only	Rate (/s) counter for totalimgsinlinedincss
totalinlinedjs	<Double>	Read-only	Total number of inlined JS files.
inlinedjsrate	<Double>	Read-only	Rate (/s) counter for totalinlinedjs
totalinlinedcss	<Double>	Read-only	Total number of inlined CSS files.
inlinedcssrate	<Double>	Read-only	Rate (/s) counter for totalinlinedcss
totalinlinedimgs	<Double>	Read-only	Total number of inlined images in HTML.
inlinedimgsrate	<Double>	Read-only	Rate (/s) counter for totalinlinedimgs
totaldatasavings	<Double>	Read-only	Total data savings in bytes.
datasavingsrate	<Double>	Read-only	Rate (/s) counter for totaldatasavings
htmlcommentsremoved	<Double>	Read-only	The total number of HTML comments removed.

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
htmlcommentsremovedrate	<Double>	Read-only	Rate (/s) counter for htmlcommentsremoved
totalcacheextended	<Double>	Read-only	The total number of objects cache extended.
cacheextendedrate	<Double>	Read-only	Rate (/s) counter for totalcacheextended
totalcsscombined	<Double>	Read-only	The total number of CSS combined.
csscombinedrate	<Double>	Read-only	Rate (/s) counter for totalcsscombined
totalimporttolink	<Double>	Read-only	The total number of CSS imports converted to links
importtolinkrate	<Double>	Read-only	Rate (/s) counter for totalimporttolink
totaljsmoved	<Double>	Read-only	The total number of JS moved to end.
jsmovedrate	<Double>	Read-only	Rate (/s) counter for totaljsmoved
totalcssmoved	<Double>	Read-only	The total number of CSS moved to head.
cssmovedrate	<Double>	Read-only	Rate (/s) counter for totalcssmoved
totaljsmin	<Double>	Read-only	The total number of JS files minified.
jsminrate	<Double>	Read-only	Rate (/s) counter for totaljsmin

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
totalcssmin	<Double>	Read-only	The total number of CSS files minified.
cssminrate	<Double>	Read-only	Rate (/s) counter for totalcssmin
totalimgstojxr	<Double>	Read-only	The total number of images converted to JPEGXR format.
imgstojxrrate	<Double>	Read-only	Rate (/s) counter for totalimgstojxr
totalimgstowebp	<Double>	Read-only	The total number of images converted to WEBP format.
imgstowebprate	<Double>	Read-only	Rate (/s) counter for totalimgstowebp
totaljpegsoptimized	<Double>	Read-only	The total number of JPEG format images optimized.
jpegsoptimizedrate	<Double>	Read-only	Rate (/s) counter for totaljpegsoptimized
totalgifstopng	<Double>	Read-only	The total number of images converted from GIF to PNG format.
gifstopngrate	<Double>	Read-only	Rate (/s) counter for totalgifstopng
totalimgsresized	<Double>	Read-only	The total number of images resized to dimensions in the <img> tag.

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
imgsresizedrate	<Double>	Read-only	Rate (/s) counter for totalimgsresized

Operations

(click to see Properties)

GET (ALL)

Some options that you can use for each operations:

- **Getting warnings in response:**NITRO allows you to get warnings in an operation by specifying the "warning" query parameter as "yes". For example, to get warnings while connecting to the Citrix ADC appliance, the URL is as follows:

  http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/login?warning=yes

  If any, the warnings are displayed in the response payload with the HTTP code "209 X-NITRO-WARNING".
- **Authenticated access for individual NITRO operations:**NITRO allows you to logon to the Citrix ADC appliance to perform individual operations. You can use this option instead of creating a NITRO session (using the login object) and then using that session to perform all operations,

  To do this, you must specify the username and password in the request header of the NITRO request as follows:

  X-NITRO-USER:<username>

  X-NITRO-PASS:<password>

  Note: In such cases, make sure that the request header DOES not include the following:

  Cookie:NITRO_AUTH_TOKEN=<tokenvalue>

`*Note:`*

Mandatory parameters are marked in redand placeholder content is marked in <green>.

get (all)

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/stat/feo

Query-parameters:

args

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/stat/feo?
args=detail:<Boolean_value>,fullvalues:<Boolean_value>,ntimes:<Double_value>,logfile:<String_value

Use this query-parameter to get feo resources based on additional properties.

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "feo": [ {

"totaljsmoved":<Double_value>,

"totalimgslazyloaded":<Double_value>,

"htmlcommentsremovedrate":<Double_value>,

"urireplacedrate":<Double_value>,

"cssmovedrate":<Double_value>,

"htmlcommentsremoved":<Double_value>,

"jsmovedrate":<Double_value>,

"totalimgsinlinedincss":<Double_value>,
```

```
"totalinlinedcss":<Double_value>,

"inlinedimgsrate":<Double_value>,

"toturireplaced":<Double_value>,

"imgstowebprate":<Double_value>,

"totalinlinedjs":<Double_value>,

"totalcssmoved":<Double_value>,

"csscombinedrate":<Double_value>,

"imgslazyloadedrate":<Double_value>,

"totaljsmin":<Double_value>,

"optcacheobjects":<Double_value>,

"totalgifstopng":<Double_value>,

"totalimgsresized":<Double_value>,

"imgsinlinedincssrate":<Double_value>,

"inlinedcssrate":<Double_value>,

"totalimgstowebp":<Double_value>,

"inlinedjsrate":<Double_value>,

"origcacheobjectsrate":<Double_value>,

"cssminrate":<Double_value>,

"totalimgsdomainsharded":<Double_value>,

"imgsdomainshardedrate":<Double_value>,

"gifstopngrate":<Double_value>,

"importtolinkrate":<Double_value>,
```

```
"totalcsscombined":<Double_value>,

"totalimgstojxr":<Double_value>,

"imgsresizedrate":<Double_value>,

"optcacheobjectsrate":<Double_value>,

"totaljpegsoptimized":<Double_value>,

"totalcacheextended":<Double_value>,

"origcacheobjects":<Double_value>,

"jsminrate":<Double_value>,

"jpegsoptimizedrate":<Double_value>,

"totaldatasavings":<Double_value>,

"imgstojxrrate":<Double_value>,

"datasavingsrate":<Double_value>,

"cacheextendedrate":<Double_value>,

"totalcssmin":<Double_value>,

"totalinlinedimgs":<Double_value>,

"totalimporttolink":<Double_value>

}]}
```

Global Server Load Balancing

Global Server Load Balancing (GSLB) statistics. GSLB feature ensures that client requests are directed to a best performing site available in a global enterprise and distributed Internet environment.

NAME	DESCRIPTION
gslbdomain	Statistics for GSLB domain resource.
gslbservice	Statistics for GSLB service resource.
gslbservicegroup	Statistics for GSLB service group resource.
gslbservicegroupmember	Statistics for GSLB service group entity resource.
gslbsite	Statistics for GSLB site resource.
gslbvserver	Statistics for Global Server Load Balancing Virtual Server resource.

gslbdomain

Statistics for GSLB domain resource.

Properties

(click to see Operations)

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
name	<String>	Read-write	Name of the GSLB domain for which to display statistics. If you do not specify a name, statistics are shown for all configured GSLB domains. Minimum length = 1
dnsrecordtype	<String>	Read-write	. Possible values = A, AAAA, CNAME, NAPTR

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
clearstats	<String>	Read-write	Clear the statsistics / counters. Possible values = basic, full
dnstotalqueries	<Double>	Read-only	Total number of DNS queries received.
dnsqueriesrate	<Double>	Read-only	Rate (/s) counter for dnstotalqueries
gslbdnsrectype	<String>	Read-only	Type of DNS record returned

Operations

(click to see Properties)

GET (ALL)| GET

Some options that you can use for each operations:

- **Getting warnings in response:**NITRO allows you to get warnings in an operation by specifying the "warning" query parameter as "yes". For example, to get warnings while connecting to the Citrix ADC appliance, the URL is as follows:

  http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/login?warning=yes

  If any, the warnings are displayed in the response payload with the HTTP code "209 X-NITRO-WARNING".

- **Authenticated access for individual NITRO operations:**NITRO allows you to logon to the Citrix ADC appliance to perform individual operations. You can use this option instead of creating a NITRO session (using the login object) and then using that session to perform all operations,

  To do this, you must specify the username and password in the request header of the NITRO request as follows:

  X-NITRO-USER:<username>

  X-NITRO-PASS:<password>

Note: In such cases, make sure that the request header DOES not include the following:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue>

`*Note:`*

Mandatory parameters are marked in redand placeholder content is marked in <green>.

get (all)

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/stat/gslbdomain

Query-parameters:

args

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/stat/gslbdomain?
args=name:<String_value>,dnsrecordtype:<String_value>,dnsrecordtype:<String_value>,detail:<Boolea

Use this query-parameter to get gslbdomain resources based on additional properties.

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "gslbdomain": [ {

"name":<String_value>,

"dnsqueriesrate":<Double_value>,

"dnstotalqueries":<Double_value>,
```

```
"gslbdnsrectype":<String_value>

}]}
```

get

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/stat/gslbdomain/name_value><String>

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "gslbdomain": [ {

"name":<String_value>,

"dnsqueriesrate":<Double_value>,

"dnstotalqueries":<Double_value>,

"gslbdnsrectype":<String_value>

}]}
```

gslbservice

Statistics for GSLB service resource.

Properties

(click to see Operations)

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
servicename	<String>	Read-write	Name of the GSLB service. Minimum length = 1
clearstats	<String>	Read-write	Clear the statsistics / counters. Possible values = basic, full
establishedconn	<Double>	Read-only	Number of client connections in ESTABLISHED state.
primaryipaddress	<String>	Read-only	The IP address on which the service is running.
primaryport	<Integer>	Read-only	The port on which the service is running.
servicetype	<String>	Read-only	The service type of this service.Possible values are ADNS, DNS, MYSQL, RTSP, SSL_DIAMETER, ADNS_TCP, DNS_TCP, NNTP, SIP_UDP, SSL_TCP, ANY, FTP, RADIUS, SNMP, TCP, DHCPRA, HTTP, RDP, SSL, TFTP, DIAMETER,

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
			MSSQL, RPCSVR, SSL_BRIDGE, UDP
state	<String>	Read-only	Current state of the server. There are seven possible values: UP(7), DOWN(1), UNKNOWN(2), BUSY(3), OFS(Out of Service)(4), TROFS(Transition Out of Service)(5), TROFS_DOWN(Down When going Out of Service)(8)
totalrequestbytes	<Double>	Read-only	Total number of request bytes received on this service or virtual server.
requestbytesrate	<Double>	Read-only	Rate (/s) counter for totalrequestbytes
totalresponsebytes	<Double>	Read-only	Number of response bytes received by this service or virtual server.
responsebytesrate	<Double>	Read-only	Rate (/s) counter for totalresponsebytes
curload	<Double>	Read-only	Load on the service that is calculated from the bound load based monitor.
totalrequests	<Double>	Read-only	Total number of requests received on this service or virtual

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
			server. (This applies to HTTP/SSL services and servers.)
requestsrate	<Double>	Read-only	Rate (/s) counter for totalrequests
totalresponses	<Double>	Read-only	Number of responses received on this service or virtual server. (This applies to HTTP/SSL services and servers.)
responsesrate	<Double>	Read-only	Rate (/s) counter for totalresponses
curclntconnections	<Double>	Read-only	Number of current client connections.
cursrvrconnections	<Double>	Read-only	Number of current connections to the actual servers behind the virtual server.
vsvrservicehits	<Double>	Read-only	Number of times that the service has been provided.
vsvrservicehitsrate	<Double>	Read-only	Rate (/s) counter for vsvrservicehits

Operations

(click to see Properties)

GET (ALL)| GET

Some options that you can use for each operations:

- **Getting warnings in response:**NITRO allows you to get warnings in an operation by specifying the "warning" query parameter as "yes". For example, to get warnings while connecting to the Citrix ADC appliance, the URL is as follows:

  http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/login?warning=yes

  If any, the warnings are displayed in the response payload with the HTTP code "209 X-NITRO-WARNING".

- **Authenticated access for individual NITRO operations:**NITRO allows you to logon to the Citrix ADC appliance to perform individual operations. You can use this option instead of creating a NITRO session (using the login object) and then using that session to perform all operations,

  To do this, you must specify the username and password in the request header of the NITRO request as follows:

  X-NITRO-USER:<username>

  X-NITRO-PASS:<password>

  Note: In such cases, make sure that the request header DOES not include the following:

  Cookie:NITRO_AUTH_TOKEN=<tokenvalue>

`*Note:`*

Mandatory parameters are marked in redand placeholder content is marked in <green>.

get (all)

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/stat/gslbservice

Query-parameters:

args

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/stat/gslbservice?
args=servicename:<String_value>,detail:<Boolean_value>,fullvalues:<Boolean_value>,ntimes:<Double

Use this query-parameter to get gslbservice resources based on additional properties.

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "gslbservice": [ {

"servicename":<String_value>,

"curclntconnections":<Double_value>,

"servicetype":<String_value>,

"establishedconn":<Double_value>,

"totalrequests":<Double_value>,

"responsebytesrate":<Double_value>,

"totalresponses":<Double_value>,

"requestbytesrate":<Double_value>,

"cursrvrconnections":<Double_value>,

"primaryipaddress":<String_value>,

"responsesrate":<Double_value>,

"curload":<Double_value>,

"totalrequestbytes":<Double_value>,

"state":<String_value>,
```

```
"vsvrservicehits":<Double_value>,

"totalresponsebytes":<Double_value>,

"primaryport":<Integer_value>,

"requestsrate":<Double_value>,

"vsvrservicehitsrate":<Double_value>

}]}
```

get

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/stat/gslbservice/servicename_value><String>

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "gslbservice": [ {

"servicename":<String_value>,

"curclntconnections":<Double_value>,

"servicetype":<String_value>,
```

```
"establishedconn":<Double_value>,

"totalrequests":<Double_value>,

"responsebytesrate":<Double_value>,

"totalresponses":<Double_value>,

"requestbytesrate":<Double_value>,

"cursrvrconnections":<Double_value>,

"primaryipaddress":<String_value>,

"responsesrate":<Double_value>,

"curload":<Double_value>,

"totalrequestbytes":<Double_value>,

"state":<String_value>,

"vsvrservicehits":<Double_value>,

"totalresponsebytes":<Double_value>,

"primaryport":<Integer_value>,

"requestsrate":<Double_value>,

"vsvrservicehitsrate":<Double_value>

}]}
```

gslbsite

Statistics for GSLB site resource.

Properties

(click to see Operations)

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
sitename	<String>	Read-write	Name of the GSLB site for which to display detailed statistics. If a name is not specified, basic information about all GSLB sites is displayed. Minimum length = 1
clearstats	<String>	Read-write	Clear the statsistics / counters. Possible values = basic, full
sitepublicip	<String>	Read-only	The public IP address of this GSLB site.
siteip	<String>	Read-only	The private IP address of this GSLB site.
sitemepstatus	<String>	Read-only	Indicates the status of the Metric Exchange Policy at this GSLB site.
persexchange	<String>	Read-only	Indicates whether Persistence entries exchange is enabled or disabled at this GSLB site.
nwmetricexchange	<String>	Read-only	Indicates whether network metric exchange is enabled or disabled at this GSLB site.

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
sitemetricexchange	<String>	Read-only	Indicates whether metric exchange is enabled or disabled at this GSLB site.
sitetype	<String>	Read-only	Indicates whether this GSLB site is local or remote.
siteipstr	<String>	Read-only	The private IP address of this GSLB site.
sitepublicipstr	<String>	Read-only	The public IP address of this GSLB site.
sitemetricmepstatus	<String>	Read-only	Indicates the status of the site metric Metric Exchange connection at this GSLB site.
nwmetricmepstatus	<String>	Read-only	Indicates the status of the network metric Metric Exchange connection at this GSLB site.
sitetotalrequestbytes	<Double>	Read-only	Total number of request bytes received by the virtual servers represented by all GSLB services associated with this GSLB site.
siterequestbytesrate	<Double>	Read-only	Rate (/s) counter for sitetotalrequestbytes

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
sitetotalresponsebytes	<Double>	Read-only	Number of response bytes received by the virtual servers represented by all GSLB services associated with this GSLB site.
siteresponsebytesrate	<Double>	Read-only	Rate (/s) counter for sitetotalresponsebytes
sitetotalrequests	<Double>	Read-only	Total number of requests received by the virtual servers represented by all GSLB services associated with this GSLB site.
siterequestsrate	<Double>	Read-only	Rate (/s) counter for sitetotalrequests
sitetotalresponses	<Double>	Read-only	Number of responses received by the virtual servers represented by all GSLB services associated with this GSLB site.
siteresponsesrate	<Double>	Read-only	Rate (/s) counter for sitetotalresponses
sitecurclntconnections	<Double>	Read-only	Number of current client connections to the virtual servers represented by all GSLB services associated with this GSLB site.

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
sitecursrvrconnections	<Double>	Read-only	Number of current connections to the real servers behind the virtual servers represented by all GSLB services associated with this GSLB site.

Operations

(click to see Properties)

GET (ALL)| GET

Some options that you can use for each operations:

- **Getting warnings in response:**NITRO allows you to get warnings in an operation by specifying the "warning" query parameter as "yes". For example, to get warnings while connecting to the Citrix ADC appliance, the URL is as follows:

  http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/login?warning=yes

  If any, the warnings are displayed in the response payload with the HTTP code "209 X-NITRO-WARNING".
- **Authenticated access for individual NITRO operations:**NITRO allows you to logon to the Citrix ADC appliance to perform individual operations. You can use this option instead of creating a NITRO session (using the login object) and then using that session to perform all operations,

  To do this, you must specify the username and password in the request header of the NITRO request as follows:

  X-NITRO-USER:<username>

  X-NITRO-PASS:<password>

  Note: In such cases, make sure that the request header DOES not include the following:

  Cookie:NITRO_AUTH_TOKEN=<tokenvalue>

`*Note:` *

Mandatory parameters are marked in redand placeholder content is marked in <green>.

get (all)

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/stat/gslbsite

Query-parameters:

args

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/stat/gslbsite?
args=sitename:<String_value>,detail:<Boolean_value>,fullvalues:<Boolean_value>,ntimes:<Double_val

Use this query-parameter to get gslbsite resources based on additional properties.

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "gslbsite": [ {

"sitename":<String_value>,

"siteresponsesrate":<Double_value>,

"siteipstr":<String_value>,

"sitetotalrequests":<Double_value>,
```

```
"siterequestsrate":<Double_value>,

"nwmetricmepstatus":<String_value>,

"sitemetricexchange":<String_value>,

"sitepublicip":<String_value>,

"sitetotalresponsebytes":<Double_value>,

"sitecursrvrconnections":<Double_value>,

"sitepublicipstr":<String_value>,

"sitetotalrequestbytes":<Double_value>,

"siteip":<String_value>,

"sitemepstatus":<String_value>,

"sitecurclntconnections":<Double_value>,

"sitetype":<String_value>,

"nwmetricexchange":<String_value>,

"persexchange":<String_value>,

"siterequestbytesrate":<Double_value>,

"siteresponsebytesrate":<Double_value>,

"sitetotalresponses":<Double_value>,

"sitemetricmepstatus":<String_value>

}]}
```

get

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/stat/gslbsite/sitename_value><String>

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "gslbsite": [ {

"sitename":<String_value>,

"siteresponsesrate":<Double_value>,

"siteipstr":<String_value>,

"sitetotalrequests":<Double_value>,

"siterequestsrate":<Double_value>,

"nwmetricmepstatus":<String_value>,

"sitemetricexchange":<String_value>,

"sitepublicip":<String_value>,

"sitetotalresponsebytes":<Double_value>,

"sitecursrvrconnections":<Double_value>,

"sitepublicipstr":<String_value>,

"sitetotalrequestbytes":<Double_value>,
```

```
"siteip":<String_value>,

"sitemepstatus":<String_value>,

"sitecurclntconnections":<Double_value>,

"sitetype":<String_value>,

"nwmetricexchange":<String_value>,

"persexchange":<String_value>,

"siterequestbytesrate":<Double_value>,

"siteresponsebytesrate":<Double_value>,

"sitetotalresponses":<Double_value>,

"sitemetricmepstatus":<String_value>

}]}
```

gslbvserver

Statistics for Global Server Load Balancing Virtual Server resource.

Properties

(click to see Operations)

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
name	<String>	Read-write	Name of the GSLB virtual server for which to display statistics. If you do not specify a name, statistics are displayed for all GSLB virtual servers. Minimum length = 1

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
clearstats	<String>	Read-write	Clear the statsistics / counters. Possible values = basic, full
establishedconn	<Double>	Read-only	Number of client connections in ESTABLISHED state.
inactsvcs	<Double>	Read-only	number of INACTIVE services bound to a vserver
vslbhealth	<Double>	Read-only	Health of the vserver. This gives percentage of UP services bound to this vserver.
type	<String>	Read-only	Protocol associated with the vserver
state	<String>	Read-only	Current state of the server. There are seven possible values: UP(7), DOWN(1), UNKNOWN(2), BUSY(3), OFS(Out of Service)(4), TROFS(Transition Out of Service)(5), TROFS_DOWN(Down When going Out of Service)(8)
actsvcs	<Double>	Read-only	number of ACTIVE services bound to a vserver

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
tothits	<Double>	Read-only	Total vserver hits
hitsrate	<Double>	Read-only	Rate (/s) counter for tothits
vsvrtotbkplbhits	<Double>	Read-only	Total backup LB method hits
vsvrtotbkplbfail	<Double>	Read-only	Total backup LB method fails
curpersistencesessions	<Double>	Read-only	current vserver owned persistence sessions
vsvrtotpersistencehits	<Double>	Read-only	Total number of Persistence hits at vserver
totalrequestbytes	<Double>	Read-only	Total number of request bytes received on this service or virtual server.
requestbytesrate	<Double>	Read-only	Rate (/s) counter for totalrequestbytes
totalresponsebytes	<Double>	Read-only	Number of response bytes received by this service or virtual server.
responsebytesrate	<Double>	Read-only	Rate (/s) counter for totalresponsebytes
sothreshold	<Double>	Read-only	Spill Over Threshold set on the VServer.

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
totspillovers	<Double>	Read-only	Number of times vserver experienced spill over.
totvserverdownbackuphits	<Double>	Read-only	Number of times traffic was diverted to backup vserver since primary vserver was DOWN.
totalrequests	<Double>	Read-only	Total number of requests received on this service or virtual server. (This applies to HTTP/SSL services and servers.)
requestsrate	<Double>	Read-only	Rate (/s) counter for totalrequests
totalresponses	<Double>	Read-only	Number of responses received on this service or virtual server. (This applies to HTTP/SSL services and servers.)
responsesrate	<Double>	Read-only	Rate (/s) counter for totalresponses
curclntconnections	<Double>	Read-only	Number of current client connections.
cursrvrconnections	<Double>	Read-only	Number of current connections to the actual servers behind the virtual server.

Operations

(click to see Properties)

GET (ALL)| GET

Some options that you can use for each operations:

- **Getting warnings in response:**NITRO allows you to get warnings in an operation by specifying the "warning" query parameter as "yes". For example, to get warnings while connecting to the Citrix ADC appliance, the URL is as follows:

  http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/login?warning=yes

  If any, the warnings are displayed in the response payload with the HTTP code "209 X-NITRO-WARNING".

- **Authenticated access for individual NITRO operations:**NITRO allows you to logon to the Citrix ADC appliance to perform individual operations. You can use this option instead of creating a NITRO session (using the login object) and then using that session to perform all operations,

  To do this, you must specify the username and password in the request header of the NITRO request as follows:

  X-NITRO-USER:<username>

  X-NITRO-PASS:<password>

  Note: In such cases, make sure that the request header DOES not include the following:

  Cookie:NITRO_AUTH_TOKEN=<tokenvalue>

`*Note:`*

Mandatory parameters are marked in redand placeholder content is marked in <green>.

get (all)

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/stat/gslbvserver

Query-parameters:

args

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/stat/gslbvserver?
args=name:<String_value>,detail:<Boolean_value>,fullvalues:<Boolean_value>,ntimes:<Double_value>

Use this query-parameter to get gslbvserver resources based on additional properties.

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "gslbvserver": [ {

"name":<String_value>,

"curclntconnections":<Double_value>,

"vsvrtotbkplbfail":<Double_value>,

"vsvrtotbkplbhits":<Double_value>,

"establishedconn":<Double_value>,

"tothits":<Double_value>,

"totalrequests":<Double_value>,

"sothreshold":<Double_value>,

"responsebytesrate":<Double_value>,

"totalresponses":<Double_value>,
```

```
"curpersistencesessions":<Double_value>,

"requestbytesrate":<Double_value>,

"type":<String_value>,

"hitsrate":<Double_value>,

"cursrvrconnections":<Double_value>,

"responsesrate":<Double_value>,

"totspillovers":<Double_value>,

"totalrequestbytes":<Double_value>,

"state":<String_value>,

"vslbhealth":<Double_value>,

"actsvcs":<Double_value>,

"totalresponsebytes":<Double_value>,

"vsvrtotpersistencehits":<Double_value>,

"requestsrate":<Double_value>,

"totvserverdownbackuphits":<Double_value>,

"inactsvcs":<Double_value>

}]}
```

get

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/stat/gslbvserver/name_value><String>

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "gslbvserver": [ {

"name":<String_value>,

"curclntconnections":<Double_value>,

"vsvrtotbkplbfail":<Double_value>,

"vsvrtotbkplbhits":<Double_value>,

"establishedconn":<Double_value>,

"tothits":<Double_value>,

"totalrequests":<Double_value>,

"sothreshold":<Double_value>,

"responsebytesrate":<Double_value>,

"totalresponses":<Double_value>,

"curpersistencesessions":<Double_value>,

"requestbytesrate":<Double_value>,

"type":<String_value>,

"hitsrate":<Double_value>,
```

```
"cursrvrconnections":<Double_value>,

"responsesrate":<Double_value>,

"totspillovers":<Double_value>,

"totalrequestbytes":<Double_value>,

"state":<String_value>,

"vslbhealth":<Double_value>,

"actsvcs":<Double_value>,

"totalresponsebytes":<Double_value>,

"vsvrtotpersistencehits":<Double_value>,

"requestsrate":<Double_value>,

"totvserverdownbackuphits":<Double_value>,

"inactsvcs":<Double_value>

}]}
```

gslbservicegroup

Statistics for GSLB service group resource.

Properties

(click to see Operations)

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
servicegroupname	<String>	Read-write	Name of the GSLB service group for which to display settings. Minimum length = 1

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
clearstats	<String>	Read-write	Clear the statsistics / counters. Possible values = basic, full
state	<String>	Read-only	Current state of the server. There are seven possible values: UP(7), DOWN(1), UNKNOWN(2), BUSY(3), OFS(Out of Service)(4), TROFS(Transition Out of Service)(5), TROFS_DOWN(Down When going Out of Service)(8)
servicetype	<String>	Read-only	The service type of this service.Possible values are ADNS, DNS, MYSQL, RTSP, SSL_DIAMETER, ADNS_TCP, DNS_TCP, NNTP, SIP_UDP, SSL_TCP, ANY, FTP, RADIUS, SNMP, TCP, DHCPRA, HTTP, RDP, SSL, TFTP, DIAMETER, MSSQL, RPCSVR, SSL_BRIDGE, UDP

Operations

(click to see Properties)

GET (ALL)| GET

Some options that you can use for each operations:

- **Getting warnings in response:**NITRO allows you to get warnings in an operation by specifying the "warning" query parameter as "yes". For example, to get warnings while connecting to the Citrix ADC appliance, the URL is as follows:

  http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/login?warning=yes

  If any, the warnings are displayed in the response payload with the HTTP code "209 X-NITRO-WARNING".

- **Authenticated access for individual NITRO operations:**NITRO allows you to logon to the Citrix ADC appliance to perform individual operations. You can use this option instead of creating a NITRO session (using the login object) and then using that session to perform all operations,

  To do this, you must specify the username and password in the request header of the NITRO request as follows:

  X-NITRO-USER:<username>

  X-NITRO-PASS:<password>

  Note: In such cases, make sure that the request header DOES not include the following:

  Cookie:NITRO_AUTH_TOKEN=<tokenvalue>

`*Note:`*

Mandatory parameters are marked in redand placeholder content is marked in <green>.

get (all)

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/stat/gslbservicegroup

Query-parameters:

args

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/stat/gslbservicegroup?
args=servicegroupname:<String_value>,detail:<Boolean_value>,fullvalues:<Boolean_value>,ntimes:<D

Use this query-parameter to get gslbservicegroup resources based on additional properties.

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "gslbservicegroup": [ {

"servicegroupname":<String_value>,

"servicetype":<String_value>,

"state":<String_value>

}]}
```

get

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/stat/gslbservicegroup/servicegroupname_value><String>

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "gslbservicegroup": [ {

"servicegroupname":<String_value>,

"servicetype":<String_value>,

"state":<String_value>

}]}
```

gslbservicegroupmember

Statistics for GSLB service group entity resource.

Properties

(click to see Operations)

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
servicegroupname	<String>	Read-write	Displays statistics for the specified GSLB service group.Name of the GSLB service group. Must begin with an ASCII alphanumeric or underscore (_) character, and must contain only ASCII alphanumeric, underscore, hash (#), period (.), space, colon (:), at sign (@), equal sign (=), and hyphen (-)

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
			characters.CLI Users: If the name includes one or more spaces, enclose the name in double or single quotation marks (for example, "my servicegroup" or 'my servicegroup'). Minimum length = 1
ip	<String>	Read-write	IP address of the GSLB service group. Mutually exclusive with the server name parameter.
servername	<String>	Read-write	Name of the server. Mutually exclusive with the IP address parameter. Minimum length = 1
port	<Integer>	Read-write	Port number of the service group member. Range 1 - 65535 * in CLI is represented as 65535 in NITRO API
clearstats	<String>	Read-write	Clear the statsistics / counters. Possible values = basic, full
establishedconn	<Double>	Read-only	Number of client connections in

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
			ESTABLISHED state.
primaryipaddress	<String>	Read-only	The IP address on which the service is running.
primaryport	<Integer>	Read-only	The port on which the service is running.
servicetype	<String>	Read-only	The service type of this service.Possible values are ADNS, DNS, MYSQL, RTSP, SSL_DIAMETER, ADNS_TCP, DNS_TCP, NNTP, SIP_UDP, SSL_TCP, ANY, FTP, RADIUS, SNMP, TCP, DHCPRA, HTTP, RDP, SSL, TFTP, DIAMETER, MSSQL, RPCSVR, SSL_BRIDGE, UDP
state	<String>	Read-only	Current state of the server. There are seven possible values: UP(7), DOWN(1), UNKNOWN(2), BUSY(3), OFS(Out of Service)(4), TROFS(Transition Out of Service)(5), TROFS_DOWN(Down When going Out of Service)(8)

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
totalrequestbytes	<Double>	Read-only	Total number of request bytes received on this service or virtual server.
requestbytesrate	<Double>	Read-only	Rate (/s) counter for totalrequestbytes
totalresponsebytes	<Double>	Read-only	Number of response bytes received by this service or virtual server.
responsebytesrate	<Double>	Read-only	Rate (/s) counter for totalresponsebytes
curload	<Double>	Read-only	Load on the service that is calculated from the bound load based monitor.
totalrequests	<Double>	Read-only	Total number of requests received on this service or virtual server. (This applies to HTTP/SSL services and servers.)
requestsrate	<Double>	Read-only	Rate (/s) counter for totalrequests
totalresponses	<Double>	Read-only	Number of responses received on this service or virtual server. (This applies to HTTP/SSL services and servers.)

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
responsesrate	<Double>	Read-only	Rate (/s) counter for totalresponses
curclntconnections	<Double>	Read-only	Number of current client connections.
cursrvrconnections	<Double>	Read-only	Number of current connections to the actual servers behind the virtual server.

Operations

(click to see Properties)

GET (ALL)

Some options that you can use for each operations:

- **Getting warnings in response:**NITRO allows you to get warnings in an operation by specifying the "warning" query parameter as "yes". For example, to get warnings while connecting to the Citrix ADC appliance, the URL is as follows:

  http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/login?warning=yes

  If any, the warnings are displayed in the response payload with the HTTP code "209 X-NITRO-WARNING".

- **Authenticated access for individual NITRO operations:**NITRO allows you to logon to the Citrix ADC appliance to perform individual operations. You can use this option instead of creating a NITRO session (using the login object) and then using that session to perform all operations,

  To do this, you must specify the username and password in the request header of the NITRO request as follows:

  X-NITRO-USER:<username>

  X-NITRO-PASS:<password>

  Note: In such cases, make sure that the request header DOES not include the following:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue>

`*Note:`*

Mandatory parameters are marked in redand placeholder content is marked in <green>.

get (all)

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/stat/gslbservicegroupmember

Query-parameters:

args

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/stat/gslbservicegroupmember?
args=servicegroupname:<String_value>,ip:<String_value>,servername:<String_value>,port:<Integer_va

Use this query-parameter to get gslbservicegroupmember resources based on additional properties.

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "gslbservicegroupmember": [ {

"servicegroupname":<String_value>,

"curclntconnections":<Double_value>,

"servicetype":<String_value>,
```

```
"establishedconn":<Double_value>,

"totalrequests":<Double_value>,

"responsebytesrate":<Double_value>,

"totalresponses":<Double_value>,

"requestbytesrate":<Double_value>,

"cursrvrconnections":<Double_value>,

"primaryipaddress":<String_value>,

"responsesrate":<Double_value>,

"curload":<Double_value>,

"totalrequestbytes":<Double_value>,

"state":<String_value>,

"totalresponsebytes":<Double_value>,

"primaryport":<Integer_value>,

"requestsrate":<Double_value>

}]}
```

High Availability

High-availability commands.

NAME	DESCRIPTION
hanode	Statistics for node resource.

hanode

Statistics for node resource.

Properties

(click to see Operations)

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
clearstats	<String>	Read-write	Clear the statsistics / counters. Possible values = basic, full
hacurstatus	<String>	Read-only	Whether a Citrix ADC is configured for high availability. Possible values are YES and NO. If the value is NO, the high availability statistics below are invalid.
hacurstate	<String>	Read-only	State of the HA node, based on its health, in a high availability setup. Possible values are: UP - Indicates that the node is accessible and can function as either a primary or secondary node. DISABLED - Indicates that the high availability status of the node has been manually disabled. Synchronization and propagation cannot take place between the peer nodes. INIT

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
			- Indicates that the node is in the process of becoming part of the high availability configuration. PARTIALFAIL - Indicates that one of the high availability monitored interfaces has failed because of a card or link failure. This state triggers a failover. COMPLETEFAIL - Indicates that all the interfaces of the node are unusable, because the interfaces on which high availability monitoring is enabled are not connected or are manually disabled. This state triggers a failover. DUMB - Indicates that the node is in listening mode. It does not participate in high availability transitions or transfer configuration from the peer node. This is a configured value, not a statistic. PARTIALFAILSSL - Indicates that the SSL card has failed. This state triggers a

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
			failover. ROUTEMONITORFAIL - Indicates that the route monitor has failed. This state triggers a failover.
hacurmasterstate	<String>	Read-only	Indicates the high availability state of the node. Possible values are: STAYSECONDARY - Indicates that the selected node remains the secondary node in a high availability setup. In this case a forced failover does not change the state but, instead, returns an appropriate error message. This is a configured value and not a statistic. PRIMARY - Indicates that the selected node is the primary node in a high availability setup. SECONDARY - Indicates that the selected node is the secondary node in a high availability setup. CLAIMING - Indicates that the secondary node is in the process of taking over as the primary node. This is the

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
			intermediate state in the transition of the secondary node to primary status. FORCE CHANGE - Indicates that the secondary node is forcibly changing its status to primary due to a forced failover issued on the secondary node.
transtime	<String>	Read-only	Time when the last master state transition occurred. You can use this statistic for debugging.
hatotpktrx	<Double>	Read-only	Number of heartbeat packets received from the peer node. Heartbeats are sent at regular intervals (default is 200 milliseconds) to determine the state of the peer node.
hapktrxrate	<Double>	Read-only	Rate (/s) counter for hatotpktrx
hatotpkttx	<Double>	Read-only	Number of heartbeat packets sent to the peer node. Heartbeats are sent at regular intervals (default is 200 milliseconds) to

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
			determine the state of the peer node.
hapkttxrate	<Double>	Read-only	Rate (/s) counter for hatotpkttx
haerrproptimeout	<Double>	Read-only	Number of times propagation timed out.
haerrsyncfailure	<Double>	Read-only	Number of times the configuration of the primary and secondary nodes failed to synchronize since that last transition. A synchronization failure results in mismatched configuration. It can be caused by a mismatch in the Remote Procedural Call (RPC) password on the two nodes forming the high availability pair.

Operations

(click to see Properties)

GET (ALL)

Some options that you can use for each operations:

- **Getting warnings in response:**NITRO allows you to get warnings in an operation by specifying the "warning" query parameter as "yes". For example, to get warnings while connecting to the Citrix ADC appliance, the URL is as follows:

  http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/login?warning=yes

If any, the warnings are displayed in the response payload with the HTTP code "209 X-NITRO-WARNING".

- **Authenticated access for individual NITRO operations:**NITRO allows you to logon to the Citrix ADC appliance to perform individual operations. You can use this option instead of creating a NITRO session (using the login object) and then using that session to perform all operations,

  To do this, you must specify the username and password in the request header of the NITRO request as follows:

  X-NITRO-USER:<username>

  X-NITRO-PASS:<password>

  Note: In such cases, make sure that the request header DOES not include the following:

  Cookie:NITRO_AUTH_TOKEN=<tokenvalue>

`*Note:`*

Mandatory parameters are marked in redand placeholder content is marked in <green>.

get (all)

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/stat/hanode

Query-parameters:

args

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/stat/hanode?
args=detail:<Boolean_value>,fullvalues:<Boolean_value>,ntimes:<Double_value>,logfile:<String_value

Use this query-parameter to get hanode resources based on additional properties.

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "hanode": [ {

"haerrsyncfailure":<Double_value>,

"transtime":<String_value>,

"hacurstatus":<String_value>,

"hacurmasterstate":<String_value>,

"hatotpkttx":<Double_value>,

"hapktrxrate":<Double_value>,

"haerrproptimeout":<Double_value>,

"hapkttxrate":<Double_value>,

"hacurstate":<String_value>,

"hatotpktrx":<Double_value>

}]}
```

HTTP DoS Protection

HTTP DOS protection statistics.

NAME	DESCRIPTION
dos	Statistics for dos.

NAME	DESCRIPTION
dospolicy	Statistics for DoS policy resource.

dos

Statistics for dos.

Properties

(click to see Operations)

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
clearstats	<String>	Read-write	Clear the statsistics / counters. Possible values = basic, full
dostotconditiontriggered	<Double>	Read-only	Number of times the Citrix ADC triggered the DOS JavaScript due to a condition match.
dosconditiontriggeredrate	<Double>	Read-only	Rate (/s) counter for dostotconditiontriggered
dostotvalidcookies	<Double>	Read-only	Number of clients from whom the Citrix ADC received a valid DOS cookie.
dosvalidcookiesrate	<Double>	Read-only	Rate (/s) counter for dostotvalidcookies
dostotdospriorityclients	<Double>	Read-only	Number of valid clients that were given DOS priority.

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
dosdospriorityclientsrate	<Double>	Read-only	Rate (/s) counter for dostotdospriorityclients

Operations

(click to see Properties)

GET (ALL)

Some options that you can use for each operations:

- **Getting warnings in response:**NITRO allows you to get warnings in an operation by specifying the "warning" query parameter as "yes". For example, to get warnings while connecting to the Citrix ADC appliance, the URL is as follows:

  http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/login?warning=yes

  If any, the warnings are displayed in the response payload with the HTTP code "209 X-NITRO-WARNING".

- **Authenticated access for individual NITRO operations:**NITRO allows you to logon to the Citrix ADC appliance to perform individual operations. You can use this option instead of creating a NITRO session (using the login object) and then using that session to perform all operations,

  To do this, you must specify the username and password in the request header of the NITRO request as follows:

  X-NITRO-USER:<username>

  X-NITRO-PASS:<password>

  Note: In such cases, make sure that the request header DOES not include the following:

  Cookie:NITRO_AUTH_TOKEN=<tokenvalue>

`*Note:`*

Mandatory parameters are marked in redand placeholder content is marked in <green>.

get (all)

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/stat/dos

Query-parameters:

args

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/stat/dos?
args=detail:<Boolean_value>,fullvalues:<Boolean_value>,ntimes:<Double_value>,logfile:<String_value

Use this query-parameter to get dos resources based on additional properties.

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "dos": [ {

"dostotvalidcookies":<Double_value>,

"dosvalidcookiesrate":<Double_value>,

"dosdospriorityclientsrate":<Double_value>,

"dosconditiontriggeredrate":<Double_value>,

"dostotconditiontriggered":<Double_value>,

"dostotdospriorityclients":<Double_value>
```

```
}]}
```

dospolicy

Statistics for DoS policy resource.

Properties

(click to see Operations)

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
name	<String>	Read-write	The name of the DoS protection policy whose statistics must be displayed. If a name is not provided, statistics of all the DoS protection policies are displayed. Minimum length = 1
clearstats	<String>	Read-write	Clear the statsistics / counters. Possible values = basic, full
doscurrentcltdetectrate	<Double>	Read-only	Current ratio of JavaScript send rate to the server response rate (Client detect rate)
dosphysicalserviceip	<String>	Read-only	IP address of the service to which this policy is bound.
dosphysicalserviceport	<Integer>	Read-only	Port address of the service to which this policy is bound.

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
doscurrentqueuesize	<Double>	Read-only	Current queue size of the server to which this policy is bound.
doscurrentqueuesize rate	<Double>	Read-only	Rate (/s) counter for doscurrentqueuesize
dostotjssent	<Double>	Read-only	Total number of DoS JavaScript transactions performed for this policy.
dosjssentrate	<Double>	Read-only	Rate (/s) counter for dostotjssent
dostotjsrefused	<Double>	Read-only	Number of times the DoS JavaScript was not sent because the set JavaScript rate was not met for this policy.
dosjsrefusedrate	<Double>	Read-only	Rate (/s) counter for dostotjsrefused
dostotvalidclients	<Double>	Read-only	Total number of valid DoS cookies received for this policy.
dosvalidclientsrate	<Double>	Read-only	Rate (/s) counter for dostotvalidclients
dostotjsbytessent	<Double>	Read-only	Total number of DoS JavaScript bytes sent for this policy.
dosjsbytessentrate	<Double>	Read-only	Rate (/s) counter for dostotjsbytessent

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
dostotnongetpostrequests	<Double>	Read-only	Number of non-GET and non-POST requests for which DOS JavaScript was sent.
dosnongetpostrequestsrate	<Double>	Read-only	Rate (/s) counter for dostotnongetpostrequests

Operations

(click to see Properties)

GET (ALL)| GET

Some options that you can use for each operations:

- **Getting warnings in response:**NITRO allows you to get warnings in an operation by specifying the "warning" query parameter as "yes". For example, to get warnings while connecting to the Citrix ADC appliance, the URL is as follows:

  http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/login?warning=yes

  If any, the warnings are displayed in the response payload with the HTTP code "209 X-NITRO-WARNING".
- **Authenticated access for individual NITRO operations:**NITRO allows you to logon to the Citrix ADC appliance to perform individual operations. You can use this option instead of creating a NITRO session (using the login object) and then using that session to perform all operations,

  To do this, you must specify the username and password in the request header of the NITRO request as follows:

  X-NITRO-USER:<username>

  X-NITRO-PASS:<password>

  Note: In such cases, make sure that the request header DOES not include the following:

  Cookie:NITRO_AUTH_TOKEN=<tokenvalue>

`*Note:` *

Mandatory parameters are marked in redand placeholder content is marked in <green>.

get (all)

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/stat/dospolicy

Query-parameters:

args

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/stat/dospolicy?
args=name:<String_value>,detail:<Boolean_value>,fullvalues:<Boolean_value>,ntimes:<Double_value>

Use this query-parameter to get dospolicy resources based on additional properties.

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "dospolicy": [ {

"name":<String_value>,

"dosjsrefusedrate":<Double_value>,

"dosphysicalserviceip":<String_value>,

"dosnongetpostrequestsrate":<Double_value>,
```

```
"dosjssentrate":<Double_value>,

"dosphysicalserviceport":<Integer_value>,

"dosjsbytessentrate":<Double_value>,

"doscurrentcltdetectrate":<Double_value>,

"dostotnongetpostrequests":<Double_value>,

"dostotvalidclients":<Double_value>,

"dosvalidclientsrate":<Double_value>,

"doscurrentqueuesizerate":<Double_value>,

"dostotjsbytessent":<Double_value>,

"dostotjssent":<Double_value>,

"dostotjsrefused":<Double_value>,

"doscurrentqueuesize":<Double_value>

}]}
```

get

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/stat/dospolicy/name_value><String>

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "dospolicy": [ {

"name":<String_value>,

"dosjsrefusedrate":<Double_value>,

"dosphysicalserviceip":<String_value>,

"dosnongetpostrequestsrate":<Double_value>,

"dosjssentrate":<Double_value>,

"dosphysicalserviceport":<Integer_value>,

"dosjsbytessentrate":<Double_value>,

"doscurrentcltdetectrate":<Double_value>,

"dostotnongetpostrequests":<Double_value>,

"dostotvalidclients":<Double_value>,

"dosvalidclientsrate":<Double_value>,

"doscurrentqueuesizerate":<Double_value>,

"dostotjsbytessent":<Double_value>,

"dostotjssent":<Double_value>,

"dostotjsrefused":<Double_value>,

"doscurrentqueuesize":<Double_value>

}]}
```

Ica

ICA statistics.

NAME	DESCRIPTION
icapolicy	Statistics for ICA policy resource.

icapolicy

Statistics for ICA policy resource.

Properties

(click to see Operations)

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
name	<String>	Read-write	Name of the ica policy. Minimum length = 1
clearstats	<String>	Read-write	Clear the statsistics / counters. Possible values = basic, full
pipolicyhits	<Double>	Read-only	Number of hits on the policy
pipolicyhitsrate	<Double>	Read-only	Rate (/s) counter for pipolicyhits
pipolicyundefhits	<Double>	Read-only	Number of undef hits on the policy
pipolicyundefhitsrate	<Double>	Read-only	Rate (/s) counter for pipolicyundefhits

Operations

(click to see Properties)

GET (ALL)| GET

Some options that you can use for each operations:

- **Getting warnings in response:**NITRO allows you to get warnings in an operation by specifying the "warning" query parameter as "yes". For example, to get warnings while connecting to the Citrix ADC appliance, the URL is as follows:

  http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/login?warning=yes

  If any, the warnings are displayed in the response payload with the HTTP code "209 X-NITRO-WARNING".

- **Authenticated access for individual NITRO operations:**NITRO allows you to logon to the Citrix ADC appliance to perform individual operations. You can use this option instead of creating a NITRO session (using the login object) and then using that session to perform all operations,

  To do this, you must specify the username and password in the request header of the NITRO request as follows:

  X-NITRO-USER:<username>

  X-NITRO-PASS:<password>

  Note: In such cases, make sure that the request header DOES not include the following:

  Cookie:NITRO_AUTH_TOKEN=<tokenvalue>

`*Note:`*

Mandatory parameters are marked in redand placeholder content is marked in <green>.

get (all)

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/stat/icapolicy

Query-parameters:

args

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/stat/icapolicy?
args=name:<String_value>,detail:<Boolean_value>,fullvalues:<Boolean_value>,ntimes:<Double_value>

Use this query-parameter to get icapolicy resources based on additional properties.

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "icapolicy": [ {

"name":<String_value>,

"pipolicyundefhitsrate":<Double_value>,

"pipolicyundefhits":<Double_value>,

"pipolicyhitsrate":<Double_value>,

"pipolicyhits":<Double_value>

}]}
```

get

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/stat/icapolicy/name_value><String>

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "icapolicy": [ {

"name":<String_value>,

"pipolicyundefhitsrate":<Double_value>,

"pipolicyundefhits":<Double_value>,

"pipolicyhitsrate":<Double_value>,

"pipolicyhits":<Double_value>

}]}
```

Integrated Caching

Integrated Caching statistics. The Integrated Caching feature is used to cache static and dynamic web application data.

NAME	DESCRIPTION
cache	Statistics for cache.
cachecontentgroup	Statistics for Integrated Cache content group resource.
cachepolicy	Statistics for Integrated Cache policy resource.
cachepolicylabel	Statistics for cache policy label resource.

cache

Statistics for cache.

Properties

(click to see Operations)

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
clearstats	<String>	Read-write	Clear the statsistics / counters. Possible values = basic, full
cachemaxmemorykb	<Double>	Read-only	Largest amount of memory the Citrix ADC can dedicate to caching, up to 50% of available memory. A 0 value disables caching, but the caching module continues to run.
cacherecentpercentsuccessfulrevalidation	<Double>	Read-only	Recently recorded percentage of times stored content was successfully revalidated by a 304 response rather than by a full response
cacherecentpercentstoreablemiss	<Double>	Read-only	Recently recorded ratio of store-able misses to all misses expressed as percentage.
cacherecentpercentparameterizedhits	<Double>	Read-only	Recently recorded ratio of parameterized 304 hits to all parameterized hits

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
			expressed as a percentage
cacherecentpercentoriginbandwidthsaved	<Double>	Read-only	Bytes served from cache divided by total bytes served to client. This ratio can be greater than 1 because of the assumption that all compression has been done in the Citrix ADC.
cacherecentpercenthit	<Double>	Read-only	Recently recorded cache hit ratio expressed as percentage
cacherecentpercentbytehit	<Double>	Read-only	Recently recorded cache byte hit ratio expressed as percentage. Here we define byte hit ratio as ((number of bytes served from the cache)/(total number of bytes served to the client)). This is the standard definition of Byte Hit Ratio. If compression is turned ON in NS then this ratio doesn't mean much. This might under or over estimate the origin-to-cache bandwidth saving (depending upon whether bytes

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
			served by CMP in Citrix ADC are more or less than compressed bytes served from the cache). If CMP is turned OFF in NS then this ratio is same as cacheRecentPercentOriginBandwidthSaved.
cacherecentpercent304hits	<Double>	Read-only	Recently recorded ratio of 304 hits to all hits expressed as percentage
cacheutilizedmemorykb	<Double>	Read-only	Amount of memory the integrated cache is currently using.
cachemaxmemoryactivekb	<Double>	Read-only	Currently active value of maximum memory
cache64maxmemorykb	<Double>	Read-only	Largest amount of memory the Citrix ADC can dedicate to caching, up to 50% of available memory. A 0 value disables caching, but the caching module continues to run.
cachepercentpethits	<Double>	Read-only	Percentage of cache hits in content groups that have Poll Every Time enabled, relative to all searches of content

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
			groups with Poll Every Time enabled.
cachetotpethits	<Double>	Read-only	Number of times a cache hit was found during a search of a content group that has Poll Every Time enabled.
cachepethitsrate	<Double>	Read-only	Rate (/s) counter for cachetotpethits
cachepercentparameterized304hits	<Double>	Read-only	Percentage of parameterized 304 hits relative to all parameterized hits.
cachetotparameterizedhits	<Double>	Read-only	Parameterized requests resulting in either a 304 or non-304 hit.
cacheparameterizedhitsrate	<Double>	Read-only	Rate (/s) counter for cachetotparameterizedhits
cachepercentsuccessfulrevalidation	<Double>	Read-only	Percentage of times stored content was successfully revalidated by a 304 (Object Not Modifed) response rather than by a full response
cachepercentstoreablemiss	<Double>	Read-only	Responses that were fetched from the origin, stored in the cache, and then served to the client,

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
			as a percentage of all cache misses.
cachetotfulltoconditionalrequest	<Double>	Read-only	Number of user-agent requests for a cached Poll Every Time (PET) response that were sent to the origin server as conditional requests.
cachefulltoconditionalrequestrate	<Double>	Read-only	Rate (/s) counter for cachetotfulltoconditionalrequest
cachetotsuccessfulrevalidation	<Double>	Read-only	Total number of times stored content was successfully revalidated by a 304 Not Modified response from the origin.
cachesuccessfulrevalidationrate	<Double>	Read-only	Rate (/s) counter for cachetotsuccessfulrevalidation
cachetotrevalidationmiss	<Double>	Read-only	Responses that an intervening cache revalidated with the integrated cache before serving, as determined by a Cache-Control: Max-Age header configurable in the integrated cache

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
cacherevalidationmissrate	<Double>	Read-only	Rate (/s) counter for cachetotrevalidation miss
cachetotnonstoreablemisses	<Double>	Read-only	Cache misses for which the fetched response is not stored in the cache. These responses match policies with a NOCACHE action or are affected by Poll Every Time.
cachenonstoreablemissesrate	<Double>	Read-only	Rate (/s) counter for cachetotnonstoreablemisses
cachetotstoreablemisses	<Double>	Read-only	Cache misses for which the fetched response is stored in the cache before serving it to the client. Storable misses conform to a built-in or user-defined caching policy that contains a CACHE action.
cachestoreablemissesrate	<Double>	Read-only	Rate (/s) counter for cachetotstoreablemisses
cachecompressedbytesserved	<Double>	Read-only	Number of compressed bytes served from the cache

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
cachecompressedbytesservedrate	<Double>	Read-only	Rate (/s) counter for cachecompressedbytesserved
cachepercentbytehit	<Double>	Read-only	Bytes served from the cache divided by total bytes served to the client. If compression is On in the Citrix ADC, this ratio may not reflect the bytes served by the compression module. If the compression is Off, this ratio is the same as cachePercentOriginBandwidthSaved.
cachebytesserved	<Double>	Read-only	Total number of bytes served from the integrated cache
cachebytesservedrate	<Double>	Read-only	Rate (/s) counter for cachebytesserved
cachetotresponsebytes	<Double>	Read-only	Total number of HTTP response bytes served by Citrix ADC from both the origin and the cache
cacheresponsebytesrate	<Double>	Read-only	Rate (/s) counter for cachetotresponsebytes
cachepercent304hits	<Double>	Read-only	304 responses as a percentage of all

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
			responses that the Citrix ADC served.
cachenummarker	<Double>	Read-only	Marker objects created when a response exceeds the maximum or minimum size for entries in its content group or has not yet received the minimum number of hits required for items in its content group.
cachepercentoriginbandwidthsaved	<Double>	Read-only	Percentage of origin bandwidth saved, expressed as number of bytes served from the integrated cache divided by all bytes served. The assumption is that all compression is done in the Citrix ADC.
cachepercenthit	<Double>	Read-only	Cache hits as percentage of the total number of requests
cachetotmisses	<Double>	Read-only	Intercepted HTTP requests requiring fetches from origin server.
cachemissesrate	<Double>	Read-only	Rate (/s) counter for cachetotmisses

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
cachetothits	<Double>	Read-only	Responses served from the integrated cache. These responses match a policy with a CACHE action.
cachehitsrate	<Double>	Read-only	Rate (/s) counter for cachetothits
cachetotrequests	<Double>	Read-only	Total cache hits plus total cache misses.
cacherequestsrate	<Double>	Read-only	Rate (/s) counter for cachetotrequests
cachenumcached	<Double>	Read-only	Responses currently in integrated cache. Includes responses fully downloaded, in the process of being downloaded, and expired or flushed but not yet removed.
cachecurhits	<Double>	Read-only	This number should be close to the number of hits being served currently.
cachecurmisses	<Double>	Read-only	Responses fetched from the origin and served from the cache. Should approximate storable misses. Does not include non-storable misses.
cachetotnon304hits	<Double>	Read-only	Total number of full (non-304) responses

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
			served from the cache. A 304 status code indicates that a response has not been modified since the last time it was served
cachenon304hitsrate	<Double>	Read-only	Rate (/s) counter for cachetotnon304hits
cachetot304hits	<Double>	Read-only	Object not modified responses served from the cache. (Status code 304 served instead of the full response.)
cache304hitsrate	<Double>	Read-only	Rate (/s) counter for cachetot304hits
cachetotsqlhits	<Double>	Read-only	sql response served from cache
cachesqlhitsrate	<Double>	Read-only	Rate (/s) counter for cachetotsqlhits
cachetotexpireatlastbyte	<Double>	Read-only	Instances of content expiring immediately after receiving the last body byte due to the Expire at Last Byte setting for the content group.
cacheexpireatlastbyterate	<Double>	Read-only	Rate (/s) counter for cachetotexpireatlastbyte
cachetotflashcachemisses	<Double>	Read-only	Number of requests to a content group with flash cache

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
			enabled that were cache misses. Flash cache distributes the response to all the clients in aqueue.
cacheflashcachemissesrate	<Double>	Read-only	Rate (/s) counter for cachetotflashcachemisses
cachetotflashcachehits	<Double>	Read-only	Number of requests to a content group with flash cache enabled that were cache hits. The flash cache setting queues requests that arrive simultaneously and distributes the response to all the clients in the queue.
cacheflashcachehitsrate	<Double>	Read-only	Rate (/s) counter for cachetotflashcachehits
cachetotparameterizedinvalidationrequests	<Double>	Read-only	Requests matching a policy with an invalidation (INVAL) action and a content group that uses an invalidation selector or parameters.
cacheparameterizedinvalidationrequestsrate	<Double>	Read-only	Rate (/s) counter for cachetotparameterizedinvalidationrequests

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
cachetotnonparameterizedinvalidationrequests	<Double>	Read-only	Requests that match an invalidation policy where the invalGroups parameter is configured and expires one or more content groups.
cachenonparameterizedinvalidationrequestsrate	<Double>	Read-only	Rate (/s) counter for cachetotnonparameterizedinvalidationrequests
cachetotinvalidationrequests	<Double>	Read-only	Requests that match an invalidation policy and result in expiration of specific cached responses or entire content groups.
cacheinvalidationrequestsrate	<Double>	Read-only	Rate (/s) counter for cachetotinvalidationrequests
cachetotparameterizedrequests	<Double>	Read-only	Total number of requests where the content group has hit and invalidation parameters or selectors.
cacheparameterizedrequestsrate	<Double>	Read-only	Rate (/s) counter for cachetotparameterizedrequests
cachetotparameterizednon304hits	<Double>	Read-only	Parameterized requests resulting in a full response (not status code 304:

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
			Object Not Updated) served from the cache.
cacheparameterized non304hitsrate	<Double>	Read-only	Rate (/s) counter for cachetotparameterized non304hits
cachetotparameterized304hits	<Double>	Read-only	Parameterized requests resulting in an object not modified (status code 304) response.
cacheparameterized 304hitsrate	<Double>	Read-only	Rate (/s) counter for cachetotparameterized304hits
cachetotpetrequests	<Double>	Read-only	Requests that triggered a search of a content group that has Poll Every Time (PET) enabled (always consult the origin server before serving cached data).
cachepetrequestsrate	<Double>	Read-only	Rate (/s) counter for cachetotpetrequests
cacheerrmemalloc	<Double>	Read-only	Total number of times the cache failed to allocate memory to store responses.
cachelargestresponsereceived	<Double>	Read-only	Size, in bytes, of largest response sent to client from

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
			the cache or the origin server.

Operations

(click to see Properties)

GET (ALL)

Some options that you can use for each operations:

- **Getting warnings in response:**NITRO allows you to get warnings in an operation by specifying the "warning" query parameter as "yes". For example, to get warnings while connecting to the Citrix ADC appliance, the URL is as follows:

  http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/login?warning=yes

  If any, the warnings are displayed in the response payload with the HTTP code "209 X-NITRO-WARNING".

- **Authenticated access for individual NITRO operations:**NITRO allows you to logon to the Citrix ADC appliance to perform individual operations. You can use this option instead of creating a NITRO session (using the login object) and then using that session to perform all operations,

  To do this, you must specify the username and password in the request header of the NITRO request as follows:

  X-NITRO-USER:<username>

  X-NITRO-PASS:<password>

  Note: In such cases, make sure that the request header DOES not include the following:

  Cookie:NITRO_AUTH_TOKEN=<tokenvalue>

`*Note:`*

Mandatory parameters are marked in redand placeholder content is marked in <green>.

get (all)

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/stat/cache

Query-parameters:

args

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/stat/cache?
args=detail:<Boolean_value>,fullvalues:<Boolean_value>,ntimes:<Double_value>,logfile:<String_value

Use this query-parameter to get cache resources based on additional properties.

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "cache": [ {

"cachehitsrate":<Double_value>,

"cachepethitsrate":<Double_value>,

"cachetotexpireatlastbyte":<Double_value>,

"cachenummarker":<Double_value>,

"cachecurhits":<Double_value>,

"cacheresponsebytesrate":<Double_value>,

"cacheutilizedmemorykb":<Double_value>,

"cachetotnonparameterizedinvalidationrequests":<Double_value>,
```

```
"cachepercentpethits":<Double_value>,

"cachebytesserved":<Double_value>,

"cachetotparameterizednon304hits":<Double_value>,

"cachebytesservedrate":<Double_value>,

"cachepercent304hits":<Double_value>,

"cachepercentbytehit":<Double_value>,

"cachepercentstoreablemiss":<Double_value>,

"cachetotpetrequests":<Double_value>,

"cachecompressedbytesserved":<Double_value>,

"cachetotresponsebytes":<Double_value>,

"cachelargestresponsereceived":<Double_value>,

"cachenonparameterizedinvalidationrequestsrate":<Double_value>,

"cacherecentpercentbytehit":<Double_value>,

"cacherecentpercentsuccessfulrevalidation":<Double_value>,

"cachesuccessfulrevalidationrate":<Double_value>,

"cachesqlhitsrate":<Double_value>,

"cacherequestsrate":<Double_value>,

"cachemissesrate":<Double_value>,

"cachemaxmemorykb":<Double_value>,

"cachenumcached":<Double_value>,

"cachetothits":<Double_value>,

"cachetotsuccessfulrevalidation":<Double_value>,
```

```
"cachepercentparameterized304hits":<Double_value>,

"cacheflashcachehitsrate":<Double_value>,

"cachestoreablemissesrate":<Double_value>,

"cachetotflashcachehits":<Double_value>,

"cachetotsqlhits":<Double_value>,

"cachetotrevalidationmiss":<Double_value>,

"cachemaxmemoryactivekb":<Double_value>,

"cachetotstoreablemisses":<Double_value>,

"cacheparameterizedhitsrate":<Double_value>,

"cacheparameterized304hitsrate":<Double_value>,

"cacheexpireatlastbyterate":<Double_value>,

"cachecurmisses":<Double_value>,

"cacherevalidationmissrate":<Double_value>,

"cachenon304hitsrate":<Double_value>,

"cachepercentsuccessfulrevalidation":<Double_value>,

"cachetotnon304hits":<Double_value>,

"cachefulltoconditionalrequestrate":<Double_value>,

"cacherecentpercentoriginbandwidthsaved":<Double_value>,

"cacherecentpercentparameterizedhits":<Double_value>,

"cachetotparameterizedinvalidationrequests":<Double_value>,

"cachetotflashcachemisses":<Double_value>,

"cacheparameterizedrequestsrate":<Double_value>,
```

```
"cacheerrmemalloc":<Double_value>,

"cacherecentpercenthit":<Double_value>,

"cachecompressedbytesservedrate":<Double_value>,

"cache304hitsrate":<Double_value>,

"cachepercenthit":<Double_value>,

"cachenonstoreablemissesrate":<Double_value>,

"cachetotinvalidationrequests":<Double_value>,

"cache64maxmemorykb":<Double_value>,

"cachetot304hits":<Double_value>,

"cachetotfulltoconditionalrequest":<Double_value>,

"cachetotparameterizedhits":<Double_value>,

"cachepetrequestsrate":<Double_value>,

"cacherecentpercent304hits":<Double_value>,

"cachetotnonstoreablemisses":<Double_value>,

"cachetotparameterizedrequests":<Double_value>,

"cachepercentoriginbandwidthsaved":<Double_value>,

"cachetotrequests":<Double_value>,

"cacheparameterizednon304hitsrate":<Double_value>,

"cachetotmisses":<Double_value>,

"cachetotpethits":<Double_value>,

"cacheparameterizedinvalidationrequestsrate":<Double_value>,

"cacherecentpercentstoreablemiss":<Double_value>,
```

```
"cacheflashcachemissesrate":<Double_value>,

"cacheinvalidationrequestsrate":<Double_value>,

"cachetotparameterized304hits":<Double_value>

}]}
```

cachecontentgroup

Statistics for Integrated Cache content group resource.

Properties

(click to see Operations)

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
name	<String>	Read-write	Name of the cache contentgroup for which to display statistics. If you do not set this parameter, statistics are shown for all cache contentgroups. Minimum length = 1
clearstats	<String>	Read-write	Clear the statsistics / counters. Possible values = basic, full
groupnon304hit	<Double>	Read-only	Non304 hits for ContentGroup
group304hit	<Double>	Read-only	304 hits for ContentGroup

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
totcell	<Double>	Read-only	Number of objects in contentgroup
totmarkercell	<Double>	Read-only	Number of marker objects in contentgroup
timesflushed	<Double>	Read-only	Number of times contentgroup is flushed
totmemory	<Double>	Read-only	current memory usage
maxmemory	<Double>	Read-only	maximum memory usage limit

Operations

(click to see Properties)

GET (ALL)| GET

Some options that you can use for each operations:

- **Getting warnings in response:**NITRO allows you to get warnings in an operation by specifying the "warning" query parameter as "yes". For example, to get warnings while connecting to the Citrix ADC appliance, the URL is as follows:

  http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/login?warning=yes

  If any, the warnings are displayed in the response payload with the HTTP code "209 X-NITRO-WARNING".

- **Authenticated access for individual NITRO operations:**NITRO allows you to logon to the Citrix ADC appliance to perform individual operations. You can use this option instead of creating a NITRO session (using the login object) and then using that session to perform all operations,

  To do this, you must specify the username and password in the request header of the NITRO request as follows:

X-NITRO-USER:<username>

X-NITRO-PASS:<password>

Note: In such cases, make sure that the request header DOES not include the following:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue>

`*Note:` *

Mandatory parameters are marked in redand placeholder content is marked in <green>.

get (all)

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/stat/cachecontentgroup

Query-parameters:

args

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/stat/cachecontentgroup?
args=name:<String_value>,detail:<Boolean_value>,fullvalues:<Boolean_value>,ntimes:<Double_value>

Use this query-parameter to get cachecontentgroup resources based on additional properties.

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "cachecontentgroup": [ {

"name":<String_value>,
```

```
"group304hit":<Double_value>,

"totmarkercell":<Double_value>,

"timesflushed":<Double_value>,

"totmemory":<Double_value>,

"totcell":<Double_value>,

"groupnon304hit":<Double_value>,

"maxmemory":<Double_value>

}]}
```

get

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/stat/cachecontentgroup/name_value><String>

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "cachecontentgroup": [ {

"name":<String_value>,
```

```
"group304hit":<Double_value>,

"totmarkercell":<Double_value>,

"timesflushed":<Double_value>,

"totmemory":<Double_value>,

"totcell":<Double_value>,

"groupnon304hit":<Double_value>,

"maxmemory":<Double_value>

}]}
```

cachepolicy

Statistics for Integrated Cache policy resource.

Properties

(click to see Operations)

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
policyname	<String>	Read-write	Name of the cache policy for which to display statistics. If you do not set this parameter, statistics are shown for all cache policies. Minimum length = 1
clearstats	<String>	Read-write	Clear the statsistics / counters. Possible values = basic, full

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
pipolicyhits	<Double>	Read-only	Number of hits on the policy
pipolicyhitsrate	<Double>	Read-only	Rate (/s) counter for pipolicyhits
pipolicyundefhits	<Double>	Read-only	Number of undef hits on the policy
pipolicyundefhitsrate	<Double>	Read-only	Rate (/s) counter for pipolicyundefhits

Operations

(click to see Properties)

GET (ALL)| GET

Some options that you can use for each operations:

- **Getting warnings in response:**NITRO allows you to get warnings in an operation by specifying the "warning" query parameter as "yes". For example, to get warnings while connecting to the Citrix ADC appliance, the URL is as follows:

  http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/login?warning=yes

  If any, the warnings are displayed in the response payload with the HTTP code "209 X-NITRO-WARNING".

- **Authenticated access for individual NITRO operations:**NITRO allows you to logon to the Citrix ADC appliance to perform individual operations. You can use this option instead of creating a NITRO session (using the login object) and then using that session to perform all operations,

  To do this, you must specify the username and password in the request header of the NITRO request as follows:

  X-NITRO-USER:<username>

  X-NITRO-PASS:<password>

  Note: In such cases, make sure that the request header DOES not include the following:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue>

`*Note:` *

Mandatory parameters are marked in redand placeholder content is marked in <green>.

get (all)

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/stat/cachepolicy

Query-parameters:

args

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/stat/cachepolicy?
args=policyname:<String_value>,detail:<Boolean_value>,fullvalues:<Boolean_value>,ntimes:<Double_

Use this query-parameter to get cachepolicy resources based on additional properties.

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "cachepolicy": [ {

"policyname":<String_value>,

"pipolicyundefhitsrate":<Double_value>,

"pipolicyundefhits":<Double_value>,

"pipolicyhitsrate":<Double_value>,
```

```
"pipolicyhits":<Double_value>

}]}
```

get

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/stat/cachepolicy/policyname_value><String>

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "cachepolicy": [ {

"policyname":<String_value>,

"pipolicyundefhitsrate":<Double_value>,

"pipolicyundefhits":<Double_value>,

"pipolicyhitsrate":<Double_value>,

"pipolicyhits":<Double_value>

}]}
```

cachepolicylabel

Statistics for cache policy label resource.

Properties

(click to see Operations)

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
labelname	<String>	Read-write	Name of the cache-policy label for which to display statistics. If you do not set this parameter statistics are shown for all cache-policy labels. Minimum length = 1
clearstats	<String>	Read-write	Clear the statsistics / counters. Possible values = basic, full
pipolicylabelhits	<Double>	Read-only	Number of times policy label was invoked.
pipolicylabelhitsrate	<Double>	Read-only	Rate (/s) counter for pipolicylabelhits

Operations

(click to see Properties)

GET (ALL)| GET

Some options that you can use for each operations:

- **Getting warnings in response:**NITRO allows you to get warnings in an operation by specifying the "warning" query parameter as "yes". For example, to get warnings while connecting to the Citrix ADC appliance, the URL is as follows:

  http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/login?warning=yes

If any, the warnings are displayed in the response payload with the HTTP code "209 X-NITRO-WARNING".

- **Authenticated access for individual NITRO operations:**NITRO allows you to logon to the Citrix ADC appliance to perform individual operations. You can use this option instead of creating a NITRO session (using the login object) and then using that session to perform all operations,

  To do this, you must specify the username and password in the request header of the NITRO request as follows:

  X-NITRO-USER:<username>

  X-NITRO-PASS:<password>

  Note: In such cases, make sure that the request header DOES not include the following:

  Cookie:NITRO_AUTH_TOKEN=<tokenvalue>

`*Note:`*

Mandatory parameters are marked in redand placeholder content is marked in <green>.

get (all)

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/stat/cachepolicylabel

Query-parameters:

args

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/stat/cachepolicylabel?
args=labelname:<String_value>,detail:<Boolean_value>,fullvalues:<Boolean_value>,ntimes:<Double_va

Use this query-parameter to get cachepolicylabel resources based on additional properties.

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "cachepolicylabel": [ {

"labelname":<String_value>,

"pipolicylabelhitsrate":<Double_value>,

"pipolicylabelhits":<Double_value>

}]}
```

get

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/stat/cachepolicylabel/labelname_value><String>

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "cachepolicylabel": [ {

"labelname":<String_value>,

"pipolicylabelhitsrate":<Double_value>,

"pipolicylabelhits":<Double_value>

}]}
```

Ipsec

IPSEC statistics.

NAME	DESCRIPTION
ipseccounters	Statistics for counters resource.

ipseccounters

Statistics for counters resource.

Properties

(click to see Operations)

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
clearstats	<String>	Read-write	Clear the statsistics / counters. Possible values = basic, full
ipsectotrxbytes	<Double>	Read-only	Bytes received during IPsec sessions.

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
ipsecrxbytesrate	<Double>	Read-only	Rate (/s) counter for ipsectotrxbytes
ipsectottxbytes	<Double>	Read-only	Bytes sent during IPsec sessions.
ipsectxbytesrate	<Double>	Read-only	Rate (/s) counter for ipsectottxbytes
ipsectotrxpkts	<Double>	Read-only	Packets received during IPsec sessions.
ipsecrxpktsrate	<Double>	Read-only	Rate (/s) counter for ipsectotrxpkts
ipsectottxpkts	<Double>	Read-only	Packets sent during IPsec sessions.
ipsectxpktsrate	<Double>	Read-only	Rate (/s) counter for ipsectottxpkts

Operations

(click to see Properties)

GET (ALL)

Some options that you can use for each operations:

- **Getting warnings in response:**NITRO allows you to get warnings in an operation by specifying the "warning" query parameter as "yes". For example, to get warnings while connecting to the Citrix ADC appliance, the URL is as follows:

  http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/login?warning=yes

  If any, the warnings are displayed in the response payload with the HTTP code "209 X-NITRO-WARNING".
- **Authenticated access for individual NITRO operations:**NITRO allows you to logon to the Citrix ADC appliance to perform individual operations. You can use this option instead of

creating a NITRO session (using the login object) and then using that session to perform all operations,

To do this, you must specify the username and password in the request header of the NITRO request as follows:

X-NITRO-USER:<username>

X-NITRO-PASS:<password>

Note: In such cases, make sure that the request header DOES not include the following:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue>

`*Note:`*

Mandatory parameters are marked in redand placeholder content is marked in <green>.

get (all)

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/stat/ipseccounters

Query-parameters:

args

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/stat/ipseccounters?
args=detail:<Boolean_value>,fullvalues:<Boolean_value>,ntimes:<Double_value>,logfile:<String_value

Use this query-parameter to get ipseccounters resources based on additional properties.

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "ipseccounters": [ {
"ipsecrxbytesrate":<Double_value>,
"ipsectxbytesrate":<Double_value>,
"ipsectottxbytes":<Double_value>,
"ipsecrxpktsrate":<Double_value>,
"ipsectotrxpkts":<Double_value>,
"ipsectottxpkts":<Double_value>,
"ipsectotrxbytes":<Double_value>,
"ipsectxpktsrate":<Double_value>
}]}
```

Ipsecalg

IPSECALG statistics.

NAME	DESCRIPTION
ipsecalgcounters	Statistics for counters resource.

ipsecalgcounters

Statistics for counters resource.

Properties

(click to see Operations)

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
name	<String>	Read-write	Name of IPSec ALG Profile. Minimum length = 1 Maximum length = 32
clearstats	<String>	Read-write	Clear the statsistics / counters. Possible values = basic, full
ipsecalgtotsessions	<Double>	Read-only	Total session for IPSec ALG.
ipsecalgsessionsrate	<Double>	Read-only	Rate (/s) counter for ipsecalgtotsessions
ipsecalgtotdrsessions	<Double>	Read-only	Total Dropped IPsec ALG sessions.
ipsecalgdrsessionsrate	<Double>	Read-only	Rate (/s) counter for ipsecalgtotdrsessions
ipsecalgcuractsessions	<Double>	Read-only	Currently active IPsec ALG sessions.
ipsecalgcuractsessionsrate	<Double>	Read-only	Rate (/s) counter for ipsecalgcuractsessions
ipsecalgcurblksessions	<Double>	Read-only	Currently blocked sessions on ESP Gate.
ipsecalgcurblksessionsrate	<Double>	Read-only	Rate (/s) counter for ipsecalgcurblksessions

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
ipsecalgtotrxpkts	<Double>	Read-only	Total Packets received during IPsec ALG sessions.
ipsecalgrxpktsrate	<Double>	Read-only	Rate (/s) counter for ipsecalgtotrxpkts
ipsecalgtottxpkts	<Double>	Read-only	Total Packets sent during IPsec ALG sessions.
ipsecalgtxpktsrate	<Double>	Read-only	Rate (/s) counter for ipsecalgtottxpkts

Operations

(click to see Properties)

GET (ALL)| GET

Some options that you can use for each operations:

- **Getting warnings in response:**NITRO allows you to get warnings in an operation by specifying the "warning" query parameter as "yes". For example, to get warnings while connecting to the Citrix ADC appliance, the URL is as follows:

  http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/login?warning=yes

  If any, the warnings are displayed in the response payload with the HTTP code "209 X-NITRO-WARNING".

- **Authenticated access for individual NITRO operations:**NITRO allows you to logon to the Citrix ADC appliance to perform individual operations. You can use this option instead of creating a NITRO session (using the login object) and then using that session to perform all operations,

  To do this, you must specify the username and password in the request header of the NITRO request as follows:

  X-NITRO-USER:<username>

  X-NITRO-PASS:<password>

Note: In such cases, make sure that the request header DOES not include the following:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue>

`*Note:` *

Mandatory parameters are marked in redand placeholder content is marked in <green>.

get (all)

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/stat/ipsecalgcounters

Query-parameters:

args

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/stat/ipsecalgcounters?
args=name:<String_value>,detail:<Boolean_value>,fullvalues:<Boolean_value>,ntimes:<Double_value>

Use this query-parameter to get ipsecalgcounters resources based on additional properties.

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "ipsecalgcounters": [ {

"name":<String_value>,

"ipsecalgdrsessionsrate":<Double_value>,

"ipsecalgtotrxpkts":<Double_value>,
```

```
"ipsecalgtottxpkts":<Double_value>,

"ipsecalgtxpktsrate":<Double_value>,

"ipsecalgtotsessions":<Double_value>,

"ipsecalgrxpktsrate":<Double_value>,

"ipsecalgtotdrsessions":<Double_value>,

"ipsecalgcuractsessionsrate":<Double_value>,

"ipsecalgsessionsrate":<Double_value>,

"ipsecalgcuractsessions":<Double_value>,

"ipsecalgcurblksessions":<Double_value>,

"ipsecalgcurblksessionsrate":<Double_value>

}]}
```

get

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/stat/ipsecalgcounters/name_value><String>

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "ipsecalgcounters": [ {
"name":<String_value>,
"ipsecalgdrsessionsrate":<Double_value>,
"ipsecalgtotrxpkts":<Double_value>,
"ipsecalgtottxpkts":<Double_value>,
"ipsecalgtxpktsrate":<Double_value>,
"ipsecalgtotsessions":<Double_value>,
"ipsecalgrxpktsrate":<Double_value>,
"ipsecalgtotdrsessions":<Double_value>,
"ipsecalgcuractsessionsrate":<Double_value>,
"ipsecalgsessionsrate":<Double_value>,
"ipsecalgcuractsessions":<Double_value>,
"ipsecalgcurblksessions":<Double_value>,
"ipsecalgcurblksessionsrate":<Double_value>
}]}
```

lldp

Statistics for lldp.

Properties

(click to see Operations)

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
ifnum	<String>	Read-write	LLDP Statistics per interfaces.
clearstats	<String>	Read-write	Clear the statsistics / counters. Possible values = basic, full
rxportframestotal	<Double>	Read-only	Total LLDP Packets received.
rxportframesrate	<Double>	Read-only	Rate (/s) counter for rxportframestotal
rxportbytestotal	<Double>	Read-only	Total LLDP bytes received
rxportbytesrate	<Double>	Read-only	Rate (/s) counter for rxportbytestotal
txportframestotal	<Double>	Read-only	Total LLDP Packets transmitted
txportframesrate	<Double>	Read-only	Rate (/s) counter for txportframestotal
txportbytestotal	<Double>	Read-only	Total LLDP bytes transmitted.
txportbytesrate	<Double>	Read-only	Rate (/s) counter for txportbytestotal
rxportframeserrors	<Double>	Read-only	Total errors in LLDP packets.
rxportframeserrorsrate	<Double>	Read-only	Rate (/s) counter for rxportframeserrors
rxporttlvsdiscardedtotal	<Double>	Read-only	Total discarded LLDP packets.

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
rxporttlvsdiscardedrate	<Double>	Read-only	Rate (/s) counter for rxporttlvsdiscardedtotal
rxporttlvsunrecognizedtotal	<Double>	Read-only	Total TLVs not Recognised.
rxporttlvsunrecognizedrate	<Double>	Read-only	Rate (/s) counter for rxporttlvsunrecognizedtotal

Operations

(click to see Properties)

GET (ALL)| GET

Some options that you can use for each operations:

- **Getting warnings in response:**NITRO allows you to get warnings in an operation by specifying the "warning" query parameter as "yes". For example, to get warnings while connecting to the Citrix ADC appliance, the URL is as follows:

  http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/login?warning=yes

  If any, the warnings are displayed in the response payload with the HTTP code "209 X-NITRO-WARNING".

- **Authenticated access for individual NITRO operations:**NITRO allows you to logon to the Citrix ADC appliance to perform individual operations. You can use this option instead of creating a NITRO session (using the login object) and then using that session to perform all operations,

  To do this, you must specify the username and password in the request header of the NITRO request as follows:

  X-NITRO-USER:<username>

  X-NITRO-PASS:<password>

  Note: In such cases, make sure that the request header DOES not include the following:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue>

`*Note:`*

Mandatory parameters are marked in redand placeholder content is marked in <green>.

get (all)

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/stat/lldp

Query-parameters:

args

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/stat/lldp?
args=ifnum:<String_value>,detail:<Boolean_value>,fullvalues:<Boolean_value>,ntimes:<Double_value>

Use this query-parameter to get lldp resources based on additional properties.

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "lldp": [ {

"ifnum":<String_value>,

"rxporttlvsdiscardedtotal":<Double_value>,

"rxportframeserrors":<Double_value>,
```

```
"rxportframeserrorsrate":<Double_value>,

"rxportframestotal":<Double_value>,

"txportbytestotal":<Double_value>,

"txportframestotal":<Double_value>,

"rxporttlvsunrecognizedtotal":<Double_value>,

"rxportbytesrate":<Double_value>,

"rxportbytestotal":<Double_value>,

"rxporttlvsdiscardedrate":<Double_value>,

"txportbytesrate":<Double_value>,

"rxporttlvsunrecognizedrate":<Double_value>,

"rxportframesrate":<Double_value>,

"txportframesrate":<Double_value>

}]}
```

get

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/stat/lldp/ifnum_value><String>

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "lldp": [ {

"ifnum":<String_value>,

"rxporttlvsdiscardedtotal":<Double_value>,

"rxportframeserrors":<Double_value>,

"rxportframeserrorsrate":<Double_value>,

"rxportframestotal":<Double_value>,

"txportbytestotal":<Double_value>,

"txportframestotal":<Double_value>,

"rxporttlvsunrecognizedtotal":<Double_value>,

"rxportbytesrate":<Double_value>,

"rxportbytestotal":<Double_value>,

"rxporttlvsdiscardedrate":<Double_value>,

"txportbytesrate":<Double_value>,

"rxporttlvsunrecognizedrate":<Double_value>,

"rxportframesrate":<Double_value>,

"txportframesrate":<Double_value>

}]}
```

Load Balancing

Load Balancing statistics. The load balancing methods manage the selection of an appropriate physical server in a server farm.

NAME	DESCRIPTION
lbvserver	Statistics for Load Balancing Virtual Server resource.

lbvserver

Statistics for Load Balancing Virtual Server resource.

Properties

(click to see Operations)

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
name	<String>	Read-write	Name of the virtual server. If no name is provided, statistical data of all configured virtual servers is displayed. Minimum length = 1
clearstats	<String>	Read-write	Clear the statsistics / counters. Possible values = basic, full
sortby	<String>	Read-write	use this argument to sort by specific key. Possible values =
sortorder	<String>	Read-write	use this argument to specify sort order.

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
			Default value: SORT_DESCENDING Possible values = ascending, descending
avgcltttlb	<Double>	Read-only	Average TTLB between the client and the server. TTLB is the time interval between sending the request packet to a service and receiving the ACK for response from client.
cltresponsetimeapdex	<Double>	Read-only	Vserver APDEX index based on client response times.
vsvrsurgecount	<Double>	Read-only	Number of requests waiting on this vserver.
establishedconn	<Double>	Read-only	Number of client connections in ESTABLISHED state.
inactsvcs	<Double>	Read-only	number of INACTIVE services bound to a vserver
vslbhealth	<Double>	Read-only	Health of the vserver. This gives percentage of UP services bound to this vserver.

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
primaryipaddress	<String>	Read-only	IP address of the vserver
primaryport	<Integer>	Read-only	The port on which the service is running.
type	<String>	Read-only	Protocol associated with the vserver
state	<String>	Read-only	Current state of the server. There are seven possible values: UP(7), DOWN(1), UNKNOWN(2), BUSY(3), OFS(Out of Service)(4), TROFS(Transition Out of Service)(5), TROFS_DOWN(Down When going Out of Service)(8)
actsvcs	<Double>	Read-only	number of ACTIVE services bound to a vserver
tothits	<Double>	Read-only	Total vserver hits
hitsrate	<Double>	Read-only	Rate (/s) counter for tothits
totalrequests	<Double>	Read-only	Total number of requests received on this service or virtual server. (This applies to HTTP/SSL services and servers.)

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
requestsrate	<Double>	Read-only	Rate (/s) counter for totalrequests
totalresponses	<Double>	Read-only	Number of responses received on this service or virtual server. (This applies to HTTP/SSL services and servers.)
responsesrate	<Double>	Read-only	Rate (/s) counter for totalresponses
totalrequestbytes	<Double>	Read-only	Total number of request bytes received on this service or virtual server.
requestbytesrate	<Double>	Read-only	Rate (/s) counter for totalrequestbytes
totalresponsebytes	<Double>	Read-only	Number of response bytes received by this service or virtual server.
responsebytesrate	<Double>	Read-only	Rate (/s) counter for totalresponsebytes
totalpktsrecvd	<Double>	Read-only	Total number of packets received by this service or virtual server.
pktsrecvdrate	<Double>	Read-only	Rate (/s) counter for totalpktsrecvd
totalpktssent	<Double>	Read-only	Total number of packets sent.

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
pktssentrate	<Double>	Read-only	Rate (/s) counter for totalpktssent
curclntconnections	<Double>	Read-only	Number of current client connections.
cursrvrconnections	<Double>	Read-only	Number of current connections to the actual servers behind the virtual server.
curpersistencesessions	<Double>	Read-only	current vserver owned persistence sessions
curbackuppersistencesessions	<Double>	Read-only	current vserver owned backup persistence sessions
surgecount	<Double>	Read-only	Number of requests in the surge queue.
svcsurgecount	<Double>	Read-only	Total number of requests in the surge queues of all the services bound to this LB-vserver.
sothreshold	<Double>	Read-only	Spill Over Threshold set on the VServer.
totspillovers	<Double>	Read-only	Number of times vserver experienced spill over.
labelledconn	<Double>	Read-only	Number of Labeled connection on this vserver

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
pushlabel	<Double>	Read-only	Number of labels for this push vserver.
deferredreq	<Double>	Read-only	Number of deferred request on this vserver
deferredreqrate	<Double>	Read-only	Rate (/s) counter for deferredreq
invalidrequestrespon se	<Double>	Read-only	Number invalid requests/responses on this vserver
invalidrequestrespon sedropped	<Double>	Read-only	Number invalid requests/responses dropped on this vserver
totvserverdownbacku phits	<Double>	Read-only	Number of times traffic was diverted to backup vserver since primary vserver was DOWN.
curmptcpsessions	<Double>	Read-only	Current Multipath TCP sessions
cursubflowconn	<Double>	Read-only	Current Multipath TCP subflows
totalconnreassembly queue75	<Double>	Read-only	Total no of connections with 75% TCP reassembly queue
totalconnreassembly queueflush	<Double>	Read-only	Total no of connections incurred TCP reassembly queue flush

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
totalsvrbusyerr	<Double>	Read-only	Total no of server busy error
svrbusyerrrate	<Double>	Read-only	Rate (/s) counter for totalsvrbusyerr
totclttlbtransactions	<Double>	Read-only	Total transactions where client TTLB is calculated.
toleratingttlbtransactions	<Double>	Read-only	Tolerable transactions based on APDEX threshold (>T ;; <4T).
frustratingttlbtransactions	<Double>	Read-only	Frustrating transactions based on APDEX threshold (>4T).

Operations

(click to see Properties)

GET (ALL)| GET

Some options that you can use for each operations:

- **Getting warnings in response:**NITRO allows you to get warnings in an operation by specifying the "warning" query parameter as "yes". For example, to get warnings while connecting to the Citrix ADC appliance, the URL is as follows:

  http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/login?warning=yes

  If any, the warnings are displayed in the response payload with the HTTP code "209 X-NITRO-WARNING".

- **Authenticated access for individual NITRO operations:**NITRO allows you to logon to the Citrix ADC appliance to perform individual operations. You can use this option instead of creating a NITRO session (using the login object) and then using that session to perform all operations,

To do this, you must specify the username and password in the request header of the NITRO request as follows:

X-NITRO-USER:<username>

X-NITRO-PASS:<password>

Note: In such cases, make sure that the request header DOES not include the following:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue>

`*Note:`*

Mandatory parameters are marked in redand placeholder content is marked in <green>.

get (all)

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/stat/lbvserver

Query-parameters:

args

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/stat/lbvserver?
args=name:<String_value>,detail:<Boolean_value>,fullvalues:<Boolean_value>,ntimes:<Double_value>

Use this query-parameter to get lbvserver resources based on additional properties.

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "lbvserver": [ {
"name":<String_value>,
"curclntconnections":<Double_value>,
"establishedconn":<Double_value>,
"totalpktssent":<Double_value>,
"labelledconn":<Double_value>,
"tothits":<Double_value>,
"totalrequests":<Double_value>,
"sothreshold":<Double_value>,
"cursubflowconn":<Double_value>,
"svrbusyerrrate":<Double_value>,
"surgecount":<Double_value>,
"responsebytesrate":<Double_value>,
"invalidrequestresponsedropped":<Double_value>,
"totalresponses":<Double_value>,
"curpersistencesessions":<Double_value>,
"requestbytesrate":<Double_value>,
"totalsvrbusyerr":<Double_value>,
"avgclttlb":<Double_value>,
"type":<String_value>,
"hitsrate":<Double_value>,
```

```
"totalconnreassemblyqueue75":<Double_value>,

"cursrvrconnections":<Double_value>,

"totalconnreassemblyqueueflush":<Double_value>,

"cltresponsetimeapdex":<Double_value>,

"totclttlbtransactions":<Double_value>,

"pktsrecvdrate":<Double_value>,

"primaryipaddress":<String_value>,

"vsvrsurgecount":<Double_value>,

"pushlabel":<Double_value>,

"toleratingttlbtransactions":<Double_value>,

"responsesrate":<Double_value>,

"deferredreq":<Double_value>,

"curmptcpsessions":<Double_value>,

"totspillovers":<Double_value>,

"svcsurgecount":<Double_value>,

"totalrequestbytes":<Double_value>,

"invalidrequestresponse":<Double_value>,

"state":<String_value>,

"vslbhealth":<Double_value>,

"deferredreqrate":<Double_value>,

"actsvcs":<Double_value>,

"totalpktsrecvd":<Double_value>,
```

```
"curbackuppersistencesessions":<Double_value>,

"pktssentrate":<Double_value>,

"frustratingttlbtransactions":<Double_value>,

"totalresponsebytes":<Double_value>,

"primaryport":<Integer_value>,

"requestsrate":<Double_value>,

"totvserverdownbackuphits":<Double_value>,

"inactsvcs":<Double_value>

}]}
```

get

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/stat/lbvserver/name_value><String>

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "lbvserver": [ {
```

```
"name":<String_value>,

"curclntconnections":<Double_value>,

"establishedconn":<Double_value>,

"totalpktssent":<Double_value>,

"labelledconn":<Double_value>,

"tothits":<Double_value>,

"totalrequests":<Double_value>,

"sothreshold":<Double_value>,

"cursubflowconn":<Double_value>,

"svrbusyerrrate":<Double_value>,

"surgecount":<Double_value>,

"responsebytesrate":<Double_value>,

"invalidrequestresponsedropped":<Double_value>,

"totalresponses":<Double_value>,

"curpersistencesessions":<Double_value>,

"requestbytesrate":<Double_value>,

"totalsvrbusyerr":<Double_value>,

"avgclttlb":<Double_value>,

"type":<String_value>,

"hitsrate":<Double_value>,

"totalconnreassemblyqueue75":<Double_value>,

"cursrvrconnections":<Double_value>,
```

```
"totalconnreassemblyqueueflush":<Double_value>,

"cltresponsetimeapdex":<Double_value>,

"totclttlbtransactions":<Double_value>,

"pktsrecvdrate":<Double_value>,

"primaryipaddress":<String_value>,

"vsvrsurgecount":<Double_value>,

"pushlabel":<Double_value>,

"toleratingttlbtransactions":<Double_value>,

"responsesrate":<Double_value>,

"deferredreq":<Double_value>,

"curmptcpsessions":<Double_value>,

"totspillovers":<Double_value>,

"svcsurgecount":<Double_value>,

"totalrequestbytes":<Double_value>,

"invalidrequestresponse":<Double_value>,

"state":<String_value>,

"vslbhealth":<Double_value>,

"deferredreqrate":<Double_value>,

"actsvcs":<Double_value>,

"totalpktsrecvd":<Double_value>,

"curbackuppersistencesessions":<Double_value>,

"pktssentrate":<Double_value>,
```

```
"frustratingttlbtransactions":<Double_value>,

"totalresponsebytes":<Double_value>,

"primaryport":<Integer_value>,

"requestsrate":<Double_value>,

"totvserverdownbackuphits":<Double_value>,

"inactsvcs":<Double_value>

}]}
```

lsn

Statistics for lsn.

Properties

(click to see Operations)

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
clearstats	<String>	Read-write	Clear the statsistics / counters. Possible values = basic, full
lsntottcprxpkts	<Double>	Read-only	Number of LSN TCP Received packets.
lsntcprxpktsrate	<Double>	Read-only	Rate (/s) counter for lsntottcprxpkts
lsntottcprxbytes	<Double>	Read-only	Number of LSN TCP Received bytes.
lsntcprxbytesrate	<Double>	Read-only	Rate (/s) counter for lsntottcprxbytes

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
lsntottcptxpkts	<Double>	Read-only	Number of LSN TCP Transmitted packets.
lsntcptxpktsrate	<Double>	Read-only	Rate (/s) counter for lsntottcptxpkts
lsntottcptxbytes	<Double>	Read-only	Number of LSN TCP Transmitted bytes.
lsntcptxbytesrate	<Double>	Read-only	Rate (/s) counter for lsntottcptxbytes
lsntottcpdrppkts	<Double>	Read-only	Number of LSN TCP Dropped packets.
lsntcpdrppktsrate	<Double>	Read-only	Rate (/s) counter for lsntottcpdrppkts
lsncurtcpsessions	<Double>	Read-only	Number of LSN TCP Current Sessions.
lsncurtcpsessionsrate	<Double>	Read-only	Rate (/s) counter for lsncurtcpsessions
lsntotudprxpkts	<Double>	Read-only	Number of LSN UDP Received packets.
lsnudprxpktsrate	<Double>	Read-only	Rate (/s) counter for lsntotudprxpkts
lsntotudprxbytes	<Double>	Read-only	Number of LSN UDP Received bytes.
lsnudprxbytesrate	<Double>	Read-only	Rate (/s) counter for lsntotudprxbytes
lsntotudptxpkts	<Double>	Read-only	Number of LSN UDP Transmitted packets.
lsnudptxpktsrate	<Double>	Read-only	Rate (/s) counter for lsntotudptxpkts

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
lsntotudptxbytes	<Double>	Read-only	Number of LSN UDP Transmitted bytes.
lsnudptxbytesrate	<Double>	Read-only	Rate (/s) counter for lsntotudptxbytes
lsntotudpdrppkts	<Double>	Read-only	Number of LSN UDP Dropped packets.
lsnudpdrppktsrate	<Double>	Read-only	Rate (/s) counter for lsntotudpdrppkts
lsncurudpsessions	<Double>	Read-only	Number of LSN UDP Current Sessions.
lsncurudpsessionsrate	<Double>	Read-only	Rate (/s) counter for lsncurudpsessions
lsntoticmprxpkts	<Double>	Read-only	Number of LSN ICMP Received packets.
lsnicmprxpktsrate	<Double>	Read-only	Rate (/s) counter for lsntoticmprxpkts
lsntoticmprxbytes	<Double>	Read-only	Number of LSN ICMP Received bytes.
lsnicmprxbytesrate	<Double>	Read-only	Rate (/s) counter for lsntoticmprxbytes
lsntoticmptxpkts	<Double>	Read-only	Number of LSN ICMP Transmitted packets.
lsnicmptxpktsrate	<Double>	Read-only	Rate (/s) counter for lsntoticmptxpkts

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
lsntoticmptxbytes	<Double>	Read-only	Number of LSN ICMP Transmitted bytes.
lsnicmptxbytesrate	<Double>	Read-only	Rate (/s) counter for lsntoticmptxbytes
lsntoticmpdrppkts	<Double>	Read-only	Number of LSN ICMP Dropped packets.
lsnicmpdrppktsrate	<Double>	Read-only	Rate (/s) counter for lsntoticmpdrppkts
lsncuricmpsessions	<Double>	Read-only	Number of LSN ICMP Current Sessions.
lsncuricmpsessionsrate	<Double>	Read-only	Rate (/s) counter for lsncuricmpsessions
lsncursessions	<Double>	Read-only	Current number of LSN sessions.
lsncursessionsrate	<Double>	Read-only	Rate (/s) counter for lsncursessions
lsncursubscribers	<Double>	Read-only	Current number of LSN subscribers.
lsncursubscribersrate	<Double>	Read-only	Rate (/s) counter for lsncursubscribers

Operations

(click to see Properties)

GET (ALL)

Some options that you can use for each operations:

- **Getting warnings in response:**NITRO allows you to get warnings in an operation by specifying the "warning" query parameter as "yes". For example, to get warnings while connecting to the Citrix ADC appliance, the URL is as follows:

  http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/login?warning=yes

  If any, the warnings are displayed in the response payload with the HTTP code "209 X-NITRO-WARNING".

- **Authenticated access for individual NITRO operations:**NITRO allows you to logon to the Citrix ADC appliance to perform individual operations. You can use this option instead of creating a NITRO session (using the login object) and then using that session to perform all operations,

  To do this, you must specify the username and password in the request header of the NITRO request as follows:

  X-NITRO-USER:<username>

  X-NITRO-PASS:<password>

  Note: In such cases, make sure that the request header DOES not include the following:

  Cookie:NITRO_AUTH_TOKEN=<tokenvalue>

`*Note:`*

Mandatory parameters are marked in redand placeholder content is marked in <green>.

get (all)

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/stat/lsn

Query-parameters:

args

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/stat/lsn?
args=detail:<Boolean_value>,fullvalues:<Boolean_value>,ntimes:<Double_value>,logfile:<String_value

Use this query-parameter to get lsn resources based on additional properties.

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "lsn": [ {

"lsncursubscribersrate":<Double_value>,

"lsntotudptxbytes":<Double_value>,

"lsncuricmpsessionsrate":<Double_value>,

"lsncursessions":<Double_value>,

"lsntottcptxbytes":<Double_value>,

"lsnudptxbytesrate":<Double_value>,

"lsntotudpdrppkts":<Double_value>,

"lsntottcpdrppkts":<Double_value>,

"lsntoticmptxpkts":<Double_value>,

"lsntcptxbytesrate":<Double_value>,

"lsnudprxbytesrate":<Double_value>,

"lsntcpdrppktsrate":<Double_value>,

"lsntotudprxbytes":<Double_value>,

"lsnicmpdrppktsrate":<Double_value>,
```

```
"lsntottcprxbytes":<Double_value>,

"lsntoticmptxbytes":<Double_value>,

"lsntoticmprxpkts":<Double_value>,

"lsnicmptxpktsrate":<Double_value>,

"lsnicmprxbytesrate":<Double_value>,

"lsntoticmpdrppkts":<Double_value>,

"lsntcprxpktsrate":<Double_value>,

"lsntottcptxpkts":<Double_value>,

"lsncurtcpsessionsrate":<Double_value>,

"lsncursubscribers":<Double_value>,

"lsncurudpsessionsrate":<Double_value>,

"lsnudptxpktsrate":<Double_value>,

"lsntottcprxpkts":<Double_value>,

"lsnicmptxbytesrate":<Double_value>,

"lsntotudprxpkts":<Double_value>,

"lsncursessionsrate":<Double_value>,

"lsntcptxpktsrate":<Double_value>,

"lsnudprxpktsrate":<Double_value>,

"lsntotudptxpkts":<Double_value>,

"lsncurudpsessions":<Double_value>,

"lsnudpdrppktsrate":<Double_value>,

"lsntoticmprxbytes":<Double_value>,
```

```
"lsncuricmpsessions":<Double_value>,

"lsncurtcpsessions":<Double_value>,

"lsntcprxbytesrate":<Double_value>,

"lsnicmprxpktsrate":<Double_value>

}]}
```

lsndslite

Read/write properties

Properties

(click to see Operations)

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
clearstats	<String>	Read-write	Clear the statsistics / counters. Possible values = basic, full
lsntotdsliterxpkts	<Double>	Read-only	Total number of LSN DS-Lite rx pkts.
lsndsliterxpktsrate	<Double>	Read-only	Rate (/s) counter for lsntotdsliterxpkts
lsntotdsliterxbytes	<Double>	Read-only	Total number of LSN DS-Lite rx bytes.
lsndsliterxbytesrate	<Double>	Read-only	Rate (/s) counter for lsntotdsliterxbytes
lsntotdslitetxpkts	<Double>	Read-only	Total number of LSN DS-Lite tx pkts.

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
lsndslitetxpktsrate	<Double>	Read-only	Rate (/s) counter for lsntotdslitetxpkts
lsntotdslitetxbytes	<Double>	Read-only	Total number of LSN DS-Lite tx bytes.
lsndslitetxbytesrate	<Double>	Read-only	Rate (/s) counter for lsntotdslitetxbytes
lsntotdslitetcprxpkts	<Double>	Read-only	Number of LSN DS-Lite TCP Received packets.
lsndslitetcprxpktsrate	<Double>	Read-only	Rate (/s) counter for lsntotdslitetcprxpkts
lsntotdslitetcprxbytes	<Double>	Read-only	Number of LSN DS-Lite TCP Received bytes.
lsndslitetcprxbytesrate	<Double>	Read-only	Rate (/s) counter for lsntotdslitetcprxbytes
lsntotdslitetcptxpkts	<Double>	Read-only	Number of LSN DS-Lite TCP Transmitted packets.
lsndslitetcptxpktsrate	<Double>	Read-only	Rate (/s) counter for lsntotdslitetcptxpkts
lsntotdslitetcptxbytes	<Double>	Read-only	Number of LSN DS-Lite TCP Transmitted bytes.
lsndslitetcptxbytesrate	<Double>	Read-only	Rate (/s) counter for lsntotdslitetcptxbytes
lsntotdslitetcpdrppkts	<Double>	Read-only	Number of LSN DS-Lite TCP Dropped packets.

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
lsndslitetcpdrppktsrate	<Double>	Read-only	Rate (/s) counter for lsntotdslitetcpdrppkts
lsncurdslitetcpsessions	<Double>	Read-only	Number of LSN DS-Lite TCP Current Sessions.
lsncurdslitetcpsessionsrate	<Double>	Read-only	Rate (/s) counter for lsncurdslitetcpsessions
lsntotdsliteudprxpkts	<Double>	Read-only	Number of LSN DS-Lite UDP Received packets.
lsndsliteudprxpktsrate	<Double>	Read-only	Rate (/s) counter for lsntotdsliteudprxpkts
lsntotdsliteudprxbytes	<Double>	Read-only	Number of LSN DS-Lite UDP Received bytes.
lsndsliteudprxbytesrate	<Double>	Read-only	Rate (/s) counter for lsntotdsliteudprxbytes
lsntotdsliteudptxpkts	<Double>	Read-only	Number of LSN DS-Lite UDP Transmitted packets.
lsndsliteudptxpktsrate	<Double>	Read-only	Rate (/s) counter for lsntotdsliteudptxpkts
lsntotdsliteudptxbytes	<Double>	Read-only	Number of LSN DS-Lite UDP Transmitted bytes.
lsndsliteudptxbytesrate	<Double>	Read-only	Rate (/s) counter for lsntotdsliteudptxbytes

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
lsntotdsliteudpdrppkts	<Double>	Read-only	Number of LSN DS-Lite UDP Dropped packets.
lsndsliteudpdrppktsrate	<Double>	Read-only	Rate (/s) counter for lsntotdsliteudpdrppkts
lsncurdsliteudpsessions	<Double>	Read-only	Number of LSN DS-Lite UDP Current Sessions.
lsncurdsliteudpsessionsrate	<Double>	Read-only	Rate (/s) counter for lsncurdsliteudpsessions
lsntotdsliteicmprxpkts	<Double>	Read-only	Number of LSN DS-Lite ICMP Received packets.
lsndsliteicmprxpktsrate	<Double>	Read-only	Rate (/s) counter for lsntotdsliteicmprxpkts
lsntotdsliteicmprxbytes	<Double>	Read-only	Number of LSN DS-Lite ICMP Received bytes.
lsndsliteicmprxbytesrate	<Double>	Read-only	Rate (/s) counter for lsntotdsliteicmprxbytes
lsntotdsliteicmptxpkts	<Double>	Read-only	Number of LSN DS-Lite ICMP Transmitted packets.
lsndsliteicmptxpktsrate	<Double>	Read-only	Rate (/s) counter for lsntotdsliteicmptxpkts

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
lsntotdsliteicmptxbytes	<Double>	Read-only	Number of LSN DS-Lite ICMP Transmitted bytes.
lsndsliteicmptxbytesrate	<Double>	Read-only	Rate (/s) counter for lsntotdsliteicmptxbytes
lsntotdsliteicmpdrppkts	<Double>	Read-only	Number of LSN DS-Lite ICMP Dropped packets.
lsndsliteicmpdrppktsrate	<Double>	Read-only	Rate (/s) counter for lsntotdsliteicmpdrppkts
lsncurdsliteicmpsessions	<Double>	Read-only	Number of LSN DS-Lite ICMP Current Sessions.
lsncurdsliteicmpsessionsrate	<Double>	Read-only	Rate (/s) counter for lsncurdsliteicmpsessions
lsncurdslitesessions	<Double>	Read-only	Current number of LSN DS-Lite sessions.
lsncurdslitesessionsrate	<Double>	Read-only	Rate (/s) counter for lsncurdslitesessions
lsndslitecursubscribers	<Double>	Read-only	Current number of LSN DS-Lite subscribers.
lsndslitecursubscribersrate	<Double>	Read-only	Rate (/s) counter for lsndslitecursubscribers

Operations

(click to see Properties)

GET (ALL)

Some options that you can use for each operations:

- **Getting warnings in response:**NITRO allows you to get warnings in an operation by specifying the "warning" query parameter as "yes". For example, to get warnings while connecting to the Citrix ADC appliance, the URL is as follows:

  http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/login?warning=yes

  If any, the warnings are displayed in the response payload with the HTTP code "209 X-NITRO-WARNING".

- **Authenticated access for individual NITRO operations:**NITRO allows you to logon to the Citrix ADC appliance to perform individual operations. You can use this option instead of creating a NITRO session (using the login object) and then using that session to perform all operations,

  To do this, you must specify the username and password in the request header of the NITRO request as follows:

  X-NITRO-USER:<username>

  X-NITRO-PASS:<password>

  Note: In such cases, make sure that the request header DOES not include the following:

  Cookie:NITRO_AUTH_TOKEN=<tokenvalue>

`*Note:`*

Mandatory parameters are marked in redand placeholder content is marked in <green>.

get (all)

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/stat/lsndslite

Query-parameters:

args

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/stat/lsndslite?
args=detail:<Boolean_value>,fullvalues:<Boolean_value>,ntimes:<Double_value>,logfile:<String_value

Use this query-parameter to get lsndslite resources based on additional properties.

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "lsndslite": [ {

"lsntotdsliteudpdrppkts":<Double_value>,

"lsndsliteudprxpktsrate":<Double_value>,

"lsntotdsliterxpkts":<Double_value>,

"lsntotdslitetxbytes":<Double_value>,

"lsncurdsliteicmpsessionsrate":<Double_value>,

"lsntotdsliteudptxbytes":<Double_value>,

"lsntotdsliteicmptxpkts":<Double_value>,

"lsndslitetxpktsrate":<Double_value>,

"lsntotdsliteicmptxbytes":<Double_value>,

"lsntotdsliterxbytes":<Double_value>,

"lsntotdsliteicmpdrppkts":<Double_value>,

"lsncurdslitetcpsessionsrate":<Double_value>,
```

```
"lsndsliteudptxbytesrate":<Double_value>,

"lsntotdsliteudprxbytes":<Double_value>,

"lsndsliterxbytesrate":<Double_value>,

"lsncurdsliteicmpsessions":<Double_value>,

"lsndsliteicmprxbytesrate":<Double_value>,

"lsndsliteudpdrppktsrate":<Double_value>,

"lsntotdslitetcpdrppkts":<Double_value>,

"lsntotdsliteicmprxbytes":<Double_value>,

"lsncurdslitesessionsrate":<Double_value>,

"lsntotdsliteicmprxpkts":<Double_value>,

"lsntotdslitetxpkts":<Double_value>,

"lsncurdsliteudpsessionsrate":<Double_value>,

"lsndslitetcptxpktsrate":<Double_value>,

"lsntotdslitetcptxpkts":<Double_value>,

"lsndsliteicmptxpktsrate":<Double_value>,

"lsndsliteicmpdrppktsrate":<Double_value>,

"lsntotdsliteudprxpkts":<Double_value>,

"lsncurdslitetcpsessions":<Double_value>,

"lsndslitetxbytesrate":<Double_value>,

"lsntotdslitetcprxbytes":<Double_value>,

"lsndsliteicmptxbytesrate":<Double_value>,

"lsndsliteicmprxpktsrate":<Double_value>,
```

```
"lsncurdslitesessions":<Double_value>,

"lsndslitetcprxbytesrate":<Double_value>,

"lsndslitecursubscribers":<Double_value>,

"lsntotdslitetcprxpkts":<Double_value>,

"lsndsliterxpktsrate":<Double_value>,

"lsncurdsliteudpsessions":<Double_value>,

"lsndslitetcprxpktsrate":<Double_value>,

"lsntotdsliteudptxpkts":<Double_value>,

"lsntotdslitetcptxbytes":<Double_value>,

"lsndslitetcpdrppktsrate":<Double_value>,

"lsndslitetcptxbytesrate":<Double_value>,

"lsndsliteudprxbytesrate":<Double_value>,

"lsndsliteudptxpktsrate":<Double_value>,

"lsndslitecursubscribersrate":<Double_value>

}]}
```

lsngroup

Statistics for LSN group resource.

Properties

(click to see Operations)

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
groupname	<String>	Read-write	Name of the LSN Group. Minimum length = 1 Maximum length = 127
clearstats	<String>	Read-write	Clear the statsistics / counters. Possible values = basic, full
lsngrpcursessions	<Double>	Read-only	Number of Current Sessions for LSN group
lsngrpcursessionsrate	<Double>	Read-only	Rate (/s) counter for lsngrpcursessions
lsngrptottranslbytes	<Double>	Read-only	Number of Translated Bytes for LSN group
lsngrptranslbytesrate	<Double>	Read-only	Rate (/s) counter for lsngrptottranslbytes
lsngrptottranslpkts	<Double>	Read-only	Number of Translated Pkts for LSN group
lsngrptranslpktsrate	<Double>	Read-only	Rate (/s) counter for lsngrptottranslpkts
lsngrptottcptranslpkts	<Double>	Read-only	Number of TCP Translated Pkts for LSN group
lsngrptcptranslpktsrate	<Double>	Read-only	Rate (/s) counter for lsngrptottcptranslpkts

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
lsngrptottcptranslbytes	<Double>	Read-only	Number of TCP Translated Bytes for LSN group
lsngrptcptranslbytesrate	<Double>	Read-only	Rate (/s) counter for lsngrptottcptranslbytes
lsngrptottcpdrppkts	<Double>	Read-only	Number of TCP Dropped Pkts for LSN group
lsngrptcpdrppktsrate	<Double>	Read-only	Rate (/s) counter for lsngrptottcpdrppkts
lsngrpcurtcpsessions	<Double>	Read-only	Number of TCP Current Sessions for LSN group
lsngrpcurtcpsessionsrate	<Double>	Read-only	Rate (/s) counter for lsngrpcurtcpsessions
lsngrptotudptranslpkts	<Double>	Read-only	Number of UDP Translated Pkts for LSN group
lsngrpudptranslpktsrate	<Double>	Read-only	Rate (/s) counter for lsngrptotudptranslpkts
lsngrptotudptranslbytes	<Double>	Read-only	Number of UDP Translated Bytes for LSN group
lsngrpudptranslbytesrate	<Double>	Read-only	Rate (/s) counter for lsngrptotudptranslbytes
lsngrptotudpdrppkts	<Double>	Read-only	Number of UDP Dropped Pkts for LSN group

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
lsngrpudpdrppktsrate	<Double>	Read-only	Rate (/s) counter for lsngrptotudpdrppkts
lsngrpcurudpsessions	<Double>	Read-only	Number of UDP Current Sessions for LSN group
lsngrpcurudpsessionsrate	<Double>	Read-only	Rate (/s) counter for lsngrpcurudpsessions
lsngrptoticmptranslpkts	<Double>	Read-only	Number of ICMP Translated Pkts for LSN group
lsngrpicmptranslpktsrate	<Double>	Read-only	Rate (/s) counter for lsngrptoticmptranslpkts
lsngrptoticmptranslbytes	<Double>	Read-only	Number of ICMP Translated Bytes for LSN group
lsngrpicmptranslbytesrate	<Double>	Read-only	Rate (/s) counter for lsngrptoticmptranslbytes
lsngrptoticmpdrppkts	<Double>	Read-only	Number of ICMP Dropped Pkts for LSN group
lsngrpicmpdrppktsrate	<Double>	Read-only	Rate (/s) counter for lsngrptoticmpdrppkts
lsngrpcuricmpsessions	<Double>	Read-only	Number of ICMP Current Sessions for LSN group
lsngrpcuricmpsessionsrate	<Double>	Read-only	Rate (/s) counter for lsngrpcuricmpsessions

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
lsngrpcursubscribers	<Double>	Read-only	Number of ICMP Current Sessions for LSN group
lsngrpcursubscribers rate	<Double>	Read-only	Rate (/s) counter for lsngrpcursubscribers

Operations

(click to see Properties)

GET (ALL)| GET

Some options that you can use for each operations:

- **Getting warnings in response:**NITRO allows you to get warnings in an operation by specifying the "warning" query parameter as "yes". For example, to get warnings while connecting to the Citrix ADC appliance, the URL is as follows:

  http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/login?warning=yes

  If any, the warnings are displayed in the response payload with the HTTP code "209 X-NITRO-WARNING".

- **Authenticated access for individual NITRO operations:**NITRO allows you to logon to the Citrix ADC appliance to perform individual operations. You can use this option instead of creating a NITRO session (using the login object) and then using that session to perform all operations,

  To do this, you must specify the username and password in the request header of the NITRO request as follows:

  X-NITRO-USER:<username>

  X-NITRO-PASS:<password>

  Note: In such cases, make sure that the request header DOES not include the following:

  Cookie:NITRO_AUTH_TOKEN=<tokenvalue>

`*Note:`*

Mandatory parameters are marked in redand placeholder content is marked in <green>.

get (all)

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/stat/lsngroup

Query-parameters:

args

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/stat/lsngroup?
args=groupname:<String_value>,detail:<Boolean_value>,fullvalues:<Boolean_value>,ntimes:<Double_

Use this query-parameter to get lsngroup resources based on additional properties.

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "lsngroup": [ {

"groupname":<String_value>,

"lsngrptcptranslbytesrate":<Double_value>,

"lsngrptottranslpkts":<Double_value>,

"lsngrptoticmptranslpkts":<Double_value>,

"lsngrpcurtcpsessions":<Double_value>,

"lsngrpicmptranslbytesrate":<Double_value>,
```

```
"lsngrptotudptranslpkts":<Double_value>,

"lsngrptottcpdrppkts":<Double_value>,

"lsngrptranslbytesrate":<Double_value>,

"lsngrptotudptranslbytes":<Double_value>,

"lsngrptoticmptranslbytes":<Double_value>,

"lsngrpcursubscribersrate":<Double_value>,

"lsngrptottranslbytes":<Double_value>,

"lsngrptoticmpdrppkts":<Double_value>,

"lsngrptcptranslpktsrate":<Double_value>,

"lsngrpudptranslpktsrate":<Double_value>,

"lsngrpcursubscribers":<Double_value>,

"lsngrpudptranslbytesrate":<Double_value>,

"lsngrpcuricmpsessionsrate":<Double_value>,

"lsngrpcurudpsessions":<Double_value>,

"lsngrptottcptranslpkts":<Double_value>,

"lsngrptotudpdrppkts":<Double_value>,

"lsngrpicmptranslpktsrate":<Double_value>,

"lsngrpcuricmpsessions":<Double_value>,

"lsngrpcurtcpsessionsrate":<Double_value>,

"lsngrpcurudpsessionsrate":<Double_value>,

"lsngrptranslpktsrate":<Double_value>,

"lsngrptottcptranslbytes":<Double_value>,
```

```
"lsngrpicmpdrppktsrate":<Double_value>,

"lsngrpcursessionsrate":<Double_value>,

"lsngrpudpdrppktsrate":<Double_value>,

"lsngrptcpdrppktsrate":<Double_value>,

"lsngrpcursessions":<Double_value>

}]}
```

get

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/stat/lsngroup/groupname_value><String>

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "lsngroup": [ {

"groupname":<String_value>,

"lsngrptcptranslbytesrate":<Double_value>,

"lsngrptottranslpkts":<Double_value>,
```

```
"lsngrptoticmptranslpkts":<Double_value>,

"lsngrpcurtcpsessions":<Double_value>,

"lsngrpicmptranslbytesrate":<Double_value>,

"lsngrptotudptranslpkts":<Double_value>,

"lsngrptottcpdrppkts":<Double_value>,

"lsngrptranslbytesrate":<Double_value>,

"lsngrptotudptranslbytes":<Double_value>,

"lsngrptoticmptranslbytes":<Double_value>,

"lsngrpcursubscribersrate":<Double_value>,

"lsngrptottranslbytes":<Double_value>,

"lsngrptoticmpdrppkts":<Double_value>,

"lsngrptcptranslpktsrate":<Double_value>,

"lsngrpudptranslpktsrate":<Double_value>,

"lsngrpcursubscribers":<Double_value>,

"lsngrpudptranslbytesrate":<Double_value>,

"lsngrpcuricmpsessionsrate":<Double_value>,

"lsngrpcurudpsessions":<Double_value>,

"lsngrptottcptranslpkts":<Double_value>,

"lsngrptotudpdrppkts":<Double_value>,

"lsngrpicmptranslpktsrate":<Double_value>,

"lsngrpcuricmpsessions":<Double_value>,

"lsngrpcurtcpsessionsrate":<Double_value>,
```

```
"lsngrpcurudpsessionsrate":<Double_value>,

"lsngrptranslpktsrate":<Double_value>,

"lsngrptottcptranslbytes":<Double_value>,

"lsngrpicmpdrppktsrate":<Double_value>,

"lsngrpcursessionsrate":<Double_value>,

"lsngrpudpdrppktsrate":<Double_value>,

"lsngrptcpdrppktsrate":<Double_value>,

"lsngrpcursessions":<Double_value>

}]}
```

lsnpool

Statistics for LSN pool resource.

Properties

(click to see Operations)

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
poolname	<String>	Read-write	Name of the LSN Pool. Minimum length = 1 Maximum length = 127
clearstats	<String>	Read-write	Clear the statsistics / counters. Possible values = basic, full

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
lsnpoolotherportusagepercen	<Double>	Read-only	Other protocols ports usage percentage in this pool.
lsnpooltcpportusagepercen	<Double>	Read-only	TCP ports usage percentage in this pool.
lsnpoolcurportalloctcp	<Double>	Read-only	Current port allocation for tcp in this pool.
lsnpoolcurportalloctcprate	<Double>	Read-only	Rate (/s) counter for lsnpoolcurportalloctcp
lsnpoolcurportallocother	<Double>	Read-only	Current port allocation for other protocols in this pool.
lsnpoolcurportallocotherrate	<Double>	Read-only	Rate (/s) counter for lsnpoolcurportallocother
lsnpooltotips	<Double>	Read-only	Total IPs in this pool.

Operations

(click to see Properties)

GET (ALL)| GET

Some options that you can use for each operations:

- **Getting warnings in response:**NITRO allows you to get warnings in an operation by specifying the "warning" query parameter as "yes". For example, to get warnings while connecting to the Citrix ADC appliance, the URL is as follows:

  http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/login?warning=yes

  If any, the warnings are displayed in the response payload with the HTTP code "209 X-NITRO-WARNING".

- **Authenticated access for individual NITRO operations:**NITRO allows you to logon to the Citrix ADC appliance to perform individual operations. You can use this option instead of creating a NITRO session (using the login object) and then using that session to perform all operations,

  To do this, you must specify the username and password in the request header of the NITRO request as follows:

  X-NITRO-USER:<username>

  X-NITRO-PASS:<password>

  Note: In such cases, make sure that the request header DOES not include the following:

  Cookie:NITRO_AUTH_TOKEN=<tokenvalue>

`*Note:`*

Mandatory parameters are marked in redand placeholder content is marked in <green>.

get (all)

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/stat/lsnpool

Query-parameters:

args

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/stat/lsnpool?
args=poolname:<String_value>,detail:<Boolean_value>,fullvalues:<Boolean_value>,ntimes:<Double_va

Use this query-parameter to get lsnpool resources based on additional properties.

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "lsnpool": [ {

"poolname":<String_value>,

"lsnpooltotips":<Double_value>,

"lsnpoolcurportallocother":<Double_value>,

"lsnpoolcurportalloctcp":<Double_value>,

"lsnpooltcpportusagepercen":<Double_value>,

"lsnpoolcurportalloctcprate":<Double_value>,

"lsnpoolotherportusagepercen":<Double_value>,

"lsnpoolcurportallocotherrate":<Double_value>

}]}
```

get

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/stat/lsnpool/poolname_value><String>

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "lsnpool": [ {

"poolname":<String_value>,

"lsnpooltotips":<Double_value>,

"lsnpoolcurportallocother":<Double_value>,

"lsnpoolcurportalloctcp":<Double_value>,

"lsnpooltcpportusagepercen":<Double_value>,

"lsnpoolcurportalloctcprate":<Double_value>,

"lsnpoolotherportusagepercen":<Double_value>,

"lsnpoolcurportallocotherrate":<Double_value>

}]}
```

lsnnat64

Read/write properties

Properties

(click to see Operations)

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
clearstats	<String>	Read-write	Clear the statsistics / counters. Possible values = basic, full
lsncurnat64sessions	<Double>	Read-only	Current number of LSN NAT64 sessions.

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
lsncurnat64sessionsrate	<Double>	Read-only	Rate (/s) counter for lsncurnat64sessions
lsnnat64cursubscribers	<Double>	Read-only	Current number of LSN NAT64 subscribers.
lsnnat64cursubscribersrate	<Double>	Read-only	Rate (/s) counter for lsnnat64cursubscribers
lsntotnat64rxpkts	<Double>	Read-only	Total number of LSN NAT64 rx pkts.
lsnnat64rxpktsrate	<Double>	Read-only	Rate (/s) counter for lsntotnat64rxpkts
lsntotnat64rxbytes	<Double>	Read-only	Total number of LSN NAT64 rx bytes.
lsnnat64rxbytesrate	<Double>	Read-only	Rate (/s) counter for lsntotnat64rxbytes
lsntotnat64txpkts	<Double>	Read-only	Total number of LSN NAT64 tx pkts.
lsnnat64txpktsrate	<Double>	Read-only	Rate (/s) counter for lsntotnat64txpkts
lsntotnat64txbytes	<Double>	Read-only	Total number of LSN NAT64 tx bytes.
lsnnat64txbytesrate	<Double>	Read-only	Rate (/s) counter for lsntotnat64txbytes
lsncurnat64tcpsessions	<Double>	Read-only	Number of LSN NAT64 TCP Current Sessions.

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
lsncurnat64tcpsessionsrate	<Double>	Read-only	Rate (/s) counter for lsncurnat64tcpsessions
lsntotnat64tcprxpkts	<Double>	Read-only	Number of LSN NAT64 TCP Received packets.
lsnnat64tcprxpktsrate	<Double>	Read-only	Rate (/s) counter for lsntotnat64tcprxpkts
lsntotnat64tcprxbytes	<Double>	Read-only	Number of LSN NAT64 TCP Received bytes.
lsnnat64tcprxbytesrate	<Double>	Read-only	Rate (/s) counter for lsntotnat64tcprxbytes
lsntotnat64tcptxpkts	<Double>	Read-only	Number of LSN NAT64 TCP Transmitted packets.
lsnnat64tcptxpktsrate	<Double>	Read-only	Rate (/s) counter for lsntotnat64tcptxpkts
lsntotnat64tcptxbytes	<Double>	Read-only	Number of LSN NAT64 TCP Transmitted bytes.
lsnnat64tcptxbytesrate	<Double>	Read-only	Rate (/s) counter for lsntotnat64tcptxbytes
lsntotnat64tcpdrppkts	<Double>	Read-only	Number of LSN NAT64 TCP Dropped packets.
lsnnat64tcpdrppktsrate	<Double>	Read-only	Rate (/s) counter for lsntotnat64tcpdrppkts

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
lsncurnat64udpsessions	<Double>	Read-only	Number of LSN NAT64 UDP Current Sessions.
lsncurnat64udpsessionsrate	<Double>	Read-only	Rate (/s) counter for lsncurnat64udpsessions
lsntotnat64udprxpkts	<Double>	Read-only	Number of LSN NAT64 UDP Received packets.
lsnnat64udprxpktsrate	<Double>	Read-only	Rate (/s) counter for lsntotnat64udprxpkts
lsntotnat64udprxbytes	<Double>	Read-only	Number of LSN NAT64 UDP Received bytes.
lsnnat64udprxbytesrate	<Double>	Read-only	Rate (/s) counter for lsntotnat64udprxbytes
lsntotnat64udptxpkts	<Double>	Read-only	Number of LSN NAT64 UDP Transmitted packets.
lsnnat64udptxpktsrate	<Double>	Read-only	Rate (/s) counter for lsntotnat64udptxpkts
lsntotnat64udptxbytes	<Double>	Read-only	Number of LSN NAT64 UDP Transmitted bytes.
lsnnat64udptxbytesrate	<Double>	Read-only	Rate (/s) counter for lsntotnat64udptxbytes
lsntotnat64udpdrppkts	<Double>	Read-only	Number of LSN NAT64 UDP Dropped packets.

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
lsnnat64udpdrppktsrate	<Double>	Read-only	Rate (/s) counter for lsntotnat64udpdrppkts
lsncurnat64icmpsessions	<Double>	Read-only	Number of LSN NAT64 ICMP Current Sessions.
lsncurnat64icmpsessionsrate	<Double>	Read-only	Rate (/s) counter for lsncurnat64icmpsessions
lsntotnat64icmprxpkts	<Double>	Read-only	Number of LSN NAT64 ICMP Received packets.
lsnnat64icmprxpktsrate	<Double>	Read-only	Rate (/s) counter for lsntotnat64icmprxpkts
lsntotnat64icmprxbytes	<Double>	Read-only	Number of LSN NAT64 ICMP Received bytes.
lsnnat64icmprxbytesrate	<Double>	Read-only	Rate (/s) counter for lsntotnat64icmprxbytes
lsntotnat64icmptxpkts	<Double>	Read-only	Number of LSN NAT64 ICMP Transmitted packets.
lsnnat64icmptxpktsrate	<Double>	Read-only	Rate (/s) counter for lsntotnat64icmptxpkts
lsntotnat64icmptxbytes	<Double>	Read-only	Number of LSN NAT64 ICMP Transmitted bytes.

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
lsnnat64icmptxbytesrate	<Double>	Read-only	Rate (/s) counter for lsntotnat64icmptxbytes
lsntotnat64icmpdrppkts	<Double>	Read-only	Number of LSN NAT64 ICMP Dropped packets.
lsnnat64icmpdrppktsrate	<Double>	Read-only	Rate (/s) counter for lsntotnat64icmpdrppkts

Operations

(click to see Properties)

GET (ALL)

Some options that you can use for each operations:

- **Getting warnings in response:**NITRO allows you to get warnings in an operation by specifying the "warning" query parameter as "yes". For example, to get warnings while connecting to the Citrix ADC appliance, the URL is as follows:

  http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/login?warning=yes

  If any, the warnings are displayed in the response payload with the HTTP code "209 X-NITRO-WARNING".

- **Authenticated access for individual NITRO operations:**NITRO allows you to logon to the Citrix ADC appliance to perform individual operations. You can use this option instead of creating a NITRO session (using the login object) and then using that session to perform all operations,

  To do this, you must specify the username and password in the request header of the NITRO request as follows:

  X-NITRO-USER:<username>

  X-NITRO-PASS:<password>

  Note: In such cases, make sure that the request header DOES not include the following:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue>

`*Note:` *

Mandatory parameters are marked in redand placeholder content is marked in <green>.

get (all)

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/stat/lsnnat64

Query-parameters:

args

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/stat/lsnnat64?
args=detail:<Boolean_value>,fullvalues:<Boolean_value>,ntimes:<Double_value>,logfile:<String_value

Use this query-parameter to get lsnnat64 resources based on additional properties.

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "lsnnat64": [ {

"lsnnat64icmptxbytesrate":<Double_value>,

"lsntotnat64icmprxpkts":<Double_value>,

"lsnnat64tcptxpktsrate":<Double_value>,

"lsncurnat64icmpsessions":<Double_value>,
```

```
"lsnnat64rxpktsrate":<Double_value>,

"lsnnat64udptxbytesrate":<Double_value>,

"lsntotnat64udprxbytes":<Double_value>,

"lsntotnat64rxpkts":<Double_value>,

"lsnnat64txpktsrate":<Double_value>,

"lsntotnat64icmptxpkts":<Double_value>,

"lsntotnat64udpdrppkts":<Double_value>,

"lsncurnat64udpsessionsrate":<Double_value>,

"lsncurnat64tcpsessions":<Double_value>,

"lsntotnat64tcptxbytes":<Double_value>,

"lsntotnat64txbytes":<Double_value>,

"lsnnat64txbytesrate":<Double_value>,

"lsncurnat64sessions":<Double_value>,

"lsnnat64icmpdrppktsrate":<Double_value>,

"lsntotnat64udptxbytes":<Double_value>,

"lsncurnat64icmpsessionsrate":<Double_value>,

"lsnnat64tcprxpktsrate":<Double_value>,

"lsnnat64icmptxpktsrate":<Double_value>,

"lsnnat64udpdrppktsrate":<Double_value>,

"lsntotnat64txpkts":<Double_value>,

"lsncurnat64udpsessions":<Double_value>,

"lsntotnat64tcprxpkts":<Double_value>,
```

```
"lsnnat64tcpdrppktsrate":<Double_value>,

"lsnnat64tcptxbytesrate":<Double_value>,

"lsntotnat64icmptxbytes":<Double_value>,

"lsnnat64udptxpktsrate":<Double_value>,

"lsntotnat64tcptxpkts":<Double_value>,

"lsnnat64tcprxbytesrate":<Double_value>,

"lsnnat64icmprxpktsrate":<Double_value>,

"lsntotnat64udprxpkts":<Double_value>,

"lsntotnat64tcpdrppkts":<Double_value>,

"lsncurnat64tcpsessionsrate":<Double_value>,

"lsnnat64udprxpktsrate":<Double_value>,

"lsntotnat64icmpdrppkts":<Double_value>,

"lsnnat64udprxbytesrate":<Double_value>,

"lsnnat64icmprxbytesrate":<Double_value>,

"lsncurnat64sessionsrate":<Double_value>,

"lsnnat64rxbytesrate":<Double_value>,

"lsntotnat64icmprxbytes":<Double_value>,

"lsnnat64cursubscribers":<Double_value>,

"lsnnat64cursubscribersrate":<Double_value>,

"lsntotnat64udptxpkts":<Double_value>,

"lsntotnat64rxbytes":<Double_value>,

"lsntotnat64tcprxbytes":<Double_value>
```

```
}]}
```

Network

Networking commands.

NAME	DESCRIPTION
Interface	Statistics for interface resource.
bridge	Statistics for bridge resource.
inat	Statistics for inbound nat resource.
inatsession	Statistics for stateful INAT resource.
mapbmr	Statistics for MAP-T Basic Mapping rule resource.
mapdomain	Statistics for MAP-T Map Domain resource.
nat64	Statistics for nat64 config resource.
rnat	Statistics for RNAT configured route resource.
rnat6	Statistics for IPv6 RNAT configured route resource.
rnatip	Statistics for RNAT ipaddress resource.
tunnelip	Statistics for TUNNEL Remote ipaddress resource.
tunnelip6	Statistics for TUNNEL Remote ip6address resource.
vlan	Statistics for "VLAN" resource.
vxlan	Statistics for "VXLAN" resource.

bridge

Statistics for bridge resource.

Properties

(click to see Operations)

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
clearstats	<String>	Read-write	Clear the statsistics / counters. Possible values = basic, full
tcptotbdgmacmoved	<Double>	Read-only	The number of times bridging registered MAC moved
tcpbdgmacmovedrate	<Double>	Read-only	Rate (/s) counter for tcptotbdgmacmoved
tcptotbdgcollisions	<Double>	Read-only	The number of bridging table collisions
tcpbdgcollisionsrate	<Double>	Read-only	Rate (/s) counter for tcptotbdgcollisions
tcperrbdgmuted	<Double>	Read-only	The number of bridging related interface mutes
tcperrbdgmutedrate	<Double>	Read-only	Rate (/s) counter for tcperrbdgmuted
totbdgpkts	<Double>	Read-only	The total number of bridged packets
bdgpktsrate	<Double>	Read-only	Rate (/s) counter for totbdgpkts
totbdgmbits	<Double>	Read-only	The total number of bridged Mbits

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
bdgmbitsrate	<Double>	Read-only	Rate (/s) counter for totbdgmbits

Operations

(click to see Properties)

GET (ALL)

Some options that you can use for each operations:

- **Getting warnings in response:**NITRO allows you to get warnings in an operation by specifying the "warning" query parameter as "yes". For example, to get warnings while connecting to the Citrix ADC appliance, the URL is as follows:

  http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/login?warning=yes

  If any, the warnings are displayed in the response payload with the HTTP code "209 X-NITRO-WARNING".
- **Authenticated access for individual NITRO operations:**NITRO allows you to logon to the Citrix ADC appliance to perform individual operations. You can use this option instead of creating a NITRO session (using the login object) and then using that session to perform all operations,

  To do this, you must specify the username and password in the request header of the NITRO request as follows:

  X-NITRO-USER:<username>

  X-NITRO-PASS:<password>

  Note: In such cases, make sure that the request header DOES not include the following:

  Cookie:NITRO_AUTH_TOKEN=<tokenvalue>

`*Note:`*

Mandatory parameters are marked in redand placeholder content is marked in <green>.

get (all)

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/stat/bridge

Query-parameters:

args

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/stat/bridge?
args=detail:<Boolean_value>,fullvalues:<Boolean_value>,ntimes:<Double_value>,logfile:<String_value

Use this query-parameter to get bridge resources based on additional properties.

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "bridge": [ {

"bdgpktsrate":<Double_value>,

"totbdgmbits":<Double_value>,

"bdgmbitsrate":<Double_value>,

"totbdgpkts":<Double_value>,

"tcperrbdgmuted":<Double_value>,

"tcpbdgcollisionsrate":<Double_value>,

"tcptotbdgcollisions":<Double_value>,

"tcperrbdgmutedrate":<Double_value>,
```

```
"tcpbdgmacmovedrate":<Double_value>,

"tcptotbdgmacmoved":<Double_value>

}]}
```

inat

Statistics for inbound nat resource.

Properties

(click to see Operations)

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
name	<String>	Read-write	The INAT. Minimum length = 1
clearstats	<String>	Read-write	Clear the statsistics / counters. Possible values = basic, full
nat46tottcp46	<Double>	Read-only	Total TCP packets translated (V4->v6).
nat46tcp46rate	<Double>	Read-only	Rate (/s) counter for nat46tottcp46
nat46totudp46	<Double>	Read-only	Total UDP packets translated (V4->v6).
nat46udp46rate	<Double>	Read-only	Rate (/s) counter for nat46totudp46
nat46toticmp46	<Double>	Read-only	Total ICMP packets translated (V4->v6).
nat46icmp46rate	<Double>	Read-only	Rate (/s) counter for nat46toticmp46

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
nat46totdrop46	<Double>	Read-only	Total IPV4 packets dropped.
nat46drop46rate	<Double>	Read-only	Rate (/s) counter for nat46totdrop46
nat46tottcp64	<Double>	Read-only	Total TCP packets translated (V6->v4).
nat46tcp64rate	<Double>	Read-only	Rate (/s) counter for nat46tottcp64
nat46totudp64	<Double>	Read-only	Total UDP packets translated (V6->v4).
nat46udp64rate	<Double>	Read-only	Rate (/s) counter for nat46totudp64
nat46toticmp64	<Double>	Read-only	Total ICMP packets translated (V6->v4).
nat46icmp64rate	<Double>	Read-only	Rate (/s) counter for nat46toticmp64
nat46totdrop64	<Double>	Read-only	Total IPV6 packets dropped.
nat46drop64rate	<Double>	Read-only	Rate (/s) counter for nat46totdrop64
inatnat46tcp46	<Double>	Read-only	TCP packets translated (V4->v6).
inatnat46tcp46rate	<Double>	Read-only	Rate (/s) counter for inatnat46tcp46
inatnat46udp46	<Double>	Read-only	UDP packets translated (V4->v6).
inatnat46udp46rate	<Double>	Read-only	Rate (/s) counter for inatnat46udp46

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
inatnat46icmp46	<Double>	Read-only	ICMP packets translated (V4->v6).
inatnat46icmp46rate	<Double>	Read-only	Rate (/s) counter for inatnat46icmp46
inatnat46drop46	<Double>	Read-only	IPV4 packets dropped.
inatnat46drop46rate	<Double>	Read-only	Rate (/s) counter for inatnat46drop46
inatnat46tcp64	<Double>	Read-only	TCP packets translated (V6->v4).
inatnat46tcp64rate	<Double>	Read-only	Rate (/s) counter for inatnat46tcp64
inatnat46udp64	<Double>	Read-only	UDP packets translated (V6->v4).
inatnat46udp64rate	<Double>	Read-only	Rate (/s) counter for inatnat46udp64
inatnat46icmp64	<Double>	Read-only	ICMP packets translated (V6->v4).
inatnat46icmp64rate	<Double>	Read-only	Rate (/s) counter for inatnat46icmp64
inatnat46drop64	<Double>	Read-only	IPV6 packets dropped.
inatnat46drop64rate	<Double>	Read-only	Rate (/s) counter for inatnat46drop64

Operations

(click to see Properties)

GET (ALL)| GET

Some options that you can use for each operations:

- **Getting warnings in response:**NITRO allows you to get warnings in an operation by specifying the "warning" query parameter as "yes". For example, to get warnings while connecting to the Citrix ADC appliance, the URL is as follows:

  http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/login?warning=yes

  If any, the warnings are displayed in the response payload with the HTTP code "209 X-NITRO-WARNING".

- **Authenticated access for individual NITRO operations:**NITRO allows you to logon to the Citrix ADC appliance to perform individual operations. You can use this option instead of creating a NITRO session (using the login object) and then using that session to perform all operations,

  To do this, you must specify the username and password in the request header of the NITRO request as follows:

  X-NITRO-USER:<username>

  X-NITRO-PASS:<password>

  Note: In such cases, make sure that the request header DOES not include the following:

  Cookie:NITRO_AUTH_TOKEN=<tokenvalue>

`*Note:`*

Mandatory parameters are marked in redand placeholder content is marked in <green>.

get (all)

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/stat/inat

Query-parameters:

args

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/stat/inat?
args=name:<String_value>,detail:<Boolean_value>,fullvalues:<Boolean_value>,ntimes:<Double_value>

Use this query-parameter to get inat resources based on additional properties.

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "inat": [ {

"name":<String_value>,

"nat46udp64rate":<Double_value>,

"nat46totdrop64":<Double_value>,

"nat46totudp46":<Double_value>,

"nat46icmp64rate":<Double_value>,

"inatnat46tcp46":<Double_value>,

"nat46totdrop46":<Double_value>,

"inatnat46tcp64":<Double_value>,

"inatnat46drop46":<Double_value>,

"inatnat46drop64":<Double_value>,

"inatnat46udp46":<Double_value>,

"nat46tottcp64":<Double_value>,

"nat46drop46rate":<Double_value>,
```

```
"inatnat46drop46rate":<Double_value>,

"inatnat46tcp64rate":<Double_value>,

"inatnat46icmp64":<Double_value>,

"nat46drop64rate":<Double_value>,

"inatnat46tcp46rate":<Double_value>,

"nat46tottcp46":<Double_value>,

"nat46toticmp46":<Double_value>,

"inatnat46icmp46rate":<Double_value>,

"inatnat46udp64":<Double_value>,

"nat46tcp46rate":<Double_value>,

"inatnat46drop64rate":<Double_value>,

"inatnat46udp64rate":<Double_value>,

"nat46icmp46rate":<Double_value>,

"inatnat46icmp46":<Double_value>,

"inatnat46icmp64rate":<Double_value>,

"nat46tcp64rate":<Double_value>,

"nat46totudp64":<Double_value>,

"nat46udp46rate":<Double_value>,

"nat46toticmp64":<Double_value>,

"inatnat46udp46rate":<Double_value>

}]}
```

get

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/stat/inat/name_value><String>

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "inat": [ {

"name":<String_value>,

"nat46udp64rate":<Double_value>,

"nat46totdrop64":<Double_value>,

"nat46totudp46":<Double_value>,

"nat46icmp64rate":<Double_value>,

"inatnat46tcp46":<Double_value>,

"nat46totdrop46":<Double_value>,

"inatnat46tcp64":<Double_value>,

"inatnat46drop46":<Double_value>,
```

```
"inatnat46drop64":<Double_value>,

"inatnat46udp46":<Double_value>,

"nat46tottcp64":<Double_value>,

"nat46drop46rate":<Double_value>,

"inatnat46drop46rate":<Double_value>,

"inatnat46tcp64rate":<Double_value>,

"inatnat46icmp64":<Double_value>,

"nat46drop64rate":<Double_value>,

"inatnat46tcp46rate":<Double_value>,

"nat46tottcp46":<Double_value>,

"nat46toticmp46":<Double_value>,

"inatnat46icmp46rate":<Double_value>,

"inatnat46udp64":<Double_value>,

"nat46tcp46rate":<Double_value>,

"inatnat46drop64rate":<Double_value>,

"inatnat46udp64rate":<Double_value>,

"nat46icmp46rate":<Double_value>,

"inatnat46icmp46":<Double_value>,

"inatnat46icmp64rate":<Double_value>,

"nat46tcp64rate":<Double_value>,

"nat46totudp64":<Double_value>,

"nat46udp46rate":<Double_value>,
```

```
"nat46toticmp64":<Double_value>,

"inatnat46udp46rate":<Double_value>

}]}
```

inatsession

Statistics for stateful INAT resource.

Properties

(click to see Operations)

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
name	<String>	Read-write	INAT name. Minimum length = 1
clearstats	<String>	Read-write	Clear the statsistics / counters. Possible values = basic, full
globalinathits	<Double>	Read-only	Total INAT Session hits
globalinathitsrate	<Double>	Read-only	Rate (/s) counter for globalinathits
globalinatcursessions	<Double>	Read-only	Total INAT current sessions
globalinatcursessionsrate	<Double>	Read-only	Rate (/s) counter for globalinatcursessions
globalinatreceivebytes	<Double>	Read-only	Total INAT Received Bytes

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
globalinatreceivebytesrate	<Double>	Read-only	Rate (/s) counter for globalinatreceivebytes
globalinattotsentbytes	<Double>	Read-only	Total INAT Sent Bytes
globalinatsentbytesrate	<Double>	Read-only	Rate (/s) counter for globalinattotsentbytes
globalinatpktreceived	<Double>	Read-only	Total INAT Packets Received
globalinatpktreceivedrate	<Double>	Read-only	Rate (/s) counter for globalinatpktreceived
globalinatpktsent	<Double>	Read-only	Total INAT Packets Sent
globalinatpktsentrate	<Double>	Read-only	Rate (/s) counter for globalinatpktsent
inattothits	<Double>	Read-only	INAT total sessions
inathitsrate	<Double>	Read-only	Rate (/s) counter for inattothits
inatcursessions	<Double>	Read-only	INAT current sessions
inatcursessionsrate	<Double>	Read-only	Rate (/s) counter for inatcursessions
inattotreceivebytes	<Double>	Read-only	INAT total Received Bytes
inatreceivebytesrate	<Double>	Read-only	Rate (/s) counter for inattotreceivebytes
inattotsentbytes	<Double>	Read-only	INAT total Sent Bytes

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
inatsentbytesrate	<Double>	Read-only	Rate (/s) counter for inattotsentbytes
inattotpktreceived	<Double>	Read-only	INAT total Packets Received
inatpktreceivedrate	<Double>	Read-only	Rate (/s) counter for inattotpktreceived
inattotpktsent	<Double>	Read-only	INAT total Packets Sent
inatpktsentrate	<Double>	Read-only	Rate (/s) counter for inattotpktsent

Operations

(click to see Properties)

GET (ALL)| GET

Some options that you can use for each operations:

- **Getting warnings in response:**NITRO allows you to get warnings in an operation by specifying the "warning" query parameter as "yes". For example, to get warnings while connecting to the Citrix ADC appliance, the URL is as follows:

  http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/login?warning=yes

  If any, the warnings are displayed in the response payload with the HTTP code "209 X-NITRO-WARNING".

- **Authenticated access for individual NITRO operations:**NITRO allows you to logon to the Citrix ADC appliance to perform individual operations. You can use this option instead of creating a NITRO session (using the login object) and then using that session to perform all operations,

  To do this, you must specify the username and password in the request header of the NITRO request as follows:

  X-NITRO-USER:<username>

X-NITRO-PASS:<password>

Note: In such cases, make sure that the request header DOES not include the following:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue>

`*Note:`*

Mandatory parameters are marked in redand placeholder content is marked in <green>.

get (all)

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/stat/inatsession

Query-parameters:

args

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/stat/inatsession?
args=name:<String_value>,detail:<Boolean_value>,fullvalues:<Boolean_value>,ntimes:<Double_value>

Use this query-parameter to get inatsession resources based on additional properties.

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "inatsession": [ {

"name":<String_value>,

"globalinatreceivebytes":<Double_value>,
```

```
"globalinatcursessions":<Double_value>,

"inattotsentbytes":<Double_value>,

"globalinatpktsent":<Double_value>,

"inatreceivebytesrate":<Double_value>,

"globalinatsentbytesrate":<Double_value>,

"globalinatcursessionsrate":<Double_value>,

"inathitsrate":<Double_value>,

"inattotreceivebytes":<Double_value>,

"inatcursessionsrate":<Double_value>,

"globalinatreceivebytesrate":<Double_value>,

"globalinatpktsentrate":<Double_value>,

"inatcursessions":<Double_value>,

"inatsentbytesrate":<Double_value>,

"globalinattotsentbytes":<Double_value>,

"globalinathits":<Double_value>,

"globalinathitsrate":<Double_value>,

"inatpktsentrate":<Double_value>,

"globalinatpktreceived":<Double_value>,

"globalinatpktreceivedrate":<Double_value>,

"inattotpktreceived":<Double_value>,

"inattothits":<Double_value>,

"inattotpktsent":<Double_value>,
```

```
"inatpktreceivedrate":<Double_value>

}]}
```

get

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/stat/inatsession/name_value><String>

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "inatsession": [ {

"name":<String_value>,

"globalinatreceivebytes":<Double_value>,

"globalinatcursessions":<Double_value>,

"inattotsentbytes":<Double_value>,

"globalinatpktsent":<Double_value>,

"inatreceivebytesrate":<Double_value>,

"globalinatsentbytesrate":<Double_value>,
```

```
"globalinatcursessionsrate":<Double_value>,

"inathitsrate":<Double_value>,

"inattotreceivebytes":<Double_value>,

"inatcursessionsrate":<Double_value>,

"globalinatreceivebytesrate":<Double_value>,

"globalinatpktsentrate":<Double_value>,

"inatcursessions":<Double_value>,

"inatsentbytesrate":<Double_value>,

"globalinattotsentbytes":<Double_value>,

"globalinathits":<Double_value>,

"globalinathitsrate":<Double_value>,

"inatpktsentrate":<Double_value>,

"globalinatpktreceived":<Double_value>,

"globalinatpktreceivedrate":<Double_value>,

"inattotpktreceived":<Double_value>,

"inattothits":<Double_value>,

"inattotpktsent":<Double_value>,

"inatpktreceivedrate":<Double_value>

}]}
```

Interface

Statistics for interface resource.

Properties

(click to see Operations)

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
id	<String>	Read-write	Interface number, in C/U format, where C can take one of the following values: * 0 - Indicates a management interface. * 1 - Indicates a 1 Gbps port. * 10 - Indicates a 10 Gbps port. * LA - Indicates a link aggregation port. * LO - Indicates a loop back port. U is a unique integer for representing an interface in a particular port group.
clearstats	<String>	Read-write	Clear the statsistics / counters. Possible values = basic, full
curintfstate	<String>	Read-only	Current state of the specified interface. The interface state

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
			set to UP only if the link state is UP and administrative state is ENABLED .
curlinkuptime	<String>	Read-only	Duration for which the link is UP. This statistic is reset when the state changes to DOWN.
curlinkdowntime	<String>	Read-only	Duration for which the link is DOWN. This statistic is reset when the state changes to UP.
totrxbytes	<Double>	Read-only	Number of bytes received by an interface since the Citrix ADC was started or the interface statistics were cleared.
rxbytesrate	<Double>	Read-only	Rate (/s) counter for totrxbytes
tottxbytes	<Double>	Read-only	Number of bytes transmitted by an interface since the Citrix ADC was started or the interface statistics were cleared.
txbytesrate	<Double>	Read-only	Rate (/s) counter for tottxbytes
totrxpkts	<Double>	Read-only	Number of packets received by an interface since the

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
			Citrix ADC was started or the interface statistics were cleared.
rxpktsrate	<Double>	Read-only	Rate (/s) counter for totrxpkts
tottxpkts	<Double>	Read-only	Number of packets transmitted by an interface since the Citrix ADC was started or the interface statistics were cleared.
txpktsrate	<Double>	Read-only	Rate (/s) counter for tottxpkts
jumbopktsreceived	<Double>	Read-only	Number of Jumbo Packets received on this interface.
jumbopktsreceivedrate	<Double>	Read-only	Rate (/s) counter for jumbopktsreceived
jumbopktstransmitted	<Double>	Read-only	Number of Jumbo packets transmitted on this interface by upper layer, with TSO enabled actual trasmission size could be non Jumbo.
jumbopktstransmitteddrate	<Double>	Read-only	Rate (/s) counter for jumbopktstransmitted
trunkpktsreceived	<Double>	Read-only	Number of Tagged Packets received on this Trunk interface

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
			through Allowed VLan List.
trunkpktsreceivedrate	<Double>	Read-only	Rate (/s) counter for trunkpktsreceived
trunkpktstransmitted	<Double>	Read-only	Number of Tagged Packets transmitted on this Trunk interface through Allowed VLan List.
trunkpktstransmittedrate	<Double>	Read-only	Rate (/s) counter for trunkpktstransmitted
nictotmulticastpkts	<Double>	Read-only	Number of multicast packets received by the specified interface since the Citrix ADC was started or the interface statistics were cleared.
nicmulticastpktsrate	<Double>	Read-only	Rate (/s) counter for nictotmulticastpkts
totnetscalerpkts	<Double>	Read-only	Number of packets, destined to the Citrix ADC, received by an interface since the Citrix ADC was started or the interface statistics were cleared. The packets destined to Citrix ADC are those that have the same MAC address as that of an interface or a VMAC address

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
			owned by the Citrix ADC.
netscalerpktsrate	<Double>	Read-only	Rate (/s) counter for totnetscalerpkts
rxlacpdu	<Double>	Read-only	Number of Link Aggregation Control Protocol Data Units(LACPDUs) received by the specified interface since the Citrix ADC was started or the interface statistics were cleared.
rxlacpdurate	<Double>	Read-only	Rate (/s) counter for rxlacpdu
txlacpdu	<Double>	Read-only	Number of Link Aggregation Control Protocol Data Units(LACPDUs) transmitted by the specified interface since the Citrix ADC was started or the interface statistics were cleared.
txlacpdurate	<Double>	Read-only	Rate (/s) counter for txlacpdu
errpktrx	<Double>	Read-only	Number of inbound packets dropped by the hardware on a specified interface once the Citrix ADC starts or the interface statistics are cleared. This happens due to

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
			following reasons: 1) The hardware receives packets at a rate higher rate than that at which the software is processing packets. In this case, the hardware FIFO overruns and starts dropping the packets . 2) The specified interface fails to receive inbound packets from the appliance because of insufficient memory. 3) The specified interface receives packets with CRC errors (Alignment or Frame Check Sequence). 4) The specified interface receives overly long packets. 5) The specified interface receives packets with alignment errors. 6) The software does less buffering because it is running out of available memory. When hardware detects that there is no space into which to push newly arrived packets, it starts dropping them. 7)

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
			The specified interface receives packets with Frame Check Sequence (FCS) errors. 8) The specified interface receives packets smaller than 64 bytes. 9) The specified interface discards error-free inbound packets because of insufficient resources. For example: NIC buffers. 10) Packets are missed because of collision detection, link lost, physical decoding error, or MAC abort.
errpktrxrate	<Double>	Read-only	Rate (/s) counter for errpktrx
errpkttx	<Double>	Read-only	Number of outbound packets dropped by the hardware on a specified interface since the Citrix ADC was started or the interface statistics were cleared. This could happen due to length (undersize or oversize) errors and lack of resources. This statistic is available only for: (1) Loop back interface

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
			(LO) of all platforms. (2) All data ports on the Citrix ADC 12000 platform. (3) Management ports on the MPX 15000 and 17000 platforms.
errpkttxrate	<Double>	Read-only	Rate (/s) counter for errpkttx
errifindiscards	<Double>	Read-only	Number of error-free inbound packets discarded by the specified interface due to a lack of resources, for example, insufficient receive buffers.
errifindiscardsrate	<Double>	Read-only	Rate (/s) counter for errifindiscards
nicerrifoutdiscards	<Double>	Read-only	Number of error-free outbound packets discarded by the specified interface due to a lack of resources. This statistic is not available on: (1) 10G ports of Citrix ADC MPX 12500/12500/15500-10G platforms. (2) 10G data ports on Citrix ADC MPX 17500/19500/21500 platforms.

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
nicerrifoutdiscardsrate	<Double>	Read-only	Rate (/s) counter for nicerrifoutdiscards
errdroppedrxpkts	<Double>	Read-only	Number of inbound packets dropped by the specified interface. Commonly dropped packets are multicast frames, spanning tree BPDUs, packets destined to a MAC not owned by the Citrix ADC when L2 mode is disabled, or packets tagged for a VLAN that is not bound to the interface. This statistic will increment in most healthy networks at a steady rate regardless of traffic load. If a sharp spike in dropped packets occurs, it generally indicates an issue with connected L2 switches, such as a forwarding database overflow resulting in packets being broadcast on all ports.
errdroppedrxpktsrate	<Double>	Read-only	Rate (/s) counter for errdroppedrxpkts

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
errdroppedtxpkts	<Double>	Read-only	Number of packets dropped in transmission by the specified interface due to one of the following reasons. (1) VLAN mismatch. (2) Oversized packets. (3) Interface congestion. (4) Loopback packets sent on non loop back interface.
errdroppedtxpktsrate	<Double>	Read-only	Rate (/s) counter for errdroppedtxpkts
errlinkhangs	<Double>	Read-only	Number of times the specified interface detected hangs in the transmit and receive paths since the Citrix ADC was started or the interface statistics were cleared.
nicstsstalls	<Double>	Read-only	Number of times the status updates for a specified interface were stalled since the Citrix ADC was started or the interface statistics were cleared. A status stall is detected when the status of the interface is not updated by the NIC hardware within 0.8

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
			seconds of the last update.
nictxstalls	<Double>	Read-only	Number of times the interface stalled, when transmitting packets, since the Citrix ADC was started or the interface statistics were cleared. Transmit (Tx) stalls are detected when a packet posted for transmission is not transmitted in 4 seconds.
nicrxstalls	<Double>	Read-only	Number of times the interface stalled, when receiving packets, since the Citrix ADC was started or the interface statistics were cleared. Receive (Rx) stalls are detected when the following conditions are met: (1)The link is up for more than 10 minutes. (2)Packets are transmitted, but no packets is received for 16 seconds.
nicerrdisables	<Double>	Read-only	Number of times the specified interface is disabled by the Citrix

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
			ADC, due to continuous Receive (Rx) or Transmit (Tx) stalls, since the Citrix ADC was started or the interface statistics were cleared. The Citrix ADC disables an interface when one of the following conditions is met: (1) Three consecutive transmit stalls occurs with at most gap of 10 seconds between any two stalls. (2) Three consecutive receive stalls occurs with at most gap of 120 seconds between any two stalls.
errduplexmismatch	<Double>	Read-only	Number of times duplex mismatches were detected on the specified interface since the Citrix ADC was started or the interface statistics were cleared. A mismatch will occur if the duplex mode is not identically set on both ends of the link. This statistic is only available on the Citrix ADC Classic edition.

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
linkreinits	<Double>	Read-only	Number of times the link has been re-initialized. A re-initialization occurs due to link state change, configuration parameter change, or administrative reset operation.
totmacmoved	<Double>	Read-only	Number of MAC moves between ports. If a high rate of MAC moves is observed, it is likely that there is a bridge loop between two interfaces.
macmovedrate	<Double>	Read-only	Rate (/s) counter for totmacmoved
errnicmuted	<Double>	Read-only	Number of times the specified interface stopped transmitting and receiving packets due to MAC moves between ports.
rxcrcerrors	<Double>	Read-only	Number of packets received with the wrong checksum by the specified interface since the Citrix ADC was started or the interface statistics were cleared. This indicates the number

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
			of Jabber frames received instead of CRC errors on the 10G data ports of Citrix ADC 12000-10G platform and the data ports of Citrix ADC MPX 15000 and 17000 platforms.
rxcrcerrorsrate	<Double>	Read-only	Rate (/s) counter for rxcrcerrors
interfacealias	<String>	Read-only	Alias Name for the Interface
curlinkstate	<String>	Read-only	The current state of the link associated with the interface. For logical interfaces (LA), the state of the link is dependent on the state of the slave interfaces. For the link to be UP at least one of the slave interfaces needs to be UP.

Operations

(click to see Properties)

GET (ALL)| GET

Some options that you can use for each operations:

- **Getting warnings in response:**NITRO allows you to get warnings in an operation by specifying the "warning" query parameter as "yes". For example, to get warnings while connecting to the Citrix ADC appliance, the URL is as follows:

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/login?warning=yes

If any, the warnings are displayed in the response payload with the HTTP code "209 X-NITRO-WARNING".

- **Authenticated access for individual NITRO operations:**NITRO allows you to logon to the Citrix ADC appliance to perform individual operations. You can use this option instead of creating a NITRO session (using the login object) and then using that session to perform all operations,

  To do this, you must specify the username and password in the request header of the NITRO request as follows:

  X-NITRO-USER:<username>

  X-NITRO-PASS:<password>

  Note: In such cases, make sure that the request header DOES not include the following:

  Cookie:NITRO_AUTH_TOKEN=<tokenvalue>

`*Note:`*

Mandatory parameters are marked in redand placeholder content is marked in <green>.

get (all)

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/stat/Interface

Query-parameters:

args

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/stat/Interface?
args=id:<String_value>,detail:<Boolean_value>,fullvalues:<Boolean_value>,ntimes:<Double_value>,log

Use this query-parameter to get Interface resources based on additional properties.

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "Interface": [ {

"id":<String_value>,

"errpkttx":<Double_value>,

"nicrxstalls":<Double_value>,

"nictotmulticastpkts":<Double_value>,

"jumbopktstransmitted":<Double_value>,

"txbytesrate":<Double_value>,

"rxlacpdu":<Double_value>,

"netscalerpktsrate":<Double_value>,

"tottxpkts":<Double_value>,

"macmovedrate":<Double_value>,

"curintfstate":<String_value>,

"errpktrx":<Double_value>,

"curlinkdowntime":<String_value>,

"rxcrcerrorsrate":<Double_value>,

"nicmulticastpktsrate":<Double_value>,

"rxbytesrate":<Double_value>,
```

```
"curlinkuptime":<String_value>,

"tottxbytes":<Double_value>,

"errdroppedtxpktsrate":<Double_value>,

"errpkttxrate":<Double_value>,

"nictxstalls":<Double_value>,

"rxlacpdurate":<Double_value>,

"totnetscalerpkts":<Double_value>,

"trunkpktstransmittedrate":<Double_value>,

"jumbopktsreceivedrate":<Double_value>,

"totrxbytes":<Double_value>,

"trunkpktstransmitted":<Double_value>,

"linkreinits":<Double_value>,

"nicerrdisables":<Double_value>,

"txpktsrate":<Double_value>,

"jumbopktsreceived":<Double_value>,

"txlacpdu":<Double_value>,

"errdroppedrxpktsrate":<Double_value>,

"errnicmuted":<Double_value>,

"errdroppedtxpkts":<Double_value>,

"nicerrifoutdiscardsrate":<Double_value>,

"errifindiscardsrate":<Double_value>,

"trunkpktsreceivedrate":<Double_value>,
```

```
"nicerrifoutdiscards":<Double_value>,

"curlinkstate":<String_value>,

"rxcrcerrors":<Double_value>,

"errifindiscards":<Double_value>,

"interfacealias":<String_value>,

"totrxpkts":<Double_value>,

"totmacmoved":<Double_value>,

"rxpktsrate":<Double_value>,

"errduplexmismatch":<Double_value>,

"jumbopktstransmittedrate":<Double_value>,

"errpktrxrate":<Double_value>,

"errlinkhangs":<Double_value>,

"errdroppedrxpkts":<Double_value>,

"txlacpdurate":<Double_value>,

"nicstsstalls":<Double_value>,

"trunkpktsreceived":<Double_value>

}]}
```

get

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/stat/Interface/id_value><String>

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "Interface": [ {

"id":<String_value>,

"errpkttx":<Double_value>,

"nicrxstalls":<Double_value>,

"nictotmulticastpkts":<Double_value>,

"jumbopktstransmitted":<Double_value>,

"txbytesrate":<Double_value>,

"rxlacpdu":<Double_value>,

"netscalerpktsrate":<Double_value>,

"tottxpkts":<Double_value>,

"macmovedrate":<Double_value>,

"curintfstate":<String_value>,

"errpktrx":<Double_value>,

"curlinkdowntime":<String_value>,

"rxcrcerrorsrate":<Double_value>,
```

```
"nicmulticastpktsrate":<Double_value>,

"rxbytesrate":<Double_value>,

"curlinkuptime":<String_value>,

"tottxbytes":<Double_value>,

"errdroppedtxpktsrate":<Double_value>,

"errpkttxrate":<Double_value>,

"nictxstalls":<Double_value>,

"rxlacpdurate":<Double_value>,

"totnetscalerpkts":<Double_value>,

"trunkpktstransmittedrate":<Double_value>,

"jumbopktsreceivedrate":<Double_value>,

"totrxbytes":<Double_value>,

"trunkpktstransmitted":<Double_value>,

"linkreinits":<Double_value>,

"nicerrdisables":<Double_value>,

"txpktsrate":<Double_value>,

"jumbopktsreceived":<Double_value>,

"txlacpdu":<Double_value>,

"errdroppedrxpktsrate":<Double_value>,

"errnicmuted":<Double_value>,

"errdroppedtxpkts":<Double_value>,

"nicerrifoutdiscardsrate":<Double_value>,
```

```
"errifindiscardsrate":<Double_value>,

"trunkpktsreceivedrate":<Double_value>,

"nicerrifoutdiscards":<Double_value>,

"curlinkstate":<String_value>,

"rxcrcerrors":<Double_value>,

"errifindiscards":<Double_value>,

"interfacealias":<String_value>,

"totrxpkts":<Double_value>,

"totmacmoved":<Double_value>,

"rxpktsrate":<Double_value>,

"errduplexmismatch":<Double_value>,

"jumbopktstransmittedrate":<Double_value>,

"errpktrxrate":<Double_value>,

"errlinkhangs":<Double_value>,

"errdroppedrxpkts":<Double_value>,

"txlacpdurate":<Double_value>,

"nicstsstalls":<Double_value>,

"trunkpktsreceived":<Double_value>

}]}
```

mapdomain

Statistics for MAP-T Map Domain resource.

Properties

(click to see Operations)

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
name	<String>	Read-write	An integer specifying the VLAN identification number (VID). Possible values: 1 through 4094. Minimum length = 1 Maximum length = 127
clearstats	<String>	Read-write	Clear the statsistics / counters. Possible values = basic, full
maptotv6rxpktstcp	<Double>	Read-only	Total number of MAP-T IPv6 TCP Recieved packets.
mapv6rxpktstcprate	<Double>	Read-only	Rate (/s) counter for maptotv6rxpktstcp
maptotv6txpktstcp	<Double>	Read-only	Total number of MAP-T IPv6 TCP Transmitted packets.
mapv6txpktstcprate	<Double>	Read-only	Rate (/s) counter for maptotv6txpktstcp
maptotv6rxbytestcp	<Double>	Read-only	Total number of MAP-T IPv6 TCP Recieved Bytes.
mapv6rxbytestcprate	<Double>	Read-only	Rate (/s) counter for maptotv6rxbytestcp

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
maptotv6txbytestcp	<Double>	Read-only	Total number of MAP-T IPv6 TCP Transmitted Bytes.
mapv6txbytestcprate	<Double>	Read-only	Rate (/s) counter for maptotv6txbytestcp
maptotv4rxpktstcp	<Double>	Read-only	Total number of MAP-T IPv4 TCP Recieved packets.
mapv4rxpktstcprate	<Double>	Read-only	Rate (/s) counter for maptotv4rxpktstcp
maptotv4txpktstcp	<Double>	Read-only	Total number of MAP-T IPv4 TCP Transmitted packets.
mapv4txpktstcprate	<Double>	Read-only	Rate (/s) counter for maptotv4txpktstcp
maptotv4rxbytestcp	<Double>	Read-only	Total number of MAP-T IPv4 TCP Recieved Bytes.
mapv4rxbytestcprate	<Double>	Read-only	Rate (/s) counter for maptotv4rxbytestcp
maptotv4txbytestcp	<Double>	Read-only	Total number of MAP-T IPv4 TCP Transmitted Bytes.
mapv4txbytestcprate	<Double>	Read-only	Rate (/s) counter for maptotv4txbytestcp
maptotv6rxpktsudp	<Double>	Read-only	Total number of MAP-T IPv6 UDP Recieved packets.
mapv6rxpktsudprate	<Double>	Read-only	Rate (/s) counter for maptotv6rxpktsudp

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
maptotv6txpktsudp	<Double>	Read-only	Total number of MAP-T IPv6 UDP Transmitted packets.
mapv6txpktsudprate	<Double>	Read-only	Rate (/s) counter for maptotv6txpktsudp
maptotv6rxbytesudp	<Double>	Read-only	Total number of MAP-T IPv6 UDP Recieved Bytes.
mapv6rxbytesudprate	<Double>	Read-only	Rate (/s) counter for maptotv6rxbytesudp
maptotv6txbytesudp	<Double>	Read-only	Total number of MAP-T IPv6 UDP Transmitted Bytes.
mapv6txbytesudprate	<Double>	Read-only	Rate (/s) counter for maptotv6txbytesudp
maptotv4rxpktsudp	<Double>	Read-only	Total number of MAP-T IPv4 UDP Recieved packets.
mapv4rxpktsudprate	<Double>	Read-only	Rate (/s) counter for maptotv4rxpktsudp
maptotv4txpktsudp	<Double>	Read-only	Total number of MAP-T IPv4 UDP Transmitted packets.
mapv4txpktsudprate	<Double>	Read-only	Rate (/s) counter for maptotv4txpktsudp
maptotv4rxbytesudp	<Double>	Read-only	Total number of MAP-T IPv4 UDP Recieved Bytes.
mapv4rxbytesudprate	<Double>	Read-only	Rate (/s) counter for maptotv4rxbytesudp

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
maptotv4txbytesudp	<Double>	Read-only	Total number of MAP-T IPv4 UDP Transmitted Bytes.
mapv4txbytesudprate	<Double>	Read-only	Rate (/s) counter for maptotv4txbytesudp
maptotv6rxpktsicmp	<Double>	Read-only	Total number of MAP-T IPv6 ICMP Recieved packets.
mapv6rxpktsicmprate	<Double>	Read-only	Rate (/s) counter for maptotv6rxpktsicmp
maptotv6txpktsicmp	<Double>	Read-only	Total number of MAP-T IPv6 ICMP Transmitted packets.
mapv6txpktsicmprate	<Double>	Read-only	Rate (/s) counter for maptotv6txpktsicmp
maptotv6rxbytesicmp	<Double>	Read-only	Total number of MAP-T IPv6 ICMP Recieved Bytes.
mapv6rxbytesicmprate	<Double>	Read-only	Rate (/s) counter for maptotv6rxbytesicmp
maptotv6txbytesicmp	<Double>	Read-only	Total number of MAP-T IPv6 ICMP Transmitted Bytes.
mapv6txbytesicmprate	<Double>	Read-only	Rate (/s) counter for maptotv6txbytesicmp
maptotv4rxpktsicmp	<Double>	Read-only	Total number of MAP-T IPv4 ICMP Recieved packets.
mapv4rxpktsicmprate	<Double>	Read-only	Rate (/s) counter for maptotv4rxpktsicmp

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
maptotv4txpktsicmp	<Double>	Read-only	Total number of MAP-T IPv4 ICMP Transmitted packets.
mapv4txpktsicmprate	<Double>	Read-only	Rate (/s) counter for maptotv4txpktsicmp
maptotv4rxbytesicmp	<Double>	Read-only	Total number of MAP-T IPv4 ICMP Recieved Bytes.
mapv4rxbytesicmprate	<Double>	Read-only	Rate (/s) counter for maptotv4rxbytesicmp
maptotv4txbytesicmp	<Double>	Read-only	Total number of MAP-T IPv4 ICMP Transmitted Bytes.
mapv4txbytesicmprate	<Double>	Read-only	Rate (/s) counter for maptotv4txbytesicmp
maptotv6rxpkts	<Double>	Read-only	Total number of MAP-T V6 Recieved packets.
mapv6rxpktsrate	<Double>	Read-only	Rate (/s) counter for maptotv6rxpkts
maptotv6txpkts	<Double>	Read-only	Total number of MAP-T V6 Transmitted packets.
mapv6txpktsrate	<Double>	Read-only	Rate (/s) counter for maptotv6txpkts
maptotv4rxpkts	<Double>	Read-only	Total number of MAP-T V4 Recieved packets.
mapv4rxpktsrate	<Double>	Read-only	Rate (/s) counter for maptotv4rxpkts

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
maptotv4txpkts	<Double>	Read-only	Total number of MAP-T V4 Transmitted packets.
mapv4txpktsrate	<Double>	Read-only	Rate (/s) counter for maptotv4txpkts

Operations

(click to see Properties)

GET (ALL)| GET

Some options that you can use for each operations:

- **Getting warnings in response:**NITRO allows you to get warnings in an operation by specifying the "warning" query parameter as "yes". For example, to get warnings while connecting to the Citrix ADC appliance, the URL is as follows:

  http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/login?warning=yes

  If any, the warnings are displayed in the response payload with the HTTP code "209 X-NITRO-WARNING".

- **Authenticated access for individual NITRO operations:**NITRO allows you to logon to the Citrix ADC appliance to perform individual operations. You can use this option instead of creating a NITRO session (using the login object) and then using that session to perform all operations,

  To do this, you must specify the username and password in the request header of the NITRO request as follows:

  X-NITRO-USER:<username>

  X-NITRO-PASS:<password>

  Note: In such cases, make sure that the request header DOES not include the following:

  Cookie:NITRO_AUTH_TOKEN=<tokenvalue>

`*Note:`*

Mandatory parameters are marked in redand placeholder content is marked in <green>.

get (all)

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/stat/mapdomain

Query-parameters:

args

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/stat/mapdomain?
args=name:<String_value>,detail:<Boolean_value>,fullvalues:<Boolean_value>,ntimes:<Double_value>

Use this query-parameter to get mapdomain resources based on additional properties.

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "mapdomain": [ {

"name":<String_value>,

"mapv6rxpktsrate":<Double_value>,

"mapv6txbytesicmprate":<Double_value>,

"mapv6rxbytesicmprate":<Double_value>,

"maptotv4txbytesudp":<Double_value>,

"mapv4txbytestcprate":<Double_value>,
```

```
"mapv4txpktsrate":<Double_value>,

"mapv4rxbytesicmprate":<Double_value>,

"mapv4txbytesudprate":<Double_value>,

"maptotv4txpktstcp":<Double_value>,

"maptotv6txbytestcp":<Double_value>,

"mapv4rxpktsrate":<Double_value>,

"maptotv6txbytesudp":<Double_value>,

"maptotv4rxbytesicmp":<Double_value>,

"maptotv4txpktsicmp":<Double_value>,

"maptotv4rxpktsudp":<Double_value>,

"maptotv6txpkts":<Double_value>,

"maptotv4rxbytestcp":<Double_value>,

"mapv4txbytesicmprate":<Double_value>,

"maptotv6rxbytesudp":<Double_value>,

"maptotv6rxbytestcp":<Double_value>,

"mapv6rxbytestcprate":<Double_value>,

"maptotv6rxpkts":<Double_value>,

"maptotv6rxpktstcp":<Double_value>,

"maptotv4txpktsudp":<Double_value>,

"maptotv6txpktsudp":<Double_value>,

"mapv4rxbytestcprate":<Double_value>,

"mapv4rxpktsicmprate":<Double_value>,
```

```
"maptotv4rxpkts":<Double_value>,

"maptotv6rxpktsicmp":<Double_value>,

"maptotv6rxbytesicmp":<Double_value>,

"mapv6rxpktsudprate":<Double_value>,

"maptotv4txbytesicmp":<Double_value>,

"mapv4txpktsicmprate":<Double_value>,

"maptotv6txpktstcp":<Double_value>,

"mapv6txpktsrate":<Double_value>,

"mapv6rxpktsicmprate":<Double_value>,

"maptotv4txpkts":<Double_value>,

"mapv6txpktsudprate":<Double_value>,

"maptotv6txpktsicmp":<Double_value>,

"mapv6txbytestcprate":<Double_value>,

"maptotv4rxpktstcp":<Double_value>,

"mapv4rxpktstcprate":<Double_value>,

"mapv4rxpktsudprate":<Double_value>,

"maptotv4rxpktsicmp":<Double_value>,

"mapv4txpktsudprate":<Double_value>,

"maptotv6txbytesicmp":<Double_value>,

"mapv4txpktstcprate":<Double_value>,

"maptotv6rxpktsudp":<Double_value>,

"maptotv4txbytestcp":<Double_value>,
```

```
"maptotv4rxbytesudp":<Double_value>,

"mapv6txpktstcprate":<Double_value>,

"mapv6rxbytesudprate":<Double_value>,

"mapv4rxbytesudprate":<Double_value>,

"mapv6txbytesudprate":<Double_value>,

"mapv6rxpktstcprate":<Double_value>,

"mapv6txpktsicmprate":<Double_value>

}]}
```

get

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/stat/mapdomain/name_value><String>

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "mapdomain": [ {

"name":<String_value>,
```

```
"mapv6rxpktsrate":<Double_value>,

"mapv6txbytesicmprate":<Double_value>,

"mapv6rxbytesicmprate":<Double_value>,

"maptotv4txbytesudp":<Double_value>,

"mapv4txbytestcprate":<Double_value>,

"mapv4txpktsrate":<Double_value>,

"mapv4rxbytesicmprate":<Double_value>,

"mapv4txbytesudprate":<Double_value>,

"maptotv4txpktstcp":<Double_value>,

"maptotv6txbytestcp":<Double_value>,

"mapv4rxpktsrate":<Double_value>,

"maptotv6txbytesudp":<Double_value>,

"maptotv4rxbytesicmp":<Double_value>,

"maptotv4txpktsicmp":<Double_value>,

"maptotv4rxpktsudp":<Double_value>,

"maptotv6txpkts":<Double_value>,

"maptotv4rxbytestcp":<Double_value>,

"mapv4txbytesicmprate":<Double_value>,

"maptotv6rxbytesudp":<Double_value>,

"maptotv6rxbytestcp":<Double_value>,

"mapv6rxbytestcprate":<Double_value>,

"maptotv6rxpkts":<Double_value>,
```

```
"maptotv6rxpktstcp":<Double_value>,

"maptotv4txpktsudp":<Double_value>,

"maptotv6txpktsudp":<Double_value>,

"mapv4rxbytestcprate":<Double_value>,

"mapv4rxpktsicmprate":<Double_value>,

"maptotv4rxpkts":<Double_value>,

"maptotv6rxpktsicmp":<Double_value>,

"maptotv6rxbytesicmp":<Double_value>,

"mapv6rxpktsudprate":<Double_value>,

"maptotv4txbytesicmp":<Double_value>,

"mapv4txpktsicmprate":<Double_value>,

"maptotv6txpktstcp":<Double_value>,

"mapv6txpktsrate":<Double_value>,

"mapv6rxpktsicmprate":<Double_value>,

"maptotv4txpkts":<Double_value>,

"mapv6txpktsudprate":<Double_value>,

"maptotv6txpktsicmp":<Double_value>,

"mapv6txbytestcprate":<Double_value>,

"maptotv4rxpktstcp":<Double_value>,

"mapv4rxpktstcprate":<Double_value>,

"mapv4rxpktsudprate":<Double_value>,

"maptotv4rxpktsicmp":<Double_value>,
```

```
"mapv4txpktsudprate":<Double_value>,

"maptotv6txbytesicmp":<Double_value>,

"mapv4txpktstcprate":<Double_value>,

"maptotv6rxpktsudp":<Double_value>,

"maptotv4txbytestcp":<Double_value>,

"maptotv4rxbytesudp":<Double_value>,

"mapv6txpktstcprate":<Double_value>,

"mapv6rxbytesudprate":<Double_value>,

"mapv4rxbytesudprate":<Double_value>,

"mapv6txbytesudprate":<Double_value>,

"mapv6rxpktstcprate":<Double_value>,

"mapv6txpktsicmprate":<Double_value>

}]}
```

mapbmr

Statistics for MAP-T Basic Mapping rule resource.

Properties

(click to see Operations)

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
name	<String>	Read-write	Basic Mapping Rule name. Minimum length = 1

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
			Maximum length = 127
clearstats	<String>	Read-write	Clear the statsistics / counters. Possible values = basic, full
bmrtotv6rxpkts	<Double>	Read-only	Total number of MAP-T BMR V6 Recieved packets.
bmrv6rxpktsrate	<Double>	Read-only	Rate (/s) counter for bmrtotv6rxpkts
bmrtotv6txpkts	<Double>	Read-only	Total number of MAP-T BMR V6 Transmitted packets.
bmrv6txpktsrate	<Double>	Read-only	Rate (/s) counter for bmrtotv6txpkts
bmrtotv4rxpkts	<Double>	Read-only	Total number of MAP-T BMR V4 Recieved packets.
bmrv4rxpktsrate	<Double>	Read-only	Rate (/s) counter for bmrtotv4rxpkts
bmrtotv4txpkts	<Double>	Read-only	Total number of MAP-T BMR V4 Transmitted packets.
bmrv4txpktsrate	<Double>	Read-only	Rate (/s) counter for bmrtotv4txpkts
bmrtotv6rxpktstcp	<Double>	Read-only	Total number of MAP-T BMR V6 TCP Recieved packets.

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
bmrv6rxpktstcprate	<Double>	Read-only	Rate (/s) counter for bmrtotv6rxpktstcp
bmrtotv6txpktstcp	<Double>	Read-only	Total number of MAP-T BMR V6 TCP Transmitted packets.
bmrv6txpktstcprate	<Double>	Read-only	Rate (/s) counter for bmrtotv6txpktstcp
bmrtotv4rxpktstcp	<Double>	Read-only	Total number of MAP-T BMR V4 TCP Recieved packets.
bmrv4rxpktstcprate	<Double>	Read-only	Rate (/s) counter for bmrtotv4rxpktstcp
bmrtotv4txpktstcp	<Double>	Read-only	Total number of MAP-T BMR V4 TCP Transmitted packets.
bmrv4txpktstcprate	<Double>	Read-only	Rate (/s) counter for bmrtotv4txpktstcp
bmrtotv6rxpktsudp	<Double>	Read-only	Total number of MAP-T BMR V6 UDP Recieved packets.
bmrv6rxpktsudprate	<Double>	Read-only	Rate (/s) counter for bmrtotv6rxpktsudp
bmrtotv6txpktsudp	<Double>	Read-only	Total number of MAP-T BMR V6 UDP Transmitted packets.
bmrv6txpktsudprate	<Double>	Read-only	Rate (/s) counter for bmrtotv6txpktsudp

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
bmrtotv4rxpktsudp	<Double>	Read-only	Total number of MAP-T BMR V4 UDP Recieved packets.
bmrv4rxpktsudprate	<Double>	Read-only	Rate (/s) counter for bmrtotv4rxpktsudp
bmrtotv4txpktsudp	<Double>	Read-only	Total number of MAP-T BMR V4 UDP Transmitted packets.
bmrv4txpktsudprate	<Double>	Read-only	Rate (/s) counter for bmrtotv4txpktsudp
bmrtotv6rxpktsicmp	<Double>	Read-only	Total number of MAP-T BMR V6 ICMP Recieved packets.
bmrv6rxpktsicmprate	<Double>	Read-only	Rate (/s) counter for bmrtotv6rxpktsicmp
bmrtotv6txpktsicmp	<Double>	Read-only	Total number of MAP-T BMR V6 ICMP Transmitted packets.
bmrv6txpktsicmprate	<Double>	Read-only	Rate (/s) counter for bmrtotv6txpktsicmp
bmrtotv4rxpktsicmp	<Double>	Read-only	Total number of MAP-T BMR V4 ICMP Recieved packets.
bmrv4rxpktsicmprate	<Double>	Read-only	Rate (/s) counter for bmrtotv4rxpktsicmp

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
bmrtotv4txpktsicmp	<Double>	Read-only	Total number of MAP-T BMR V4 ICMP Transmitted packets.
bmrv4txpktsicmprate	<Double>	Read-only	Rate (/s) counter for bmrtotv4txpktsicmp

Operations

(click to see Properties)

GET (ALL)| GET

Some options that you can use for each operations:

- **Getting warnings in response:**NITRO allows you to get warnings in an operation by specifying the "warning" query parameter as "yes". For example, to get warnings while connecting to the Citrix ADC appliance, the URL is as follows:

  http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/login?warning=yes

  If any, the warnings are displayed in the response payload with the HTTP code "209 X-NITRO-WARNING".

- **Authenticated access for individual NITRO operations:**NITRO allows you to logon to the Citrix ADC appliance to perform individual operations. You can use this option instead of creating a NITRO session (using the login object) and then using that session to perform all operations,

  To do this, you must specify the username and password in the request header of the NITRO request as follows:

  X-NITRO-USER:<username>

  X-NITRO-PASS:<password>

  Note: In such cases, make sure that the request header DOES not include the following:

  Cookie:NITRO_AUTH_TOKEN=<tokenvalue>

`*Note:`*

Mandatory parameters are marked in redand placeholder content is marked in <green>.

get (all)

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/stat/mapbmr

Query-parameters:

args

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/stat/mapbmr?
args=name:<String_value>,detail:<Boolean_value>,fullvalues:<Boolean_value>,ntimes:<Double_value>

Use this query-parameter to get mapbmr resources based on additional properties.

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "mapbmr": [ {

"name":<String_value>,

"bmrv6rxpktsrate":<Double_value>,

"bmrv4rxpktsudprate":<Double_value>,

"bmrtotv6txpkts":<Double_value>,

"bmrv6rxpktsudprate":<Double_value>,
```

```
"bmrtotv6txpktsicmp":<Double_value>,

"bmrtotv4rxpktsudp":<Double_value>,

"bmrv6txpktsudprate":<Double_value>,

"bmrv4txpktsicmprate":<Double_value>,

"bmrtotv6rxpkts":<Double_value>,

"bmrv4txpktsrate":<Double_value>,

"bmrtotv4rxpkts":<Double_value>,

"bmrv4txpktstcprate":<Double_value>,

"bmrv4rxpktsicmprate":<Double_value>,

"bmrv4rxpktstcprate":<Double_value>,

"bmrv6txpktsicmprate":<Double_value>,

"bmrv6txpktstcprate":<Double_value>,

"bmrtotv6rxpktsicmp":<Double_value>,

"bmrtotv6txpktsudp":<Double_value>,

"bmrv6rxpktstcprate":<Double_value>,

"bmrtotv4txpktsicmp":<Double_value>,

"bmrtotv4txpktstcp":<Double_value>,

"bmrtotv4rxpktsicmp":<Double_value>,

"bmrv6txpktsrate":<Double_value>,

"bmrtotv4txpktsudp":<Double_value>,

"bmrtotv4rxpktstcp":<Double_value>,

"bmrtotv4txpkts":<Double_value>,
```

```
"bmrv4txpktsudprate":<Double_value>,

"bmrtotv6rxpktsudp":<Double_value>,

"bmrtotv6txpktstcp":<Double_value>,

"bmrtotv6rxpktstcp":<Double_value>,

"bmrv4rxpktsrate":<Double_value>,

"bmrv6rxpktsicmprate":<Double_value>

}]}
```

get

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/stat/mapbmr/name_value><String>

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "mapbmr": [ {

"name":<String_value>,

"bmrv6rxpktsrate":<Double_value>,
```

```
"bmrv4rxpktsudprate":<Double_value>,

"bmrtotv6txpkts":<Double_value>,

"bmrv6rxpktsudprate":<Double_value>,

"bmrtotv6txpktsicmp":<Double_value>,

"bmrtotv4rxpktsudp":<Double_value>,

"bmrv6txpktsudprate":<Double_value>,

"bmrv4txpktsicmprate":<Double_value>,

"bmrtotv6rxpkts":<Double_value>,

"bmrv4txpktsrate":<Double_value>,

"bmrtotv4rxpkts":<Double_value>,

"bmrv4txpktstcprate":<Double_value>,

"bmrv4rxpktsicmprate":<Double_value>,

"bmrv4rxpktstcprate":<Double_value>,

"bmrv6txpktsicmprate":<Double_value>,

"bmrv6txpktstcprate":<Double_value>,

"bmrtotv6rxpktsicmp":<Double_value>,

"bmrtotv6txpktsudp":<Double_value>,

"bmrv6rxpktstcprate":<Double_value>,

"bmrtotv4txpktsicmp":<Double_value>,

"bmrtotv4txpktstcp":<Double_value>,

"bmrtotv4rxpktsicmp":<Double_value>,

"bmrv6txpktsrate":<Double_value>,
```

```
"bmrtotv4txpktsudp":<Double_value>,

"bmrtotv4rxpktstcp":<Double_value>,

"bmrtotv4txpkts":<Double_value>,

"bmrv4txpktsudprate":<Double_value>,

"bmrtotv6rxpktsudp":<Double_value>,

"bmrtotv6txpktstcp":<Double_value>,

"bmrtotv6rxpktstcp":<Double_value>,

"bmrv4rxpktsrate":<Double_value>,

"bmrv6rxpktsicmprate":<Double_value>

}]}
```

nat64

Statistics for nat64 config resource.

Properties

(click to see Operations)

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
clearstats	<String>	Read-write	Clear the statsistics / counters. Possible values = basic, full
nat64tottcpsessions	<Double>	Read-only	Total number of TCP sessions created by NAT64.
nat64tcpsessionsrate	<Double>	Read-only	Rate (/s) counter for nat64tottcpsessions

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
nat64totudpsessions	<Double>	Read-only	Total number of UDP sessions created by NAT64.
nat64udpsessionsrate	<Double>	Read-only	Rate (/s) counter for nat64totudpsessions
nat64toticmpsessions	<Double>	Read-only	Total number of ICMP sessions created by NAT64.
nat64icmpsessionsrate	<Double>	Read-only	Rate (/s) counter for nat64toticmpsessions
nat64totsessions	<Double>	Read-only	Total number of sessions created by NAT64.
nat64sessionsrate	<Double>	Read-only	Rate (/s) counter for nat64totsessions

Operations

(click to see Properties)

GET (ALL)

Some options that you can use for each operations:

- **Getting warnings in response:**NITRO allows you to get warnings in an operation by specifying the "warning" query parameter as "yes". For example, to get warnings while connecting to the Citrix ADC appliance, the URL is as follows:

  http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/login?warning=yes

  If any, the warnings are displayed in the response payload with the HTTP code "209 X-NITRO-WARNING".
- **Authenticated access for individual NITRO operations:**NITRO allows you to logon to the Citrix ADC appliance to perform individual operations. You can use this option instead of

creating a NITRO session (using the login object) and then using that session to perform all operations,

To do this, you must specify the username and password in the request header of the NITRO request as follows:

X-NITRO-USER:<username>

X-NITRO-PASS:<password>

Note: In such cases, make sure that the request header DOES not include the following:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue>

`*Note:`*

Mandatory parameters are marked in redand placeholder content is marked in <green>.

get (all)

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/stat/nat64

Query-parameters:

args

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/stat/nat64?
args=detail:<Boolean_value>,fullvalues:<Boolean_value>,ntimes:<Double_value>,logfile:<String_value

Use this query-parameter to get nat64 resources based on additional properties.

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "nat64": [ {
"nat64icmpsessionsrate":<Double_value>,
"nat64toticmpsessions":<Double_value>,
"nat64sessionsrate":<Double_value>,
"nat64totudpsessions":<Double_value>,
"nat64udpsessionsrate":<Double_value>,
"nat64tottcpsessions":<Double_value>,
"nat64totsessions":<Double_value>,
"nat64tcpsessionsrate":<Double_value>
}]}
```

rnat

Statistics for RNAT configured route resource.

Properties

(click to see Operations)

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
clearstats	<String>	Read-write	Clear the statsistics / counters. Possible values = basic, full
rnattotrxbytes	<Double>	Read-only	Bytes received during RNAT sessions.

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
rnatrxbytesrate	<Double>	Read-only	Rate (/s) counter for rnattotrxbytes
rnattottxbytes	<Double>	Read-only	Bytes sent during RNAT sessions.
rnattxbytesrate	<Double>	Read-only	Rate (/s) counter for rnattottxbytes
rnattotrxpkts	<Double>	Read-only	Packets received during RNAT sessions.
rnatrxpktsrate	<Double>	Read-only	Rate (/s) counter for rnattotrxpkts
rnattottxpkts	<Double>	Read-only	Packets sent during RNAT sessions.
rnattxpktsrate	<Double>	Read-only	Rate (/s) counter for rnattottxpkts
rnattottxsyn	<Double>	Read-only	Requests for connections sent during RNAT sessions.
rnattxsynrate	<Double>	Read-only	Rate (/s) counter for rnattottxsyn
rnatcursessions	<Double>	Read-only	Currently active RNAT sessions.

Operations

(click to see Properties)

GET (ALL)

Some options that you can use for each operations:

- **Getting warnings in response:**NITRO allows you to get warnings in an operation by specifying the "warning" query parameter as "yes". For example, to get warnings while connecting to the Citrix ADC appliance, the URL is as follows:

  http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/login?warning=yes

  If any, the warnings are displayed in the response payload with the HTTP code "209 X-NITRO-WARNING".

- **Authenticated access for individual NITRO operations:**NITRO allows you to logon to the Citrix ADC appliance to perform individual operations. You can use this option instead of creating a NITRO session (using the login object) and then using that session to perform all operations,

  To do this, you must specify the username and password in the request header of the NITRO request as follows:

  X-NITRO-USER:<username>

  X-NITRO-PASS:<password>

  Note: In such cases, make sure that the request header DOES not include the following:

  Cookie:NITRO_AUTH_TOKEN=<tokenvalue>

`*Note:`*

Mandatory parameters are marked in redand placeholder content is marked in <green>.

get (all)

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/stat/rnat

Query-parameters:

args

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/stat/rnat?
args=detail:<Boolean_value>,fullvalues:<Boolean_value>,ntimes:<Double_value>,logfile:<String_value

Use this query-parameter to get rnat resources based on additional properties.

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "rnat": [ {

"rnattottxpkts":<Double_value>,

"rnatrxbytesrate":<Double_value>,

"rnattxsynrate":<Double_value>,

"rnattxpktsrate":<Double_value>,

"rnattottxsyn":<Double_value>,

"rnattxbytesrate":<Double_value>,

"rnatrxpktsrate":<Double_value>,

"rnatcursessions":<Double_value>,

"rnattotrxpkts":<Double_value>,

"rnattotrxbytes":<Double_value>,

"rnattottxbytes":<Double_value>

}]}
```

rnat6

Statistics for IPv6 RNAT configured route resource.

Properties

(click to see Operations)

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
clearstats	<String>	Read-write	Clear the statsistics / counters. Possible values = basic, full
rnat6totrxbytes	<Double>	Read-only	Bytes received during RNAT6 sessions.
rnat6rxbytesrate	<Double>	Read-only	Rate (/s) counter for rnat6totrxbytes
rnat6tottxbytes	<Double>	Read-only	Bytes sent during RNAT6 sessions.
rnat6txbytesrate	<Double>	Read-only	Rate (/s) counter for rnat6tottxbytes
rnat6totrxpkts	<Double>	Read-only	Packets received during RNAT6 sessions.
rnat6rxpktsrate	<Double>	Read-only	Rate (/s) counter for rnat6totrxpkts
rnat6tottxpkts	<Double>	Read-only	Packets sent during RNAT6 sessions.
rnat6txpktsrate	<Double>	Read-only	Rate (/s) counter for rnat6tottxpkts
rnat6tottxsyn	<Double>	Read-only	Requests for connections sent

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
			during RNAT6 sessions.
rnat6txsynrate	<Double>	Read-only	Rate (/s) counter for rnat6tottxsyn
rnat6cursessions	<Double>	Read-only	Currently active RNAT6 sessions.

Operations

(click to see Properties)

GET (ALL)

Some options that you can use for each operations:

- **Getting warnings in response:**NITRO allows you to get warnings in an operation by specifying the "warning" query parameter as "yes". For example, to get warnings while connecting to the Citrix ADC appliance, the URL is as follows:

  http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/login?warning=yes

  If any, the warnings are displayed in the response payload with the HTTP code "209 X-NITRO-WARNING".

- **Authenticated access for individual NITRO operations:**NITRO allows you to logon to the Citrix ADC appliance to perform individual operations. You can use this option instead of creating a NITRO session (using the login object) and then using that session to perform all operations,

  To do this, you must specify the username and password in the request header of the NITRO request as follows:

  X-NITRO-USER:<username>

  X-NITRO-PASS:<password>

  Note: In such cases, make sure that the request header DOES not include the following:

  Cookie:NITRO_AUTH_TOKEN=<tokenvalue>

`*Note:` *

Mandatory parameters are marked in redand placeholder content is marked in <green>.

get (all)

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/stat/rnat6

Query-parameters:

args

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/stat/rnat6?
args=detail:<Boolean_value>,fullvalues:<Boolean_value>,ntimes:<Double_value>,logfile:<String_value

Use this query-parameter to get rnat6 resources based on additional properties.

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "rnat6": [ {

"rnat6cursessions":<Double_value>,

"rnat6txpktsrate":<Double_value>,

"rnat6tottxsyn":<Double_value>,

"rnat6tottxbytes":<Double_value>,

"rnat6txsynrate":<Double_value>,
```

```
"rnat6rxpktsrate":<Double_value>,

"rnat6rxbytesrate":<Double_value>,

"rnat6txbytesrate":<Double_value>,

"rnat6totrxbytes":<Double_value>,

"rnat6totrxpkts":<Double_value>,

"rnat6tottxpkts":<Double_value>

}]}
```

rnatip

Statistics for RNAT ipaddress resource.

Properties

(click to see Operations)

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
Rnatip	<String>	Read-write	Specifies the NAT IP address of the configured RNAT entry for which you want to see the statistics. If you do not specify an IP address, this displays the statistics for all the configured RNAT entries. Minimum length = 1
clearstats	<String>	Read-write	Clear the statsistics / counters. Possible values = basic, full

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
iptd	<Double>	Read-only	Traffic domain for ipaddr.
iprnattotrxbytes	<Double>	Read-only	Bytes received on this IP address during RNAT sessions.
iprnatrxbytesrate	<Double>	Read-only	Rate (/s) counter for iprnattotrxbytes
iprnattottxbytes	<Double>	Read-only	Bytes sent from this IP address during RNAT sessions.
iprnattxbytesrate	<Double>	Read-only	Rate (/s) counter for iprnattottxbytes
iprnattotrxpkts	<Double>	Read-only	Packets received on this IP address during RNAT sessions.
iprnatrxpktsrate	<Double>	Read-only	Rate (/s) counter for iprnattotrxpkts
iprnattottxpkts	<Double>	Read-only	Packets sent from this IP address during RNAT sessions.
iprnattxpktsrate	<Double>	Read-only	Rate (/s) counter for iprnattottxpkts
iprnattottxsyn	<Double>	Read-only	Requests for connections sent from this IP address during RNAT sessions.

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
iprnattxsynrate	<Double>	Read-only	Rate (/s) counter for iprnattottxsyn
iprnatcursessions	<Double>	Read-only	Currently active RNAT sessions started from this IP address.

Operations

(click to see Properties)

GET (ALL)| GET

Some options that you can use for each operations:

- **Getting warnings in response:**NITRO allows you to get warnings in an operation by specifying the "warning" query parameter as "yes". For example, to get warnings while connecting to the Citrix ADC appliance, the URL is as follows:

  http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/login?warning=yes

  If any, the warnings are displayed in the response payload with the HTTP code "209 X-NITRO-WARNING".

- **Authenticated access for individual NITRO operations:**NITRO allows you to logon to the Citrix ADC appliance to perform individual operations. You can use this option instead of creating a NITRO session (using the login object) and then using that session to perform all operations,

  To do this, you must specify the username and password in the request header of the NITRO request as follows:

  X-NITRO-USER:<username>

  X-NITRO-PASS:<password>

  Note: In such cases, make sure that the request header DOES not include the following:

  Cookie:NITRO_AUTH_TOKEN=<tokenvalue>

`*Note:`*

Mandatory parameters are marked in redand placeholder content is marked in <green>.

get (all)

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/stat/rnatip

Query-parameters:

args

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/stat/rnatip?
args=Rnatip:<String_value>,detail:<Boolean_value>,fullvalues:<Boolean_value>,ntimes:<Double_value

Use this query-parameter to get rnatip resources based on additional properties.

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "rnatip": [ {

"Rnatip":<String_value>,

"iprnatrxpktsrate":<Double_value>,

"iprnattxpktsrate":<Double_value>,

"iprnattottxpkts":<Double_value>,

"iptd":<Double_value>,
```

```
"iprnattottxbytes":<Double_value>,

"iprnatcursessions":<Double_value>,

"iprnatrxbytesrate":<Double_value>,

"iprnattotrxbytes":<Double_value>,

"iprnattxsynrate":<Double_value>,

"iprnattxbytesrate":<Double_value>,

"iprnattotrxpkts":<Double_value>,

"iprnattottxsyn":<Double_value>

}]}
```

get

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/stat/rnatip/Rnatip_value><String>

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "rnatip": [ {
```

```
"Rnatip":<String_value>,

"iprnatrxpktsrate":<Double_value>,

"iprnattxpktsrate":<Double_value>,

"iprnattottxpkts":<Double_value>,

"iptd":<Double_value>,

"iprnattottxbytes":<Double_value>,

"iprnatcursessions":<Double_value>,

"iprnatrxbytesrate":<Double_value>,

"iprnattotrxbytes":<Double_value>,

"iprnattxsynrate":<Double_value>,

"iprnattxbytesrate":<Double_value>,

"iprnattotrxpkts":<Double_value>,

"iprnattottxsyn":<Double_value>

}]}
```

tunnelip

Statistics for TUNNEL Remote ipaddress resource.

Properties

(click to see Operations)

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
tunnelip	<String>	Read-write	remote IP address of the configured tunnel.

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
			Minimum length = 1
clearstats	<String>	Read-write	Clear the statsistics / counters. Possible values = basic, full
tnltotrxpkts	<Double>	Read-only	Total number of packets received on the tunnel.
tnlrxpktsrate	<Double>	Read-only	Rate (/s) counter for tnltotrxpkts
tnltottxpkts	<Double>	Read-only	Total number of packets transmitted on the tunnel.
tnltxpktsrate	<Double>	Read-only	Rate (/s) counter for tnltottxpkts
tnltotrxbytes	<Double>	Read-only	Total number of bytes received on the tunnel.
tnlrxbytesrate	<Double>	Read-only	Rate (/s) counter for tnltotrxbytes
tnltottxbytes	<Double>	Read-only	Total number of bytes transmitted on the tunnel.
tnltxbytesrate	<Double>	Read-only	Rate (/s) counter for tnltottxbytes

Operations

(click to see Properties)

GET (ALL)| GET

Some options that you can use for each operations:

- **Getting warnings in response:**NITRO allows you to get warnings in an operation by specifying the "warning" query parameter as "yes". For example, to get warnings while connecting to the Citrix ADC appliance, the URL is as follows:

  http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/login?warning=yes

  If any, the warnings are displayed in the response payload with the HTTP code "209 X-NITRO-WARNING".

- **Authenticated access for individual NITRO operations:**NITRO allows you to logon to the Citrix ADC appliance to perform individual operations. You can use this option instead of creating a NITRO session (using the login object) and then using that session to perform all operations,

  To do this, you must specify the username and password in the request header of the NITRO request as follows:

  X-NITRO-USER:<username>

  X-NITRO-PASS:<password>

  Note: In such cases, make sure that the request header DOES not include the following:

  Cookie:NITRO_AUTH_TOKEN=<tokenvalue>

`*Note:`*

Mandatory parameters are marked in redand placeholder content is marked in <green>.

get (all)

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/stat/tunnelip

Query-parameters:

args

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/stat/tunnelip?
args=tunnelip:<String_value>,detail:<Boolean_value>,fullvalues:<Boolean_value>,ntimes:<Double_valu

Use this query-parameter to get tunnelip resources based on additional properties.

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "tunnelip": [ {

"tunnelip":<String_value>,

"tnltxpktsrate":<Double_value>,

"tnltxbytesrate":<Double_value>,

"tnlrxpktsrate":<Double_value>,

"tnltotrxpkts":<Double_value>,

"tnltottxbytes":<Double_value>,

"tnltottxpkts":<Double_value>,

"tnlrxbytesrate":<Double_value>,

"tnltotrxbytes":<Double_value>

}]}
```

get

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/stat/tunnelip/tunnelip_value><String>

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "tunnelip": [ {

"tunnelip":<String_value>,

"tnltxpktsrate":<Double_value>,

"tnltxbytesrate":<Double_value>,

"tnlrxpktsrate":<Double_value>,

"tnltotrxpkts":<Double_value>,

"tnltottxbytes":<Double_value>,

"tnltottxpkts":<Double_value>,

"tnlrxbytesrate":<Double_value>,

"tnltotrxbytes":<Double_value>

}]}
```

tunnelip6

Statistics for TUNNEL Remote ip6address resource.

Properties

(click to see Operations)

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
tunnelip6	<String>	Read-write	remote IPv6 address of the configured tunnel. Minimum length = 1
clearstats	<String>	Read-write	Clear the statsistics / counters. Possible values = basic, full
tnltotrxpkts	<Double>	Read-only	Total number of packets received on the tunnel.
tnlrxpktsrate	<Double>	Read-only	Rate (/s) counter for tnltotrxpkts
tnltottxpkts	<Double>	Read-only	Total number of packets transmitted on the tunnel.
tnltxpktsrate	<Double>	Read-only	Rate (/s) counter for tnltottxpkts
tnltotrxbytes	<Double>	Read-only	Total number of bytes received on the tunnel.
tnlrxbytesrate	<Double>	Read-only	Rate (/s) counter for tnltotrxbytes
tnltottxbytes	<Double>	Read-only	Total number of bytes transmitted on the tunnel.
tnltxbytesrate	<Double>	Read-only	Rate (/s) counter for tnltottxbytes

Operations

(click to see Properties)

GET (ALL)| GET

Some options that you can use for each operations:

- **Getting warnings in response:**NITRO allows you to get warnings in an operation by specifying the "warning" query parameter as "yes". For example, to get warnings while connecting to the Citrix ADC appliance, the URL is as follows:

  http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/login?warning=yes

  If any, the warnings are displayed in the response payload with the HTTP code "209 X-NITRO-WARNING".

- **Authenticated access for individual NITRO operations:**NITRO allows you to logon to the Citrix ADC appliance to perform individual operations. You can use this option instead of creating a NITRO session (using the login object) and then using that session to perform all operations,

  To do this, you must specify the username and password in the request header of the NITRO request as follows:

  X-NITRO-USER:<username>

  X-NITRO-PASS:<password>

  Note: In such cases, make sure that the request header DOES not include the following:

  Cookie:NITRO_AUTH_TOKEN=<tokenvalue>

`*Note:`*

Mandatory parameters are marked in redand placeholder content is marked in <green>.

get (all)

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/stat/tunnelip6

Query-parameters:

args

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/stat/tunnelip6?
args=tunnelip6:<String_value>,detail:<Boolean_value>,fullvalues:<Boolean_value>,ntimes:<Double_va

Use this query-parameter to get tunnelip6 resources based on additional properties.

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "tunnelip6": [ {

"tunnelip6":<String_value>,

"tnltxbytesrate":<Double_value>,

"tnltotrxpkts":<Double_value>,

"tnltottxbytes":<Double_value>,

"tnlrxbytesrate":<Double_value>,

"tnltotrxbytes":<Double_value>,

"tnltxpktsrate":<Double_value>,

"tnlrxpktsrate":<Double_value>,

"tnltottxpkts":<Double_value>

}]}
```

get

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/stat/tunnelip6/tunnelip6_value><String>

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "tunnelip6": [ {

"tunnelip6":<String_value>,

"tnltxbytesrate":<Double_value>,

"tnltotrxpkts":<Double_value>,

"tnltottxbytes":<Double_value>,

"tnlrxbytesrate":<Double_value>,

"tnltotrxbytes":<Double_value>,

"tnltxpktsrate":<Double_value>,

"tnlrxpktsrate":<Double_value>,

"tnltottxpkts":<Double_value>
```

```
}]}
```

vlan

Statistics for "VLAN" resource.

Properties

(click to see Operations)

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
id	<Double>	Read-write	An integer specifying the VLAN identification number (VID). Possible values: 1 through 4094. Minimum value = 1 Maximum value = 4094
clearstats	<String>	Read-write	Clear the statsistics / counters. Possible values = basic, full
vlantotrxpkts	<Double>	Read-only	Packets received on the VLAN.
vlanrxpktsrate	<Double>	Read-only	Rate (/s) counter for vlantotrxpkts
vlantotrxbytes	<Double>	Read-only	Bytes of data received on the VLAN.
vlanrxbytesrate	<Double>	Read-only	Rate (/s) counter for vlantotrxbytes

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
vlantottxpkts	<Double>	Read-only	Packets transmitted on the VLAN.
vlantxpktsrate	<Double>	Read-only	Rate (/s) counter for vlantottxpkts
vlantottxbytes	<Double>	Read-only	Bytes of data transmitted on the VLAN.
vlantxbytesrate	<Double>	Read-only	Rate (/s) counter for vlantottxbytes
vlantotdroppedpkts	<Double>	Read-only	Inbound packets dropped by the VLAN upon reception.
vlantotbroadcastpkts	<Double>	Read-only	Broadcast packets sent and received on the VLAN.

Operations

(click to see Properties)

GET (ALL)| GET

Some options that you can use for each operations:

- **Getting warnings in response:**NITRO allows you to get warnings in an operation by specifying the "warning" query parameter as "yes". For example, to get warnings while connecting to the Citrix ADC appliance, the URL is as follows:

  http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/login?warning=yes

  If any, the warnings are displayed in the response payload with the HTTP code "209 X-NITRO-WARNING".
- **Authenticated access for individual NITRO operations:**NITRO allows you to logon to the Citrix ADC appliance to perform individual operations. You can use this option instead of

creating a NITRO session (using the login object) and then using that session to perform all operations,

To do this, you must specify the username and password in the request header of the NITRO request as follows:

X-NITRO-USER:<username>

X-NITRO-PASS:<password>

Note: In such cases, make sure that the request header DOES not include the following:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue>

`*Note:`*

Mandatory parameters are marked in redand placeholder content is marked in <green>.

get (all)

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/stat/vlan

Query-parameters:

args

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/stat/vlan?
args=id:<Double_value>,detail:<Boolean_value>,fullvalues:<Boolean_value>,ntimes:<Double_value>,lo

Use this query-parameter to get vlan resources based on additional properties.

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "vlan": [ {
"id":<Double_value>,
"vlanrxbytesrate":<Double_value>,
"vlantottxpkts":<Double_value>,
"vlantotdroppedpkts":<Double_value>,
"vlanrxpktsrate":<Double_value>,
"vlantotrxpkts":<Double_value>,
"vlantotbroadcastpkts":<Double_value>,
"vlantxbytesrate":<Double_value>,
"vlantotrxbytes":<Double_value>,
"vlantxpktsrate":<Double_value>,
"vlantottxbytes":<Double_value>
}]}
```

get

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/stat/vlan/id_value><Double>

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "vlan": [ {

"id":<Double_value>,

"vlanrxbytesrate":<Double_value>,

"vlantottxpkts":<Double_value>,

"vlantotdroppedpkts":<Double_value>,

"vlanrxpktsrate":<Double_value>,

"vlantotrxpkts":<Double_value>,

"vlantotbroadcastpkts":<Double_value>,

"vlantxbytesrate":<Double_value>,

"vlantotrxbytes":<Double_value>,

"vlantxpktsrate":<Double_value>,

"vlantottxbytes":<Double_value>

}]}
```

vxlan

Statistics for "VXLAN" resource.

Properties

(click to see Operations)

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
id	<Double>	Read-write	An integer specifying the VXLAN identification number (VNID). Minimum value = 1 Maximum value = 16777215
clearstats	<String>	Read-write	Clear the statsistics / counters. Possible values = basic, full
vxlantotrxpkts	<Double>	Read-only	Packets received on the VXLAN.
vxlanrxpktsrate	<Double>	Read-only	Rate (/s) counter for vxlantotrxpkts
vxlantotrxbytes	<Double>	Read-only	Bytes of data received on the VXLAN.
vxlanrxbytesrate	<Double>	Read-only	Rate (/s) counter for vxlantotrxbytes
vxlantottxpkts	<Double>	Read-only	Packets transmitted on the VXLAN.
vxlantxpktsrate	<Double>	Read-only	Rate (/s) counter for vxlantottxpkts
vxlantottxbytes	<Double>	Read-only	Bytes of data transmitted on the VXLAN.
vxlantxbytesrate	<Double>	Read-only	Rate (/s) counter for vxlantottxbytes

Operations

(click to see Properties)

GET (ALL)| GET

Some options that you can use for each operations:

- **Getting warnings in response:**NITRO allows you to get warnings in an operation by specifying the "warning" query parameter as "yes". For example, to get warnings while connecting to the Citrix ADC appliance, the URL is as follows:

  http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/login?warning=yes

  If any, the warnings are displayed in the response payload with the HTTP code "209 X-NITRO-WARNING".

- **Authenticated access for individual NITRO operations:**NITRO allows you to logon to the Citrix ADC appliance to perform individual operations. You can use this option instead of creating a NITRO session (using the login object) and then using that session to perform all operations,

  To do this, you must specify the username and password in the request header of the NITRO request as follows:

  X-NITRO-USER:<username>

  X-NITRO-PASS:<password>

  Note: In such cases, make sure that the request header DOES not include the following:

  Cookie:NITRO_AUTH_TOKEN=<tokenvalue>

`*Note:`*

Mandatory parameters are marked in redand placeholder content is marked in <green>.

get (all)

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/stat/vxlan

Query-parameters:

args

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/stat/vxlan?
args=id:<Double_value>,detail:<Boolean_value>,fullvalues:<Boolean_value>,ntimes:<Double_value>,lo

Use this query-parameter to get vxlan resources based on additional properties.

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "vxlan": [ {

"id":<Double_value>,

"vxlanrxbytesrate":<Double_value>,

"vxlantottxpkts":<Double_value>,

"vxlanrxpktsrate":<Double_value>,

"vxlantotrxbytes":<Double_value>,

"vxlantxbytesrate":<Double_value>,

"vxlantxpktsrate":<Double_value>,

"vxlantottxbytes":<Double_value>,

"vxlantotrxpkts":<Double_value>

}]}
```

get

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/stat/vxlan/id_value><Double>

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "vxlan": [ {

"id":<Double_value>,

"vxlanrxbytesrate":<Double_value>,

"vxlantottxpkts":<Double_value>,

"vxlanrxpktsrate":<Double_value>,

"vxlantotrxbytes":<Double_value>,

"vxlantxbytesrate":<Double_value>,

"vxlantxpktsrate":<Double_value>,

"vxlantottxbytes":<Double_value>,

"vxlantotrxpkts":<Double_value>
```

```
}]}
```

ns

Statistics for ns.

Properties

(click to see Operations)

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
clearstats	<String>	Read-write	Clear the statsistics / counters. Possible values = basic, full
rescpuusagepcnt	<Double>	Read-only	Average CPU utilization percentage. Not applicable for a single-CPU system.
cpuusagepcnt	<Double>	Read-only	CPU utilization percentage.
cachemaxmemorykb	<Double>	Read-only	Largest amount of memory the Citrix ADC can dedicate to caching, up to 50% of available memory. A 0 value disables caching, but the caching module continues to run.
delcmpratio	<Double>	Read-only	Ratio of compressible data received to compressed data

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
			transmitted.If this ratio is one (uncmp:1.0) that means compression is disabled or we are not able to compress even a single compressible packet.
rescpuusage	<Double>	Read-only	Shows average CPU utilization percentage if more than 1 CPU is present.
cpuusage	<Double>	Read-only	CPU utilization percentage.
resmemusage	<Double>	Read-only	Percentage of memory utilization on Citrix ADC.
comptotaldatacompressionratio	<Double>	Read-only	Ratio of total HTTP data received to total HTTP data transmitted.
compratio	<Double>	Read-only	Ratio of the compressible data received from the server to the compressed data sent to the client.
cacheutilizedmemorykb	<Double>	Read-only	Amount of memory the integrated cache is currently using.
cachemaxmemoryactivekb	<Double>	Read-only	Currently active value of maximum memory

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
cache64maxmemorykb	<Double>	Read-only	Largest amount of memory the Citrix ADC can dedicate to caching, up to 50% of available memory. A 0 value disables caching, but the caching module continues to run.
cachepercentoriginbandwidthsaved	<Double>	Read-only	Percentage of origin bandwidth saved, expressed as number of bytes served from the integrated cache divided by all bytes served. The assumption is that all compression is done in the Citrix ADC.
cachetotmisses	<Double>	Read-only	Intercepted HTTP requests requiring fetches from origin server.
cachemissesrate	<Double>	Read-only	Rate (/s) counter for cachetotmisses
cachetothits	<Double>	Read-only	Responses served from the integrated cache. These responses match a policy with a CACHE action.
cachehitsrate	<Double>	Read-only	Rate (/s) counter for cachetothits

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
sslnumcardsup	<Double>	Read-only	Number of SSL cards that are UP. If the number of cards UP is lower than a threshold, a failover is initiated.
memusagepcnt	<Double>	Read-only	Percentage of memory utilization on Citrix ADC.
memuseinmb	<Double>	Read-only	Main memory currently in use, in megabytes.
mgmtcpuusagepcnt	<Double>	Read-only	Average Management CPU utilization percentage.
pktcpuusagepcnt	<Double>	Read-only	Average CPU utilization percentage for all packet engines excluding management PE.
starttime	<String>	Read-only	Time when the Citrix ADC was last started.
transtime	<String>	Read-only	Time when the last master state transition occurred. You can use this statistic for debugging.
hacurstate	<String>	Read-only	State of the HA node, based on its health, in a high availability setup.

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
			Possible values are: UP - Indicates that the node is accessible and can function as either a primary or secondary node. DISABLED - Indicates that the high availability status of the node has been manually disabled. Synchronization and propagation cannot take place between the peer nodes. INIT - Indicates that the node is in the process of becoming part of the high availability configuration. PARTIALFAIL - Indicates that one of the high availability monitored interfaces has failed because of a card or link failure. This state triggers a failover. COMPLETEFAIL - Indicates that all the interfaces of the node are unusable, because the interfaces on which high availability monitoring is enabled are not connected or are manually disabled.

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
			This state triggers a failover. DUMB - Indicates that the node is in listening mode. It does not participate in high availability transitions or transfer configuration from the peer node. This is a configured value, not a statistic. PARTIALFAILSSL - Indicates that the SSL card has failed. This state triggers a failover. ROUTEMONITORFAIL - Indicates that the route monitor has failed. This state triggers a failover.
hacurmasterstate	<String>	Read-only	Indicates the high availability state of the node. Possible values are: STAYSECONDARY - Indicates that the selected node remains the secondary node in a high availability setup. In this case a forced failover does not change the state but, instead, returns an appropriate error message. This is a configured value and not a statistic.

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
			PRIMARY - Indicates that the selected node is the primary node in a high availability setup. SECONDARY - Indicates that the selected node is the secondary node in a high availability setup. CLAIMING - Indicates that the secondary node is in the process of taking over as the primary node. This is the intermediate state in the transition of the secondary node to primary status. FORCE CHANGE - Indicates that the secondary node is forcibly changing its status to primary due to a forced failover issued on the secondary node.
sslcards	<Double>	Read-only	Number of SSL crypto cards present on the Citrix ADC.
disk0perusage	<Double>	Read-only	Used space in /flash partition of the disk, as a percentage. This is a critical counter. You can configure /flash Used (%) by using the Set snmp alarm DISK-

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
			USAGE-HIGH command.
disk1perusage	<Double>	Read-only	Used space in /var partition of the disk, as a percentage. This is a critical counter. You can configure /var Used (%) by using the Set snmp alarm DISK-USAGE-HIGH command.
disk0avail	<Double>	Read-only	Available space in /flash partition of the hard disk.
disk1avail	<Double>	Read-only	Available space in /var partition of the hard disk.
totrxmbits	<Double>	Read-only	Number of megabytes received by the Citrix ADC.
rxmbitsrate	<Double>	Read-only	Rate (/s) counter for totrxmbits
tottxmbits	<Double>	Read-only	Number of megabytes transmitted by the Citrix ADC.
txmbitsrate	<Double>	Read-only	Rate (/s) counter for tottxmbits
tcpcurclientconn	<Double>	Read-only	Client connections, including connections in the Opening,

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
			Established, and Closing state.
tcpcurclientconnestablished	<Double>	Read-only	Current client connections in the Established state, which indicates that data transfer can occur between the Citrix ADC and the client.
tcpcurserverconn	<Double>	Read-only	Server connections, including connections in the Opening, Established, and Closing state.
tcpcurserverconnestablished	<Double>	Read-only	Current server connections in the Established state, which indicates that data transfer can occur between the Citrix ADC and the server.
httptotrequests	<Double>	Read-only	Total number of HTTP requests received.
httprequestsrate	<Double>	Read-only	Rate (/s) counter for httptotrequests
httptotresponses	<Double>	Read-only	Total number of HTTP responses sent.
httpresponsesrate	<Double>	Read-only	Rate (/s) counter for httptotresponses

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
httptotrxrequestbytes	<Double>	Read-only	Total number of bytes of HTTP request data received.
httprxrequestbytesrate	<Double>	Read-only	Rate (/s) counter for httptotrxrequestbytes
httptotrxresponsebytes	<Double>	Read-only	Total number of bytes of HTTP response data received.
httprxresponsebytesrate	<Double>	Read-only	Rate (/s) counter for httptotrxresponsebytes
ssltottransactions	<Double>	Read-only	Number of SSL transactions on the Citrix ADC
ssltransactionsrate	<Double>	Read-only	Rate (/s) counter for ssltottransactions
ssltotsessionhits	<Double>	Read-only	Number of SSL session reuse hits on the Citrix ADC.
sslsessionhitsrate	<Double>	Read-only	Rate (/s) counter for ssltotsessionhits
appfirewallrequests	<Double>	Read-only	HTTP/HTTPS requests sent to your protected web servers via the Application Firewall.
appfirewallrequestsrate	<Double>	Read-only	Rate (/s) counter for appfirewallrequests

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
appfirewallresponses	<Double>	Read-only	HTTP/HTTPS responses sent by your protected web servers via the Application Firewall.
appfirewallresponses rate	<Double>	Read-only	Rate (/s) counter for appfirewallresponses
appfirewallaborts	<Double>	Read-only	Incomplete HTTP/ HTTPS requests aborted by the client before the Application Firewall could finish processing them.
appfirewallabortsrate	<Double>	Read-only	Rate (/s) counter for appfirewallaborts
appfirewallredirects	<Double>	Read-only	HTTP/HTTPS requests redirected by the Application Firewall to a different Web page or web server. (HTTP 302)
appfirewallredirectsr ate	<Double>	Read-only	Rate (/s) counter for appfirewallredirects
misccounter0	<Double>	Read-only	Miscellaneous Counter 0.
misccounter1	<Double>	Read-only	Miscellaneous Counter 1.
numcpus	<Double>	Read-only	The number of CPUs on the Citrix ADC.

Operations

(click to see Properties)

GET (ALL)

Some options that you can use for each operations:

- **Getting warnings in response:**NITRO allows you to get warnings in an operation by specifying the "warning" query parameter as "yes". For example, to get warnings while connecting to the Citrix ADC appliance, the URL is as follows:

  http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/login?warning=yes

  If any, the warnings are displayed in the response payload with the HTTP code "209 X-NITRO-WARNING".

- **Authenticated access for individual NITRO operations:**NITRO allows you to logon to the Citrix ADC appliance to perform individual operations. You can use this option instead of creating a NITRO session (using the login object) and then using that session to perform all operations,

  To do this, you must specify the username and password in the request header of the NITRO request as follows:

  X-NITRO-USER:<username>

  X-NITRO-PASS:<password>

  Note: In such cases, make sure that the request header DOES not include the following:

  Cookie:NITRO_AUTH_TOKEN=<tokenvalue>

`*Note:` *

Mandatory parameters are marked in redand placeholder content is marked in <green>.

get (all)

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/stat/ns

Query-parameters:

args

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/stat/ns?
args=detail:<Boolean_value>,fullvalues:<Boolean_value>,ntimes:<Double_value>,logfile:<String_value

Use this query-parameter to get ns resources based on additional properties.

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "ns": [ {

"appfirewallredirectsrate":<Double_value>,

"ssltottransactions":<Double_value>,

"sslnumcardsup":<Double_value>,

"hacurmasterstate":<String_value>,

"httprxrequestbytesrate":<Double_value>,

"httptotrxresponsebytes":<Double_value>,

"disk1perusage":<Double_value>,

"rescpuusagepcnt":<Double_value>,

"appfirewallresponsesrate":<Double_value>,

"delcmpratio":<Double_value>,
```

```
"appfirewallresponses":<Double_value>,

"comptotaldatacompressionratio":<Double_value>,

"httprxresponsebytesrate":<Double_value>,

"disk0perusage":<Double_value>,

"appfirewallabortsrate":<Double_value>,

"appfirewallrequestsrate":<Double_value>,

"ssltotsessionhits":<Double_value>,

"appfirewallredirects":<Double_value>,

"mgmtcpuusagepcnt":<Double_value>,

"cpuusage":<Double_value>,

"httptotresponses":<Double_value>,

"compratio":<Double_value>,

"memuseinmb":<Double_value>,

"sslsessionhitsrate":<Double_value>,

"resmemusage":<Double_value>,

"cache64maxmemorykb":<Double_value>,

"starttime":<String_value>,

"cachepercentoriginbandwidthsaved":<Double_value>,

"misccounter0":<Double_value>,

"memusagepcnt":<Double_value>,

"cachehitsrate":<Double_value>,

"transtime":<String_value>,
```

```
"cacheutilizedmemorykb":<Double_value>,

"appfirewallaborts":<Double_value>,

"disk1avail":<Double_value>,

"tcpcurserverconnestablished":<Double_value>,

"tcpcurclientconn":<Double_value>,

"tcpcurserverconn":<Double_value>,

"rescpuusage":<Double_value>,

"appfirewallrequests":<Double_value>,

"totrxmbits":<Double_value>,

"tottxmbits":<Double_value>,

"cachemissesrate":<Double_value>,

"cachemaxmemorykb":<Double_value>,

"httptotrequests":<Double_value>,

"cachetothits":<Double_value>,

"misccounter1":<Double_value>,

"numcpus":<Double_value>,

"pktcpuusagepcnt":<Double_value>,

"ssltransactionsrate":<Double_value>,

"httprequestsrate":<Double_value>,

"cachemaxmemoryactivekb":<Double_value>,

"httpresponsesrate":<Double_value>,

"disk0avail":<Double_value>,
```

```
"txmbitsrate":<Double_value>,

"sslcards":<Double_value>,

"hacurstate":<String_value>,

"rxmbitsrate":<Double_value>,

"tcpcurclientconnestablished":<Double_value>,

"cpuusagepcnt":<Double_value>,

"cachetotmisses":<Double_value>,

"httptotrxrequestbytes":<Double_value>

}]}
```

nsacl

Statistics for ACL entry resource.

Properties

(click to see Operations)

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
aclname	<String>	Read-write	Name of the extended ACL rule whose statistics you want the Citrix ADC to display. Minimum length = 1
clearstats	<String>	Read-write	Clear the statsistics / counters. Possible values = basic, full

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
acltotpktsbridged	<Double>	Read-only	Packets matching a bridge ACL, which is in transparent mode and bypasses service processing.
aclpktsbridgedrate	<Double>	Read-only	Rate (/s) counter for acltotpktsbridged
acltotpktsdenied	<Double>	Read-only	Packets dropped because they match ACLs with processing mode set to DENY.
aclpktsdeniedrate	<Double>	Read-only	Rate (/s) counter for acltotpktsdenied
acltotpktsallowed	<Double>	Read-only	Packets matching ACLs with processing mode set to ALLOW. Citrix ADC processes these packets.
aclpktsallowedrate	<Double>	Read-only	Rate (/s) counter for acltotpktsallowed
acltotpktsnat	<Double>	Read-only	Packets matching a NAT ACL, resulting in a NAT session.
aclpktsnatrate	<Double>	Read-only	Rate (/s) counter for acltotpktsnat
acltothits	<Double>	Read-only	Packets matching an ACL.
aclhitsrate	<Double>	Read-only	Rate (/s) counter for acltothits

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
acltotmisses	<Double>	Read-only	Packets not matching any ACL.
aclmissesrate	<Double>	Read-only	Rate (/s) counter for acltotmisses
acltotcount	<Double>	Read-only	Total number of ACL rules configured.
dfdacltothits	<Double>	Read-only	Packets matching an dfd ACL.
dfdaclhitsrate	<Double>	Read-only	Rate (/s) counter for dfdacltothits
dfdacltotmisses	<Double>	Read-only	Packets not matching any DFD ACL.
dfdaclmissesrate	<Double>	Read-only	Rate (/s) counter for dfdacltotmisses
dfdacltotcount	<Double>	Read-only	Total number of DFD ACL rules configured.
aclperhits	<Double>	Read-only	Number of times the acl was hit
aclperhitsrate	<Double>	Read-only	Rate (/s) counter for aclperhits

Operations

(click to see Properties)

GET (ALL)| GET

Some options that you can use for each operations:

- **Getting warnings in response:**NITRO allows you to get warnings in an operation by specifying the "warning" query parameter as "yes". For example, to get warnings while connecting to the Citrix ADC appliance, the URL is as follows:

  http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/login?warning=yes

  If any, the warnings are displayed in the response payload with the HTTP code "209 X-NITRO-WARNING".

- **Authenticated access for individual NITRO operations:**NITRO allows you to logon to the Citrix ADC appliance to perform individual operations. You can use this option instead of creating a NITRO session (using the login object) and then using that session to perform all operations,

  To do this, you must specify the username and password in the request header of the NITRO request as follows:

  X-NITRO-USER:<username>

  X-NITRO-PASS:<password>

  Note: In such cases, make sure that the request header DOES not include the following:

  Cookie:NITRO_AUTH_TOKEN=<tokenvalue>

`*Note:`*

Mandatory parameters are marked in redand placeholder content is marked in <green>.

get (all)

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/stat/nsacl

Query-parameters:

args

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/stat/nsacl?
args=aclname:<String_value>,detail:<Boolean_value>,fullvalues:<Boolean_value>,ntimes:<Double_valu

Use this query-parameter to get nsacl resources based on additional properties.

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "nsacl": [ {

"aclname":<String_value>,

"acltotpktsnat":<Double_value>,

"aclpktsdeniedrate":<Double_value>,

"aclhitsrate":<Double_value>,

"aclpktsallowedrate":<Double_value>,

"dfdaclhitsrate":<Double_value>,

"dfdacltotcount":<Double_value>,

"dfdacltothits":<Double_value>,

"acltotpktsallowed":<Double_value>,

"aclpktsbridgedrate":<Double_value>,

"acltotpktsdenied":<Double_value>,

"acltothits":<Double_value>,

"aclperhits":<Double_value>,

"dfdaclmissesrate":<Double_value>,
```

```
"aclperhitsrate":<Double_value>,

"aclpktsnatrate":<Double_value>,

"dfdacltotmisses":<Double_value>,

"acltotcount":<Double_value>,

"acltotmisses":<Double_value>,

"acltotpktsbridged":<Double_value>,

"aclmissesrate":<Double_value>

}]}
```

get

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/stat/nsacl/aclname_value><String>

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "nsacl": [ {

"aclname":<String_value>,
```

```
"acltotpktsnat":<Double_value>,

"aclpktsdeniedrate":<Double_value>,

"aclhitsrate":<Double_value>,

"aclpktsallowedrate":<Double_value>,

"dfdaclhitsrate":<Double_value>,

"dfdacltotcount":<Double_value>,

"dfdacltothits":<Double_value>,

"acltotpktsallowed":<Double_value>,

"aclpktsbridgedrate":<Double_value>,

"acltotpktsdenied":<Double_value>,

"acltothits":<Double_value>,

"aclperhits":<Double_value>,

"dfdaclmissesrate":<Double_value>,

"aclperhitsrate":<Double_value>,

"aclpktsnatrate":<Double_value>,

"dfdacltotmisses":<Double_value>,

"acltotcount":<Double_value>,

"acltotmisses":<Double_value>,

"acltotpktsbridged":<Double_value>,

"aclmissesrate":<Double_value>

}]}
```

nsacl6

Statistics for ACL6 entry resource.

Properties

(click to see Operations)

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
acl6name	<String>	Read-write	Name of the ACL6 rule whose statistics you want the Citrix ADC to display. Minimum length = 1
clearstats	<String>	Read-write	Clear the statsistics / counters. Possible values = basic, full
acl6totpktsbridged	<Double>	Read-only	Packets matching a bridge IPv6 ACL, which is in transparent mode and bypasses service processing.
acl6pktsbridgedrate	<Double>	Read-only	Rate (/s) counter for acl6totpktsbridged
acl6totpktsdenied	<Double>	Read-only	Packets dropped because they match IPv6 ACLs with processing mode set to DENY.
acl6pktsdeniedrate	<Double>	Read-only	Rate (/s) counter for acl6totpktsdenied

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
acl6totpktsallowed	<Double>	Read-only	Packets matching IPv6 ACLs with processing mode set to ALLOW. Citrix ADC processes these packets.
acl6pktsallowedrate	<Double>	Read-only	Rate (/s) counter for acl6totpktsallowed
acl6totpktsnat	<Double>	Read-only	Packets matching a NAT ACL6, resulting in a NAT session.
acl6pktsnatrate	<Double>	Read-only	Rate (/s) counter for acl6totpktsnat
acl6tothits	<Double>	Read-only	Packets matching an IPv6 ACL.
acl6hitsrate	<Double>	Read-only	Rate (/s) counter for acl6tothits
acl6totmisses	<Double>	Read-only	Packets not matching any IPv6 ACL.
acl6missesrate	<Double>	Read-only	Rate (/s) counter for acl6totmisses
acl6totpktsnat64	<Double>	Read-only	Packets matching a NAT64 ACL6, resulting in a NAT64 translation.
acl6pktsnat64rate	<Double>	Read-only	Rate (/s) counter for acl6totpktsnat64
acl6totcount	<Double>	Read-only	Total number of ACL6 rules configured.

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
dfdacl6totcount	<Double>	Read-only	Total number of DFD ACL6 rules configured.
dfdacl6tothits	<Double>	Read-only	Packets matching an dfd ACL6.
dfdacl6hitsrate	<Double>	Read-only	Rate (/s) counter for dfdacl6tothits
dfdacl6totmisses	<Double>	Read-only	Packets not matching any DFD ACL6.
dfdacl6missesrate	<Double>	Read-only	Rate (/s) counter for dfdacl6totmisses
acl6perhits	<Double>	Read-only	Number of times the acl6 was hit
acl6perhitsrate	<Double>	Read-only	Rate (/s) counter for acl6perhits

Operations

(click to see Properties)

GET (ALL)| GET

Some options that you can use for each operations:

- **Getting warnings in response:**NITRO allows you to get warnings in an operation by specifying the "warning" query parameter as "yes". For example, to get warnings while connecting to the Citrix ADC appliance, the URL is as follows:

  http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/login?warning=yes

  If any, the warnings are displayed in the response payload with the HTTP code "209 X-NITRO-WARNING".

- **Authenticated access for individual NITRO operations:**NITRO allows you to logon to the Citrix ADC appliance to perform individual operations. You can use this option instead of

creating a NITRO session (using the login object) and then using that session to perform all operations,

To do this, you must specify the username and password in the request header of the NITRO request as follows:

X-NITRO-USER:<username>

X-NITRO-PASS:<password>

Note: In such cases, make sure that the request header DOES not include the following:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue>

`*Note:`*

Mandatory parameters are marked in redand placeholder content is marked in <green>.

get (all)

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/stat/nsacl6

Query-parameters:

args

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/stat/nsacl6?
args=acl6name:<String_value>,detail:<Boolean_value>,fullvalues:<Boolean_value>,ntimes:<Double_va

Use this query-parameter to get nsacl6 resources based on additional properties.

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "nsacl6": [ {

"acl6name":<String_value>,

"acl6pktsnatrate":<Double_value>,

"acl6missesrate":<Double_value>,

"acl6totmisses":<Double_value>,

"dfdacl6hitsrate":<Double_value>,

"acl6totpktsnat64":<Double_value>,

"acl6pktsnat64rate":<Double_value>,

"acl6totpktsnat":<Double_value>,

"dfdacl6tothits":<Double_value>,

"dfdacl6totmisses":<Double_value>,

"acl6perhitsrate":<Double_value>,

"acl6perhits":<Double_value>,

"acl6tothits":<Double_value>,

"acl6pktsbridgedrate":<Double_value>,

"acl6pktsallowedrate":<Double_value>,

"acl6totpktsbridged":<Double_value>,

"acl6totpktsdenied":<Double_value>,

"acl6totcount":<Double_value>,

"acl6totpktsallowed":<Double_value>,

"acl6pktsdeniedrate":<Double_value>,
```

```
"acl6hitsrate":<Double_value>,

"dfdacl6missesrate":<Double_value>,

"dfdacl6totcount":<Double_value>

}]}
```

get

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/stat/nsacl6/acl6name_value><String>

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "nsacl6": [ {

"acl6name":<String_value>,

"acl6pktsnatrate":<Double_value>,

"acl6missesrate":<Double_value>,

"acl6totmisses":<Double_value>,

"dfdacl6hitsrate":<Double_value>,
```

```
"acl6totpktsnat64":<Double_value>,

"acl6pktsnat64rate":<Double_value>,

"acl6totpktsnat":<Double_value>,

"dfdacl6tothits":<Double_value>,

"dfdacl6totmisses":<Double_value>,

"acl6perhitsrate":<Double_value>,

"acl6perhits":<Double_value>,

"acl6tothits":<Double_value>,

"acl6pktsbridgedrate":<Double_value>,

"acl6pktsallowedrate":<Double_value>,

"acl6totpktsbridged":<Double_value>,

"acl6totpktsdenied":<Double_value>,

"acl6totcount":<Double_value>,

"acl6totpktsallowed":<Double_value>,

"acl6pktsdeniedrate":<Double_value>,

"acl6hitsrate":<Double_value>,

"dfdacl6missesrate":<Double_value>,

"dfdacl6totcount":<Double_value>

}]}
```

nslimitidentifier

Statistics for limit Indetifier resource.

Properties

(click to see Operations)

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
name	<String>	Read-write	The name of the identifier. Minimum length = 1
pattern	<String[]>	Read-write	Pattern for the selector field, ? means field is required, * means field value does not matter, anything else is a regular pattern.
clearstats	<String>	Read-write	Clear the statsistics / counters. Possible values = basic, full
sortby	<String>	Read-write	use this argument to sort by specific key. Possible values =
sortorder	<String>	Read-write	use this argument to specify sort order. Default value: SORT_DESCENDING Possible values = ascending, descending
ratelmtobjhits	<Double>	Read-only	Total hits.
ratelmtobjdrops	<Double>	Read-only	Total drops
ratelmtsessionobjhits	<Double>	Read-only	Total hits.

Operations

(click to see Properties)

GET (ALL)| GET

Some options that you can use for each operations:

- **Getting warnings in response:**NITRO allows you to get warnings in an operation by specifying the "warning" query parameter as "yes". For example, to get warnings while connecting to the Citrix ADC appliance, the URL is as follows:

  http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/login?warning=yes

  If any, the warnings are displayed in the response payload with the HTTP code "209 X-NITRO-WARNING".

- **Authenticated access for individual NITRO operations:**NITRO allows you to logon to the Citrix ADC appliance to perform individual operations. You can use this option instead of creating a NITRO session (using the login object) and then using that session to perform all operations,

  To do this, you must specify the username and password in the request header of the NITRO request as follows:

  X-NITRO-USER:<username>

  X-NITRO-PASS:<password>

  Note: In such cases, make sure that the request header DOES not include the following:

  Cookie:NITRO_AUTH_TOKEN=<tokenvalue>

`*Note:`*

Mandatory parameters are marked in redand placeholder content is marked in <green>.

get (all)

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/stat/nslimitidentifier

Query-parameters:

args

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/stat/nslimitidentifier?
args=name:<String_value>,pattern:<String_value>,detail:<Boolean_value>,fullvalues:<Boolean_value>,ntimes:<Double_value>,logfile:<String_val

Use this query-parameter to get nslimitidentifier resources based on additional properties.

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "nslimitidentifier": [ {

"name":<String_value>,

"ratelmtobjhits":<Double_value>,

"ratelmtsessionobjhits":<Double_value>,

"ratelmtobjdrops":<Double_value>

}]}
```

get

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/stat/nslimitidentifier/name_value><String>

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "nslimitidentifier": [ {

"name":<String_value>,

"ratelmtobjhits":<Double_value>,

"ratelmtsessionobjhits":<Double_value>,

"ratelmtobjdrops":<Double_value>

}]}
```

nsmemory

Statistics for resource.

Properties

(click to see Operations)

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
pool	<String>	Read-write	Feature name for which to display memory statistics. Minimum length = 1

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
clearstats	<String>	Read-write	Clear the statsistics / counters. Possible values = basic, full
allocf	<Double>	Read-only	Memory allocation failure for particular feature.
memcurallocper	<Double>	Read-only	Percentage of Citrix ADC memory used by the feature.
memcurinkb	<Double>	Read-only	Total current Citrix ADC memory available for use by the feature, in kilobytes.

Operations

(click to see Properties)

GET (ALL)| GET

Some options that you can use for each operations:

- **Getting warnings in response:**NITRO allows you to get warnings in an operation by specifying the "warning" query parameter as "yes". For example, to get warnings while connecting to the Citrix ADC appliance, the URL is as follows:

  http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/login?warning=yes

  If any, the warnings are displayed in the response payload with the HTTP code "209 X-NITRO-WARNING".
- **Authenticated access for individual NITRO operations:**NITRO allows you to logon to the Citrix ADC appliance to perform individual operations. You can use this option instead of creating a NITRO session (using the login object) and then using that session to perform all operations,

To do this, you must specify the username and password in the request header of the NITRO request as follows:

X-NITRO-USER:<username>

X-NITRO-PASS:<password>

Note: In such cases, make sure that the request header DOES not include the following:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue>

`*Note:`*

Mandatory parameters are marked in redand placeholder content is marked in <green>.

get (all)

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/stat/nsmemory

Query-parameters:

args

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/stat/nsmemory?
args=pool:<String_value>,detail:<Boolean_value>,fullvalues:<Boolean_value>,ntimes:<Double_value>,

Use this query-parameter to get nsmemory resources based on additional properties.

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "nsmemory": [ {

"pool":<String_value>,

"memcurinkb":<Double_value>,

"allocf":<Double_value>,

"memcurallocper":<Double_value>

}]}
```

get

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/stat/nsmemory/pool_value><String>

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "nsmemory": [ {

"pool":<String_value>,

"memcurinkb":<Double_value>,

"allocf":<Double_value>,
```

```
"memcurallocper":<Double_value>

}]}
```

nspartition

Statistics for admin partition resource.

Properties

(click to see Operations)

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
partitionname	<String>	Read-write	Name of the partition. Minimum length = 1
clearstats	<String>	Read-write	Clear the statsistics / counters. Possible values = basic, full
maxbandwidth	<Double>	Read-only	Maximum Banwidth allowed for the partition.
maxconnection	<Double>	Read-only	Maximum Connection allowed for the partition.
maxmemory	<Double>	Read-only	Maximum memory limit for the partition.
currentbandwidth	<Double>	Read-only	Current Bandwidth usage for the partition.
currentconnections	<Double>	Read-only	Current Connections on this partition.

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
memoryusagepcnt	<Double>	Read-only	Memory usage(%) on this partition.
totaldrops	<Double>	Read-only	Total packet drops for the partition.
dropsrate	<Double>	Read-only	Rate (/s) counter for totaldrops
totaltokendrops	<Double>	Read-only	Total drops(KB) for the partition.
tokendropsrate	<Double>	Read-only	Rate (/s) counter for totaltokendrops
totalconnectiondrops	<Double>	Read-only	Total connection drops for the partition.
connectiondropsrate	<Double>	Read-only	Rate (/s) counter for totalconnectiondrops

Operations

(click to see Properties)

GET (ALL)| GET

Some options that you can use for each operations:

- **Getting warnings in response:**NITRO allows you to get warnings in an operation by specifying the "warning" query parameter as "yes". For example, to get warnings while connecting to the Citrix ADC appliance, the URL is as follows:

  http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/login?warning=yes

  If any, the warnings are displayed in the response payload with the HTTP code "209 X-NITRO-WARNING".
- **Authenticated access for individual NITRO operations:**NITRO allows you to logon to the Citrix ADC appliance to perform individual operations. You can use this option instead of

creating a NITRO session (using the login object) and then using that session to perform all operations,

To do this, you must specify the username and password in the request header of the NITRO request as follows:

X-NITRO-USER:<username>

X-NITRO-PASS:<password>

Note: In such cases, make sure that the request header DOES not include the following:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue>

`*Note:`*

Mandatory parameters are marked in redand placeholder content is marked in <green>.

get (all)

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/stat/nspartition

Query-parameters:

args

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/stat/nspartition?
args=partitionname:<String_value>,detail:<Boolean_value>,fullvalues:<Boolean_value>,ntimes:<Doubl

Use this query-parameter to get nspartition resources based on additional properties.

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "nspartition": [ {
"partitionname":<String_value>,
"tokendropsrate":<Double_value>,
"totaltokendrops":<Double_value>,
"totaldrops":<Double_value>,
"dropsrate":<Double_value>,
"totalconnectiondrops":<Double_value>,
"currentconnections":<Double_value>,
"currentbandwidth":<Double_value>,
"maxconnection":<Double_value>,
"maxbandwidth":<Double_value>,
"connectiondropsrate":<Double_value>,
"maxmemory":<Double_value>,
"memoryusagepcnt":<Double_value>
}]}
```

get

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/stat/nspartition/partitionname_value><String>

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "nspartition": [ {

"partitionname":<String_value>,

"tokendropsrate":<Double_value>,

"totaltokendrops":<Double_value>,

"totaldrops":<Double_value>,

"dropsrate":<Double_value>,

"totalconnectiondrops":<Double_value>,

"currentconnections":<Double_value>,

"currentbandwidth":<Double_value>,

"maxconnection":<Double_value>,

"maxbandwidth":<Double_value>,

"connectiondropsrate":<Double_value>,

"maxmemory":<Double_value>,

"memoryusagepcnt":<Double_value>

}]}
```

nspbr

Statistics for Policy Based Routing(PBR) entry resource.

Properties

(click to see Operations)

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
name	<String>	Read-write	Name of the PBR whose statistics you want the Citrix ADC to display. Minimum length = 1
clearstats	<String>	Read-write	Clear the statsistics / counters. Possible values = basic, full
pbrtotpktsallowed	<Double>	Read-only	Total packets that matched the PBR (Policy-Based Routes) with action ALLOW
pbrpktsallowedrate	<Double>	Read-only	Rate (/s) counter for pbrtotpktsallowed
pbrtotpktsdenied	<Double>	Read-only	Total packets that matched the PBR with action DENY
pbrpktsdeniedrate	<Double>	Read-only	Rate (/s) counter for pbrtotpktsdenied
pbrtothits	<Double>	Read-only	Total packets that matched one of the configured PBR
pbrhitsrate	<Double>	Read-only	Rate (/s) counter for pbrtothits

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
pbrtotmisses	<Double>	Read-only	Total packets that did not match any PBR
pbrmissesrate	<Double>	Read-only	Rate (/s) counter for pbrtotmisses
pbrtotnulldrop	<Double>	Read-only	Total packets that are dropped due to null nexthop
pbrnulldroprate	<Double>	Read-only	Rate (/s) counter for pbrtotnulldrop
pbrperhits	<Double>	Read-only	Number of times the pbr was hit
pbrperhitsrate	<Double>	Read-only	Rate (/s) counter for pbrperhits

Operations

(click to see Properties)

GET (ALL)| GET

Some options that you can use for each operations:

- **Getting warnings in response:**NITRO allows you to get warnings in an operation by specifying the "warning" query parameter as "yes". For example, to get warnings while connecting to the Citrix ADC appliance, the URL is as follows:

  http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/login?warning=yes

  If any, the warnings are displayed in the response payload with the HTTP code "209 X-NITRO-WARNING".
- **Authenticated access for individual NITRO operations:**NITRO allows you to logon to the Citrix ADC appliance to perform individual operations. You can use this option instead of creating a NITRO session (using the login object) and then using that session to perform all operations,

To do this, you must specify the username and password in the request header of the NITRO request as follows:

X-NITRO-USER:<username>

X-NITRO-PASS:<password>

Note: In such cases, make sure that the request header DOES not include the following:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue>

`*Note:`*

Mandatory parameters are marked in redand placeholder content is marked in <green>.

get (all)

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/stat/nspbr

Query-parameters:

args

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/stat/nspbr?
args=name:<String_value>,detail:<Boolean_value>,fullvalues:<Boolean_value>,ntimes:<Double_value>

Use this query-parameter to get nspbr resources based on additional properties.

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "nspbr": [ {

"name":<String_value>,

"pbrpktsdeniedrate":<Double_value>,

"pbrmissesrate":<Double_value>,

"pbrnulldroprate":<Double_value>,

"pbrtothits":<Double_value>,

"pbrperhitsrate":<Double_value>,

"pbrtotnulldrop":<Double_value>,

"pbrhitsrate":<Double_value>,

"pbrperhits":<Double_value>,

"pbrtotmisses":<Double_value>,

"pbrtotpktsallowed":<Double_value>,

"pbrtotpktsdenied":<Double_value>,

"pbrpktsallowedrate":<Double_value>

}]}
```

get

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/stat/nspbr/name_value><String>

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "nspbr": [ {

"name":<String_value>,

"pbrpktsdeniedrate":<Double_value>,

"pbrmissesrate":<Double_value>,

"pbrnulldroprate":<Double_value>,

"pbrtothits":<Double_value>,

"pbrperhitsrate":<Double_value>,

"pbrtotnulldrop":<Double_value>,

"pbrhitsrate":<Double_value>,

"pbrperhits":<Double_value>,

"pbrtotmisses":<Double_value>,

"pbrtotpktsallowed":<Double_value>,

"pbrtotpktsdenied":<Double_value>,

"pbrpktsallowedrate":<Double_value>

}]}
```

nspbr6

Statistics for PBR6 entry resource.

Properties

(click to see Operations)

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
name	<String>	Read-write	Name of the PBR6 whose statistics you want the Citrix ADC to display. Minimum length = 1
clearstats	<String>	Read-write	Clear the statsistics / counters. Possible values = basic, full
pbr6totpktsallowed	<Double>	Read-only	Total packets that matched the PBR6 with action ALLOW
pbr6pktsallowedrate	<Double>	Read-only	Rate (/s) counter for pbr6totpktsallowed
pbr6totpktsdenied	<Double>	Read-only	Total packets that matched PBR6 with action DENY
pbr6pktsdeniedrate	<Double>	Read-only	Rate (/s) counter for pbr6totpktsdenied
pbr6tothits	<Double>	Read-only	Total packets that matched one of the configured PBR6
pbr6hitsrate	<Double>	Read-only	Rate (/s) counter for pbr6tothits
pbr6totmisses	<Double>	Read-only	Total packets that did not match any PBR6

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
pbr6missesrate	<Double>	Read-only	Rate (/s) counter for pbr6totmisses
pbr6totnulldrop	<Double>	Read-only	Total packets that are dropped due to null next hop
pbr6nulldroprate	<Double>	Read-only	Rate (/s) counter for pbr6totnulldrop
pbr6perhits	<Double>	Read-only	Number of times the pbr6 was hit
pbr6perhitsrate	<Double>	Read-only	Rate (/s) counter for pbr6perhits

Operations

(click to see Properties)

GET (ALL)| GET

Some options that you can use for each operations:

- **Getting warnings in response:**NITRO allows you to get warnings in an operation by specifying the "warning" query parameter as "yes". For example, to get warnings while connecting to the Citrix ADC appliance, the URL is as follows:

  http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/login?warning=yes

  If any, the warnings are displayed in the response payload with the HTTP code "209 X-NITRO-WARNING".

- **Authenticated access for individual NITRO operations:**NITRO allows you to logon to the Citrix ADC appliance to perform individual operations. You can use this option instead of creating a NITRO session (using the login object) and then using that session to perform all operations,

  To do this, you must specify the username and password in the request header of the NITRO request as follows:

  X-NITRO-USER:<username>

X-NITRO-PASS:<password>

Note: In such cases, make sure that the request header DOES not include the following:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue>

`*Note:`*

Mandatory parameters are marked in redand placeholder content is marked in <green>.

get (all)

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/stat/nspbr6

Query-parameters:

args

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/stat/nspbr6?
args=name:<String_value>,detail:<Boolean_value>,fullvalues:<Boolean_value>,ntimes:<Double_value>

Use this query-parameter to get nspbr6 resources based on additional properties.

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "nspbr6": [ {

"name":<String_value>,

"pbr6perhits":<Double_value>,
```

```
"pbr6totpktsallowed":<Double_value>,

"pbr6pktsallowedrate":<Double_value>,

"pbr6totpktsdenied":<Double_value>,

"pbr6hitsrate":<Double_value>,

"pbr6nulldroprate":<Double_value>,

"pbr6totnulldrop":<Double_value>,

"pbr6perhitsrate":<Double_value>,

"pbr6totmisses":<Double_value>,

"pbr6pktsdeniedrate":<Double_value>,

"pbr6missesrate":<Double_value>,

"pbr6tothits":<Double_value>

}]}
```

get

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/stat/nspbr6/name_value><String>

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "nspbr6": [ {
"name":<String_value>,
"pbr6perhits":<Double_value>,
"pbr6totpktsallowed":<Double_value>,
"pbr6pktsallowedrate":<Double_value>,
"pbr6totpktsdenied":<Double_value>,
"pbr6hitsrate":<Double_value>,
"pbr6nulldroprate":<Double_value>,
"pbr6totnulldrop":<Double_value>,
"pbr6perhitsrate":<Double_value>,
"pbr6totmisses":<Double_value>,
"pbr6pktsdeniedrate":<Double_value>,
"pbr6missesrate":<Double_value>,
"pbr6tothits":<Double_value>
}]}
```

nssimpleacl

Statistics for simple ACL resource.

Properties

(click to see Operations)

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
clearstats	<String>	Read-write	Clear the statsistics / counters. Possible values = basic, full
sacltothits	<Double>	Read-only	Packets matching a SimpleACL.
saclhitsrate	<Double>	Read-only	Rate (/s) counter for sacltothits
sacltotmisses	<Double>	Read-only	Packets not matching any SimpleACL.
saclmissesrate	<Double>	Read-only	Rate (/s) counter for sacltotmisses
saclscount	<Double>	Read-only	Number of SimpleACLs configured.
sacltotpktsallowed	<Double>	Read-only	Total packets that matched a SimpleACL with action ALLOW and got consumed by Citrix ADC.
saclpktsallowedrate	<Double>	Read-only	Rate (/s) counter for sacltotpktsallowed
sacltotpktsbridged	<Double>	Read-only	Total packets that matched a SimpleACL with action BRIDGE and got bridged by Citrix ADC.
saclpktsbridgedrate	<Double>	Read-only	Rate (/s) counter for sacltotpktsbridged

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
sacltotpktsdenied	<Double>	Read-only	Packets dropped because they match SimpleACL (Access Control List) with processing mode set to DENY.
saclpktsdeniedrate	<Double>	Read-only	Rate (/s) counter for sacltotpktsdenied

Operations

(click to see Properties)

GET (ALL)

Some options that you can use for each operations:

- **Getting warnings in response:**NITRO allows you to get warnings in an operation by specifying the "warning" query parameter as "yes". For example, to get warnings while connecting to the Citrix ADC appliance, the URL is as follows:

  http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/login?warning=yes

  If any, the warnings are displayed in the response payload with the HTTP code "209 X-NITRO-WARNING".
- **Authenticated access for individual NITRO operations:**NITRO allows you to logon to the Citrix ADC appliance to perform individual operations. You can use this option instead of creating a NITRO session (using the login object) and then using that session to perform all operations,

  To do this, you must specify the username and password in the request header of the NITRO request as follows:

  X-NITRO-USER:<username>

  X-NITRO-PASS:<password>

  Note: In such cases, make sure that the request header DOES not include the following:

  Cookie:NITRO_AUTH_TOKEN=<tokenvalue>

`*Note:` *

Mandatory parameters are marked in redand placeholder content is marked in <green>.

get (all)

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/stat/nssimpleacl

Query-parameters:

args

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/stat/nssimpleacl?
args=detail:<Boolean_value>,fullvalues:<Boolean_value>,ntimes:<Double_value>,logfile:<String_value

Use this query-parameter to get nssimpleacl resources based on additional properties.

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "nssimpleacl": [ {

"saclscount":<Double_value>,

"saclhitsrate":<Double_value>,

"sacltotmisses":<Double_value>,

"saclpktsdeniedrate":<Double_value>,
```

```
"sacltotpktsallowed":<Double_value>,

"sacltothits":<Double_value>,

"sacltotpktsdenied":<Double_value>,

"saclpktsallowedrate":<Double_value>,

"saclpktsbridgedrate":<Double_value>,

"saclmissesrate":<Double_value>,

"sacltotpktsbridged":<Double_value>

}]}
```

nssimpleacl6

Statistics for simple ACL6 resource.

Properties

(click to see Operations)

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
clearstats	<String>	Read-write	Clear the statsistics / counters. Possible values = basic, full
sacl6tothits	<Double>	Read-only	Packets matching a SimpleACL6.
sacl6hitsrate	<Double>	Read-only	Rate (/s) counter for sacl6tothits
sacl6totmisses	<Double>	Read-only	Packets not matching any SimpleACL6.

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
sacl6missesrate	<Double>	Read-only	Rate (/s) counter for sacl6totmisses
sacl6scount	<Double>	Read-only	Number of SimpleACL6s configured.
sacl6totpktsallowed	<Double>	Read-only	Total packets that matched a SimpleACL6 with action ALLOW and got consumed by Citrix ADC.
sacl6pktsallowedrate	<Double>	Read-only	Rate (/s) counter for sacl6totpktsallowed
sacl6totpktsbridged	<Double>	Read-only	Total packets that matched a SimpleACL6 with action BRIDGE and got bridged by Citrix ADC.
sacl6pktsbridgedrate	<Double>	Read-only	Rate (/s) counter for sacl6totpktsbridged
sacl6totpktsdenied	<Double>	Read-only	Packets dropped because they match SimpleACL6 with processing mode set to DENY.
sacl6pktsdeniedrate	<Double>	Read-only	Rate (/s) counter for sacl6totpktsdenied

Operations

(click to see Properties)

GET (ALL)

Some options that you can use for each operations:

- **Getting warnings in response:**NITRO allows you to get warnings in an operation by specifying the "warning" query parameter as "yes". For example, to get warnings while connecting to the Citrix ADC appliance, the URL is as follows:

  http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/login?warning=yes

  If any, the warnings are displayed in the response payload with the HTTP code "209 X-NITRO-WARNING".

- **Authenticated access for individual NITRO operations:**NITRO allows you to logon to the Citrix ADC appliance to perform individual operations. You can use this option instead of creating a NITRO session (using the login object) and then using that session to perform all operations,

  To do this, you must specify the username and password in the request header of the NITRO request as follows:

  X-NITRO-USER:<username>

  X-NITRO-PASS:<password>

  Note: In such cases, make sure that the request header DOES not include the following:

  Cookie:NITRO_AUTH_TOKEN=<tokenvalue>

`*Note:`*

Mandatory parameters are marked in redand placeholder content is marked in <green>.

get (all)

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/stat/nssimpleacl6

Query-parameters:

args

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/stat/nssimpleacl6?
args=detail:<Boolean_value>,fullvalues:<Boolean_value>,ntimes:<Double_value>,logfile:<String_value

Use this query-parameter to get nssimpleacl6 resources based on additional properties.

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "nssimpleacl6": [ {

"sacl6scount":<Double_value>,

"sacl6pktsbridgedrate":<Double_value>,

"sacl6totpktsallowed":<Double_value>,

"sacl6totmisses":<Double_value>,

"sacl6missesrate":<Double_value>,

"sacl6hitsrate":<Double_value>,

"sacl6tothits":<Double_value>,

"sacl6totpktsdenied":<Double_value>,

"sacl6pktsallowedrate":<Double_value>,

"sacl6totpktsbridged":<Double_value>,

"sacl6pktsdeniedrate":<Double_value>

}]}
```

nstrafficdomain

Statistics for Traffic Domain resource.

Properties

(click to see Operations)

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
td	<Double>	Read-write	An integer specifying the Traffic Domain ID. Possible values: 1 through 4094. Minimum value = 1 Maximum value = 4094
clearstats	<String>	Read-write	Clear the statsistics / counters. Possible values = basic, full
nstdtotrxpkts	<Double>	Read-only	Packets received on this TD.
nstdrxpktsrate	<Double>	Read-only	Rate (/s) counter for nstdtotrxpkts
nstdtottxpkts	<Double>	Read-only	Packets transmitted from this TD.
nstdtxpktsrate	<Double>	Read-only	Rate (/s) counter for nstdtottxpkts
nstdtotdroppedpkts	<Double>	Read-only	Inbound packets dropped on this TD by reception.
nstddroppedpktsrate	<Double>	Read-only	Rate (/s) counter for nstdtotdroppedpkts

Operations

(click to see Properties)

GET (ALL)| GET

Some options that you can use for each operations:

- **Getting warnings in response:**NITRO allows you to get warnings in an operation by specifying the "warning" query parameter as "yes". For example, to get warnings while connecting to the Citrix ADC appliance, the URL is as follows:

  http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/login?warning=yes

  If any, the warnings are displayed in the response payload with the HTTP code "209 X-NITRO-WARNING".

- **Authenticated access for individual NITRO operations:**NITRO allows you to logon to the Citrix ADC appliance to perform individual operations. You can use this option instead of creating a NITRO session (using the login object) and then using that session to perform all operations,

  To do this, you must specify the username and password in the request header of the NITRO request as follows:

  X-NITRO-USER:<username>

  X-NITRO-PASS:<password>

  Note: In such cases, make sure that the request header DOES not include the following:

  Cookie:NITRO_AUTH_TOKEN=<tokenvalue>

`*Note:`*

Mandatory parameters are marked in redand placeholder content is marked in <green>.

get (all)

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/stat/nstrafficdomain

Query-parameters:

args

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/stat/nstrafficdomain?
args=td:<Double_value>,detail:<Boolean_value>,fullvalues:<Boolean_value>,ntimes:<Double_value>,lo

Use this query-parameter to get nstrafficdomain resources based on additional properties.

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "nstrafficdomain": [ {

"td":<Double_value>,

"nstdtotrxpkts":<Double_value>,

"nstddroppedpktsrate":<Double_value>,

"nstdtotdroppedpkts":<Double_value>,

"nstdrxpktsrate":<Double_value>,

"nstdtottxpkts":<Double_value>,

"nstdtxpktsrate":<Double_value>

}]}
```

get

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/stat/nstrafficdomain/td_value><Double>

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "nstrafficdomain": [ {

"td":<Double_value>,

"nstdtotrxpkts":<Double_value>,

"nstddroppedpktsrate":<Double_value>,

"nstdtotdroppedpkts":<Double_value>,

"nstdrxpktsrate":<Double_value>,

"nstdtottxpkts":<Double_value>,

"nstdtxpktsrate":<Double_value>

}]}
```

Pcp

Port Control Protocol statistics.

NAME	DESCRIPTION
pcpserver	Statistics for server resource.

pcpserver

Statistics for server resource.

Properties

(click to see Operations)

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
name	<String>	Read-write	PCP Statistics per Server.
clearstats	<String>	Read-write	Clear the statsistics / counters. Possible values = basic, full
pcptotrequests	<Double>	Read-only	PCP Request received.
pcprequestsrate	<Double>	Read-only	Rate (/s) counter for pcptotrequests
pcptotresponses	<Double>	Read-only	Number PCP responces sent.
pcpresponsesrate	<Double>	Read-only	Rate (/s) counter for pcptotresponses
pcptotmaprequests	<Double>	Read-only	PCP MAP Requests received.
pcpmaprequestsrate	<Double>	Read-only	Rate (/s) counter for pcptotmaprequests
pcptotpeerrequests	<Double>	Read-only	PCP PEER Requests received.

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
pcppeerrequestsrate	<Double>	Read-only	Rate (/s) counter for pcptotpeerrequests
pcptoterrinrquest	<Double>	Read-only	total PCP request with error.
pcperrinrquestrate	<Double>	Read-only	Rate (/s) counter for pcptoterrinrquest
pcptoterrinres	<Double>	Read-only	Total PCP responses with errors.
pcperrinresrate	<Double>	Read-only	Rate (/s) counter for pcptoterrinres
pcptotunsuppvers	<Double>	Read-only	PCP request with unsupported version.
pcpunsuppversrate	<Double>	Read-only	Rate (/s) counter for pcptotunsuppvers
pcptotmalformedreq	<Double>	Read-only	total PCP request having malformed PCP packets.
pcpmalformedreqrate	<Double>	Read-only	Rate (/s) counter for pcptotmalformedreq
pcptotunsuppopcode	<Double>	Read-only	total Unsupproted OPCODES received Requests.
pcpunsuppopcoderate	<Double>	Read-only	Rate (/s) counter for pcptotunsuppopcode
pcptotunsuppoption	<Double>	Read-only	total Unsupproted OPTIONS received in requests.
pcpunsuppoptionrate	<Double>	Read-only	Rate (/s) counter for pcptotunsuppoption

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
pcptotmalformedoption	<Double>	Read-only	total malformed OPTIONS received in requests.
pcpmalformedoptionrate	<Double>	Read-only	Rate (/s) counter for pcptotmalformedoption
pcptotnetfailure	<Double>	Read-only	Total Network Failures.
pcpnetfailurerate	<Double>	Read-only	Rate (/s) counter for pcptotnetfailure
pcptotnoresources	<Double>	Read-only	no resources
pcpnoresourcesrate	<Double>	Read-only	Rate (/s) counter for pcptotnoresources
pcptotunsupportedprotocol	<Double>	Read-only	Total Unsupported Protocols requests.
pcpunsupportedprotocolrate	<Double>	Read-only	Rate (/s) counter for pcptotunsupportedprotocol
pcptotuserexqouta	<Double>	Read-only	Total user ex quota.
pcpuserexqoutarate	<Double>	Read-only	Rate (/s) counter for pcptotuserexqouta
pcptotnoexternal	<Double>	Read-only	Total responses with opcode can not provide external.
pcpnoexternalrate	<Double>	Read-only	Rate (/s) counter for pcptotnoexternal
pcptotaddrmismatch	<Double>	Read-only	Total address mismatch.

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
pcpaddrmismatchrate	<Double>	Read-only	Rate (/s) counter for pcptotaddrmismatch
pcptotexcesspeers	<Double>	Read-only	Total responses with opcode excessive remote peers.
pcpexcesspeersrate	<Double>	Read-only	Rate (/s) counter for pcptotexcesspeers

Operations

(click to see Properties)

GET (ALL)| GET

Some options that you can use for each operations:

- **Getting warnings in response:**NITRO allows you to get warnings in an operation by specifying the "warning" query parameter as "yes". For example, to get warnings while connecting to the Citrix ADC appliance, the URL is as follows:

  http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/login?warning=yes

  If any, the warnings are displayed in the response payload with the HTTP code "209 X-NITRO-WARNING".
- **Authenticated access for individual NITRO operations:**NITRO allows you to logon to the Citrix ADC appliance to perform individual operations. You can use this option instead of creating a NITRO session (using the login object) and then using that session to perform all operations,

  To do this, you must specify the username and password in the request header of the NITRO request as follows:

  X-NITRO-USER:<username>

  X-NITRO-PASS:<password>

  Note: In such cases, make sure that the request header DOES not include the following:

  Cookie:NITRO_AUTH_TOKEN=<tokenvalue>

`*Note:` *

Mandatory parameters are marked in redand placeholder content is marked in <green>.

get (all)

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/stat/pcpserver

Query-parameters:

args

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/stat/pcpserver?
args=name:<String_value>,detail:<Boolean_value>,fullvalues:<Boolean_value>,ntimes:<Double_value>

Use this query-parameter to get pcpserver resources based on additional properties.

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "pcpserver": [ {

"name":<String_value>,

"pcpaddrmismatchrate":<Double_value>,

"pcptotmalformedreq":<Double_value>,

"pcpnoresourcesrate":<Double_value>,
```

```
"pcpunsuppopcoderate":<Double_value>,

"pcptotpeerrequests":<Double_value>,

"pcpnetfailurerate":<Double_value>,

"pcptoterrinrquest":<Double_value>,

"pcptotmalformedoption":<Double_value>,

"pcptotresponses":<Double_value>,

"pcpunsupportedprotocolrate":<Double_value>,

"pcptotnoresources":<Double_value>,

"pcptotnetfailure":<Double_value>,

"pcppeerrequestsrate":<Double_value>,

"pcptotnoexternal":<Double_value>,

"pcperrinresrate":<Double_value>,

"pcptotunsuppopcode":<Double_value>,

"pcpunsuppversrate":<Double_value>,

"pcptotuserexqouta":<Double_value>,

"pcptotunsuppvers":<Double_value>,

"pcpuserexqoutarate":<Double_value>,

"pcptotunsupportedprotocol":<Double_value>,

"pcpexcesspeersrate":<Double_value>,

"pcpmalformedoptionrate":<Double_value>,

"pcptotrequests":<Double_value>,

"pcperrinrquestrate":<Double_value>,
```

```
"pcpresponsesrate":<Double_value>,

"pcpunsuppoptionrate":<Double_value>,

"pcpmaprequestsrate":<Double_value>,

"pcpnoexternalrate":<Double_value>,

"pcptotunsuppoption":<Double_value>,

"pcptoterrinres":<Double_value>,

"pcptotmaprequests":<Double_value>,

"pcptotexcesspeers":<Double_value>,

"pcpmalformedreqrate":<Double_value>,

"pcprequestsrate":<Double_value>,

"pcptotaddrmismatch":<Double_value>

}]}
```

get

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/stat/pcpserver/name_value><String>

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "pcpserver": [ {
"name":<String_value>,
"pcpaddrmismatchrate":<Double_value>,
"pcptotmalformedreq":<Double_value>,
"pcpnoresourcesrate":<Double_value>,
"pcpunsuppopcoderate":<Double_value>,
"pcptotpeerrequests":<Double_value>,
"pcpnetfailurerate":<Double_value>,
"pcptoterrinrquest":<Double_value>,
"pcptotmalformedoption":<Double_value>,
"pcptotresponses":<Double_value>,
"pcpunsupportedprotocolrate":<Double_value>,
"pcptotnoresources":<Double_value>,
"pcptotnetfailure":<Double_value>,
"pcppeerrequestsrate":<Double_value>,
"pcptotnoexternal":<Double_value>,
"pcperrinresrate":<Double_value>,
"pcptotunsuppopcode":<Double_value>,
"pcpunsuppversrate":<Double_value>,
"pcptotuserexqouta":<Double_value>,
"pcptotunsuppvers":<Double_value>,
```

```
"pcpuserexqoutarate":<Double_value>,

"pcptotunsupportedprotocol":<Double_value>,

"pcpexcesspeersrate":<Double_value>,

"pcpmalformedoptionrate":<Double_value>,

"pcptotrequests":<Double_value>,

"pcperrinrquestrate":<Double_value>,

"pcpresponsesrate":<Double_value>,

"pcpunsuppoptionrate":<Double_value>,

"pcpmaprequestsrate":<Double_value>,

"pcpnoexternalrate":<Double_value>,

"pcptotunsuppoption":<Double_value>,

"pcptoterrinres":<Double_value>,

"pcptotmaprequests":<Double_value>,

"pcptotexcesspeers":<Double_value>,

"pcpmalformedreqrate":<Double_value>,

"pcprequestsrate":<Double_value>,

"pcptotaddrmismatch":<Double_value>

}]}
```

Priority Queuing

Priority Queuing statistics. Priority Queuing ensures that key customer web sites get the timely responses they need. System provides priority-based queuing, which provides different levels of service to customers.

NAME	DESCRIPTION
pq	Statistics for pq.
pqpolicy	Statistics for PQ policy resource.

pq

Statistics for pq.

Properties

(click to see Operations)

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
clearstats	<String>	Read-write	Clear the statsistics / counters. Possible values = basic, full
pqtotalpolicymatches	<Double>	Read-only	Number of times the Citrix ADC matched an incoming request using any priority queuing policy.
pqpolicymatchesrate	<Double>	Read-only	Rate (/s) counter for pqtotalpolicymatches
pqtotalthresholdfailed	<Double>	Read-only	Number of times the Citrix ADC failed to match an incoming request to any of priority queing policy.

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
pqthresholdfailedrate	<Double>	Read-only	Rate (/s) counter for pqtotalthresholdfailed
pqpriority1requests	<Double>	Read-only	Number of priority 1 requests that the Citrix ADC received.
pqpriority1requestsrate	<Double>	Read-only	Rate (/s) counter for pqpriority1requests
pqpriority2requests	<Double>	Read-only	Number of priority 2 requests that the Citrix ADC received.
pqpriority2requestsrate	<Double>	Read-only	Rate (/s) counter for pqpriority2requests
pqpriority3requests	<Double>	Read-only	Number of priority 3 requests that the Citrix ADC received.
pqpriority3requestsrate	<Double>	Read-only	Rate (/s) counter for pqpriority3requests

Operations

(click to see Properties)

GET (ALL)

Some options that you can use for each operations:

- **Getting warnings in response:**NITRO allows you to get warnings in an operation by specifying the "warning" query parameter as "yes". For example, to get warnings while connecting to the Citrix ADC appliance, the URL is as follows:

  http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/login?warning=yes

  If any, the warnings are displayed in the response payload with the HTTP code "209 X-NITRO-WARNING".

- **Authenticated access for individual NITRO operations:**NITRO allows you to logon to the Citrix ADC appliance to perform individual operations. You can use this option instead of creating a NITRO session (using the login object) and then using that session to perform all operations,

  To do this, you must specify the username and password in the request header of the NITRO request as follows:

  X-NITRO-USER:<username>

  X-NITRO-PASS:<password>

  Note: In such cases, make sure that the request header DOES not include the following:

  Cookie:NITRO_AUTH_TOKEN=<tokenvalue>

`*Note:`*

Mandatory parameters are marked in redand placeholder content is marked in <green>.

get (all)

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/stat/pq

Query-parameters:

args

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/stat/pq?
args=detail:<Boolean_value>,fullvalues:<Boolean_value>,ntimes:<Double_value>,logfile:<String_value

Use this query-parameter to get pq resources based on additional properties.

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "pq": [ {
"pqpriority2requestsrate":<Double_value>,
"pqpolicymatchesrate":<Double_value>,
"pqpriority1requestsrate":<Double_value>,
"pqthresholdfailedrate":<Double_value>,
"pqtotalpolicymatches":<Double_value>,
"pqpriority1requests":<Double_value>,
"pqpriority3requestsrate":<Double_value>,
"pqpriority3requests":<Double_value>,
"pqpriority2requests":<Double_value>,
"pqtotalthresholdfailed":<Double_value>
}]}
```

pqpolicy

Statistics for PQ policy resource.

Properties

(click to see Operations)

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
policyname	<String>	Read-write	Name of the priority queuing policy whose statistics must be displayed. If a

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
			name is not provided, statistics of all priority queuing policies are shown. Minimum length = 1
clearstats	<String>	Read-write	Clear the statsistics / counters. Possible values = basic, full
pqtotavgqueuewaittime	<Double>	Read-only	Average wait time for clients for this priority queuing policy.
pqavgqueuewaittimerate	<Double>	Read-only	Rate (/s) counter for pqtotavgqueuewaittime
pqavgclienttransactiontimems	<Double>	Read-only	Average time taken by a priority queuing client to complete its transaction for this priority queuing policy.
pqavgclienttransactiontimemsrate	<Double>	Read-only	Rate (/s) counter for pqavgclienttransactiontimems
pqvserverip	<String>	Read-only	IP address of the virtual server to which this priority queuing policy is bound.
pqvserverport	<Integer>	Read-only	Port number of the virtual server to which this priority queuing policy is bound.

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
pqqdepth	<Double>	Read-only	Number of clients waiting currently for this priority queuing policy.
pqqdepthrate	<Double>	Read-only	Rate (/s) counter for pqqdepth
pqcurrentclientconnections	<Double>	Read-only	Current number of server connections established for serving clients for this priority queuing policy.
pqcurrentclientconnectionsrate	<Double>	Read-only	Rate (/s) counter for pqcurrentclientconnections
pqtotclientconnections	<Double>	Read-only	Total number of server connections established for serving clients for this priority queuing policy.
pqclientconnectionsrate	<Double>	Read-only	Rate (/s) counter for pqtotclientconnections
pqdropped	<Double>	Read-only	Total number of dropped transactions for this priority queuing policy.
pqdroppedrate	<Double>	Read-only	Rate (/s) counter for pqdropped
totclienttransactions	<Double>	Read-only	Total number of client transactions for

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
			this priority queuing policy.
clienttransactionsrate	<Double>	Read-only	Rate (/s) counter for totclienttransactions
pqtotqueuedepth	<Double>	Read-only	Total number of waiting clients for this priority queuing policy.
pqqueuedepthrate	<Double>	Read-only	Rate (/s) counter for pqtotqueuedepth

Operations

(click to see Properties)

GET (ALL)| GET

Some options that you can use for each operations:

- **Getting warnings in response:**NITRO allows you to get warnings in an operation by specifying the "warning" query parameter as "yes". For example, to get warnings while connecting to the Citrix ADC appliance, the URL is as follows:

  http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/login?warning=yes

  If any, the warnings are displayed in the response payload with the HTTP code "209 X-NITRO-WARNING".
- **Authenticated access for individual NITRO operations:**NITRO allows you to logon to the Citrix ADC appliance to perform individual operations. You can use this option instead of creating a NITRO session (using the login object) and then using that session to perform all operations,

  To do this, you must specify the username and password in the request header of the NITRO request as follows:

  X-NITRO-USER:<username>

  X-NITRO-PASS:<password>

Note: In such cases, make sure that the request header DOES not include the following:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue>

`*Note:`*

Mandatory parameters are marked in redand placeholder content is marked in <green>.

get (all)

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/stat/pqpolicy

Query-parameters:

args

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/stat/pqpolicy?
args=policyname:<String_value>,detail:<Boolean_value>,fullvalues:<Boolean_value>,ntimes:<Double_

Use this query-parameter to get pqpolicy resources based on additional properties.

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "pqpolicy": [ {

"policyname":<String_value>,

"pqvserverip":<String_value>,

"pqqueuedepthrate":<Double_value>,
```

```
"pqdroppedrate":<Double_value>,

"pqtotqueuedepth":<Double_value>,

"pqvserverport":<Integer_value>,

"totclienttransactions":<Double_value>,

"clienttransactionsrate":<Double_value>,

"pqcurrentclientconnectionsrate":<Double_value>,

"pqqdepthrate":<Double_value>,

"pqqdepth":<Double_value>,

"pqclientconnectionsrate":<Double_value>,

"pqavgclienttransactiontimems":<Double_value>,

"pqavgclienttransactiontimemsrate":<Double_value>,

"pqtotavgqueuewaittime":<Double_value>,

"pqdropped":<Double_value>,

"pqtotclientconnections":<Double_value>,

"pqavgqueuewaittimerate":<Double_value>,

"pqcurrentclientconnections":<Double_value>

}]}
```

get

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/stat/pqpolicy/policyname_value><String>

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "pqpolicy": [ {

"policyname":<String_value>,

"pqvserverip":<String_value>,

"pqqueuedepthrate":<Double_value>,

"pqdroppedrate":<Double_value>,

"pqtotqueuedepth":<Double_value>,

"pqvserverport":<Integer_value>,

"totclienttransactions":<Double_value>,

"clienttransactionsrate":<Double_value>,

"pqcurrentclientconnectionsrate":<Double_value>,

"pqqdepthrate":<Double_value>,

"pqqdepth":<Double_value>,

"pqclientconnectionsrate":<Double_value>,

"pqavgclienttransactiontimems":<Double_value>,

"pqavgclienttransactiontimemsrate":<Double_value>,
```

```
"pqtotavgqueuewaittime":<Double_value>,

"pqdropped":<Double_value>,

"pqtotclientconnections":<Double_value>,

"pqavgqueuewaittimerate":<Double_value>,

"pqcurrentclientconnections":<Double_value>

}]}
```

Protocol

Protocol statistics.

NAME	DESCRIPTION
protocolhttp	Statistics for HTTP resource.
protocolicmp	Statistics for icmp resource.
protocolicmpv6	Statistics for icmpv6 resource.
protocolip	Statistics for protocolip resource.
protocolipv6	Statistics for ip v6 resource.
protocolmptcp	Statistics for mptcp protocol resource.
protocoltcp	Statistics for tcp protocol resource.
protocoludp	Statistics for UDP Protocol resource.

protocolhttp

Statistics for HTTP resource.

Properties

(click to see Operations)

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
clearstats	<String>	Read-write	Clear the statsistics / counters. Possible values = basic, full
spdytotstreams	<Double>	Read-only	Total number of requests received over SPDYv2 and SPDYv3
spdystreamsrate	<Double>	Read-only	Rate (/s) counter for spdytotstreams
httptotrequests	<Double>	Read-only	Total number of HTTP requests received.
httprequestsrate	<Double>	Read-only	Rate (/s) counter for httptotrequests
httptotresponses	<Double>	Read-only	Total number of HTTP responses sent.
httpresponsesrate	<Double>	Read-only	Rate (/s) counter for httptotresponses
httptotrxrequestbytes	<Double>	Read-only	Total number of bytes of HTTP request data received.
httprxrequestbytesrate	<Double>	Read-only	Rate (/s) counter for httptotrxrequestbytes
httptotrxresponsebytes	<Double>	Read-only	Total number of bytes of HTTP response data received.

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
httprxresponsebytesrate	<Double>	Read-only	Rate (/s) counter for httptotrxresponsebytes
httptotgets	<Double>	Read-only	Total number of HTTP requests received with the GET method.
httpgetsrate	<Double>	Read-only	Rate (/s) counter for httptotgets
httptotposts	<Double>	Read-only	Total number of HTTP requests received with the POST method.
httppostsrate	<Double>	Read-only	Rate (/s) counter for httptotposts
httptotothers	<Double>	Read-only	Total number of HTTP requests received with methods other than GET and POST. Some of the other well-defined HTTP methods are HEAD, PUT, DELETE, OPTIONS, and TRACE. User-defined methods are also allowed.
httptohersrate	<Double>	Read-only	Rate (/s) counter for httptotothers
httptot10requests	<Double>	Read-only	Total number of HTTP/1.0 requests received.

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
http10requestsrate	<Double>	Read-only	Rate (/s) counter for httptot10requests
httptot11requests	<Double>	Read-only	Total number of HTTP/1.1 requests received.
http11requestsrate	<Double>	Read-only	Rate (/s) counter for httptot11requests
httptotclenrequests	<Double>	Read-only	Total number of HTTP requests in which the Content-length field of the HTTP header has been set. Content-length specifies the length of the content, in bytes, in the associated HTTP body.
httpclenrequestsrate	<Double>	Read-only	Rate (/s) counter for httptotclenrequests
httptotchunkedrequests	<Double>	Read-only	Total number of HTTP requests in which the Transfer-Encoding field of the HTTP header has been set to chunked.
httpchunkedrequestsrate	<Double>	Read-only	Rate (/s) counter for httptotchunkedrequests
httptottxrequestbytes	<Double>	Read-only	Total number of bytes of HTTP request data transmitted.

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
httptxrequestbytesrate	<Double>	Read-only	Rate (/s) counter for httptottxrequestbytes
httptot10responses	<Double>	Read-only	Total number of HTTP/1.0 responses sent.
http10responsesrate	<Double>	Read-only	Rate (/s) counter for httptot10responses
httptot11responses	<Double>	Read-only	Total number of HTTP/1.1 responses sent.
http11responsesrate	<Double>	Read-only	Rate (/s) counter for httptot11responses
httptotclenresponses	<Double>	Read-only	Total number of HTTP responses sent in which the Content-length field of the HTTP header has been set. Content-length specifies the length of the content, in bytes, in the associated HTTP body.
httpclenresponsesrate	<Double>	Read-only	Rate (/s) counter for httptotclenresponses
httptotchunkedresponses	<Double>	Read-only	Total number of HTTP responses sent in which the Transfer-Encoding field of the HTTP header has been set to chunked. This setting is used when

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
			the server wants to start sending the response before knowing its total length. The server breaks the response into chunks and sends them in sequence, inserting the length of each chunk before the actual data. The message ends with a chunk of size zero.
httpchunkedrespons esrate	<Double>	Read-only	Rate (/s) counter for httptotchunkedrespo nses
httperrnoreusemultip art	<Double>	Read-only	Total number of HTTP multi-part responses sent. In multi-part responses, one or more entities are encapsulated within the body of a single message.
httperrnoreusemultip artrate	<Double>	Read-only	Rate (/s) counter for httperrnoreusemultip art
httptotnoclenchunkre sponses	<Double>	Read-only	Total number of FIN-terminated responses sent. In FIN-terminated responses, the server finishes sending the data and closes the connection.

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
httpnoclenchunkresponsesrate	<Double>	Read-only	Rate (/s) counter for httptotnoclenchunkresponses
httptottxresponsebytes	<Double>	Read-only	Total number of bytes of HTTP response data transmitted.
httptxresponsebytesrate	<Double>	Read-only	Rate (/s) counter for httptottxresponsebytes
httperrincompleteheaders	<Double>	Read-only	Total number of HTTP requests and responses received in which the HTTP header spans more than one packet.
httperrincompletereq uests	<Double>	Read-only	Total number of HTTP requests received in which the header spans more than one packet.
httperrincompletereq uestsrate	<Double>	Read-only	Rate (/s) counter for httperrincompletereq uests
httperrincompleteres ponses	<Double>	Read-only	Total number of HTTP responses received in which the header spans more than one packet.
httperrincompleteres ponsesrate	<Double>	Read-only	Rate (/s) counter for httperrincompleteres ponses

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
httperrserverbusy	<Double>	Read-only	Total number of HTTP error responses received. Some of the error responses are: 500 Internal Server Error 501 Not Implemented 502 Bad Gateway 503 Service Unavailable 504 Gateway Timeout 505 HTTP Version Not Supported
httperrserverbusyrate	<Double>	Read-only	Rate (/s) counter for httperrserverbusy
httperrlargecontent	<Double>	Read-only	Total number of requests and responses received with large body.
httperrlargechunk	<Double>	Read-only	Total number of requests received with large chunk size, in which the Transfer-Encoding field of the HTTP header has been set to chunked.
httperrlargectlen	<Double>	Read-only	Total number of requests received with large content, in which the Content-length field of the HTTP header has been set. Content-length specifies the length of the content,

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
			in bytes, in the associated HTTP body.
spdyv2totstreams	<Double>	Read-only	Total number of requests received over SPDYv2
spdyv2streamsrate	<Double>	Read-only	Rate (/s) counter for spdyv2totstreams
spdyv3totstreams	<Double>	Read-only	Total number of requests received over SPDYv3
spdyv3streamsrate	<Double>	Read-only	Rate (/s) counter for spdyv3totstreams

Operations

(click to see Properties)

GET (ALL)

Some options that you can use for each operations:

- **Getting warnings in response:**NITRO allows you to get warnings in an operation by specifying the "warning" query parameter as "yes". For example, to get warnings while connecting to the Citrix ADC appliance, the URL is as follows:

  http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/login?warning=yes

  If any, the warnings are displayed in the response payload with the HTTP code "209 X-NITRO-WARNING".

- **Authenticated access for individual NITRO operations:**NITRO allows you to logon to the Citrix ADC appliance to perform individual operations. You can use this option instead of creating a NITRO session (using the login object) and then using that session to perform all operations,

  To do this, you must specify the username and password in the request header of the NITRO request as follows:

X-NITRO-USER:<username>

X-NITRO-PASS:<password>

Note: In such cases, make sure that the request header DOES not include the following:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue>

`*Note:`*

Mandatory parameters are marked in redand placeholder content is marked in <green>.

get (all)

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/stat/protocolhttp

Query-parameters:

args

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/stat/protocolhttp?
args=detail:<Boolean_value>,fullvalues:<Boolean_value>,ntimes:<Double_value>,logfile:<String_value

Use this query-parameter to get protocolhttp resources based on additional properties.

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "protocolhttp": [ {

"httptotclenresponses":<Double_value>,
```

```
"httperrnoreusemultipart":<Double_value>,

"spdyv3streamsrate":<Double_value>,

"httprxrequestbytesrate":<Double_value>,

"httptotrxresponsebytes":<Double_value>,

"httptotchunkedresponses":<Double_value>,

"httperrincompleterequests":<Double_value>,

"httptot10requests":<Double_value>,

"httptotgets":<Double_value>,

"httperrlargecontent":<Double_value>,

"http10responsesrate":<Double_value>,

"httperrserverbusy":<Double_value>,

"httptxrequestbytesrate":<Double_value>,

"httptotposts":<Double_value>,

"httptotclenrequests":<Double_value>,

"httprxresponsebytesrate":<Double_value>,

"spdyv2streamsrate":<Double_value>,

"httperrincompleteresponsesrate":<Double_value>,

"httpnoclenchunkresponsesrate":<Double_value>,

"httpchunkedrequestsrate":<Double_value>,

"http11responsesrate":<Double_value>,

"httptotrequests":<Double_value>,

"httptotresponses":<Double_value>,
```

```
"httperrincompleteresponses":<Double_value>,

"spdytotstreams":<Double_value>,

"httptottxrequestbytes":<Double_value>,

"httperrserverbusyrate":<Double_value>,

"http11requestsrate":<Double_value>,

"spdystreamsrate":<Double_value>,

"httptxresponsebytesrate":<Double_value>,

"httptohersrate":<Double_value>,

"httpchunkedresponsesrate":<Double_value>,

"httptot11responses":<Double_value>,

"http10requestsrate":<Double_value>,

"httprequestsrate":<Double_value>,

"httpresponsesrate":<Double_value>,

"httptot10responses":<Double_value>,

"httptotnoclenchunkresponses":<Double_value>,

"httperrlargechunk":<Double_value>,

"httperrnoreusemultipartrate":<Double_value>,

"spdyv3totstreams":<Double_value>,

"httperrincompleteheaders":<Double_value>,

"httptot11requests":<Double_value>,

"httppostsrate":<Double_value>,

"httpgetsrate":<Double_value>,
```

```
"httpclenresponsesrate":<Double_value>,

"httperrlargectlen":<Double_value>,

"httptottxresponsebytes":<Double_value>,

"spdyv2totstreams":<Double_value>,

"httptotchunkedrequests":<Double_value>,

"httperrincompleterequestsrate":<Double_value>,

"httpclenrequestsrate":<Double_value>,

"httptotrxrequestbytes":<Double_value>,

"httptotothers":<Double_value>

}]}
```

protocolicmp

Statistics for icmp resource.

Properties

(click to see Operations)

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
clearstats	<String>	Read-write	Clear the statsistics / counters. Possible values = basic, full
icmptotrxpkts	<Double>	Read-only	ICMP packets received.
icmprxpktsrate	<Double>	Read-only	Rate (/s) counter for icmptotrxpkts

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
icmptotrxbytes	<Double>	Read-only	Bytes of ICMP data received.
icmprxbytesrate	<Double>	Read-only	Rate (/s) counter for icmptotrxbytes
icmptottxpkts	<Double>	Read-only	ICMP packets transmitted.
icmptxpktsrate	<Double>	Read-only	Rate (/s) counter for icmptottxpkts
icmptottxbytes	<Double>	Read-only	Bytes of ICMP data transmitted.
icmptxbytesrate	<Double>	Read-only	Rate (/s) counter for icmptottxbytes
icmptotrxechoreply	<Double>	Read-only	ICMP Ping echo replies received.
icmprxechoreplyrate	<Double>	Read-only	Rate (/s) counter for icmptotrxechoreply
icmptottxechoreply	<Double>	Read-only	ICMP Ping echo replies transmitted.
icmptxechoreplyrate	<Double>	Read-only	Rate (/s) counter for icmptottxechoreply
icmptotrxecho	<Double>	Read-only	ICMP Ping Echo Request and Echo Reply packets received.
icmprxechorate	<Double>	Read-only	Rate (/s) counter for icmptotrxecho
icmptotdstiplookup	<Double>	Read-only	Total number of MTU lookup on destination IP info received on a need fragmentation

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
			ICMP error message failed.
icmpcurratethreshold	<Double>	Read-only	Limit for ICMP packets handled every 10 milliseconds. Default value, 0, applies no limit. This is a configurable value using the set rateControl command.
icmptotportunreachablerx	<Double>	Read-only	ICMP Port Unreachable error messages received. This error is generated when there is no service is running on the port.
icmptotportunreachabletx	<Double>	Read-only	ICMP Port Unreachable error messages generated. This error is generated when there is no service is running on the port.
icmptotneedfragrx	<Double>	Read-only	ICMP Fragmentation Needed error messages received for packets that need to be fragmented but for which Don't Fragment is specified the header.
icmptotthresholdexceeds	<Double>	Read-only	Times the ICMP rate threshold is exceeded. If this

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
			counter continuously increases, first make sure the ICMP packets received are genuine. If they are, increase the current rate threshold.
icmptotpktsdropped	<Double>	Read-only	ICMP packets dropped because the rate threshold has been exceeded.
icmptotbadchecksum	<Double>	Read-only	ICMP Fragmentation Needed error messages received with an ICMP checksum error.
icmptotnonfirstipfrag	<Double>	Read-only	ICMP Fragmentation Needed error messages received that were generated by an IP fragment other than the first one.
icmptotinvalidbodylen	<Double>	Read-only	ICMP Fragmentation Needed error messages received that specified an invalid body length.
icmptotnotcpconn	<Double>	Read-only	ICMP Need Fragmentation error messages received for TCP packets. The state of the connection for these packets is not

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
			maintained on the Citrix ADC.
icmptotnoudpconn	<Double>	Read-only	ICMP Need Fragmentation error messages received for UDP packets. The state of the connection for these packets is not maintained on the Citrix ADC.
icmptotinvalidtcpseqno	<Double>	Read-only	ICMP Fragmentation Needed error messages received for packets that contain an invalid TCP address.
icmptotinvalidnextmtuval	<Double>	Read-only	ICMP Fragmentation Needed error messages received in which the Maximum Transmission Unit (MTU) for the next hop is out of range. The range for the MTU is 576-1500.
icmptotbignextmtu	<Double>	Read-only	ICMP Fragmentation Needed error messages received in which the value for the next MTU is higher than that of the current MTU.
icmptotinvalidprotocol	<Double>	Read-only	ICMP Fragmentation Needed error messages received

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
			that contain a protocol other than TCP and UDP.
icmptotbadpmtuipchecksum	<Double>	Read-only	ICMP Fragmentation Needed error messages received with an IP checksum error.
icmptotpmtunolink	<Double>	Read-only	ICMP Fragmentation Needed error messages received on a Protocol Control Block (PCB) with no link. The PCB maintains the state of the connection.
icmptotpmtudiscoverydisabled	<Double>	Read-only	ICMP Need Fragmentation error messages received when the PMTU Discovery mode is not enabled.

Operations

(click to see Properties)

GET (ALL)

Some options that you can use for each operations:

- **Getting warnings in response:**NITRO allows you to get warnings in an operation by specifying the "warning" query parameter as "yes". For example, to get warnings while connecting to the Citrix ADC appliance, the URL is as follows:

  http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/login?warning=yes

  If any, the warnings are displayed in the response payload with the HTTP code "209 X-NITRO-WARNING".

- **Authenticated access for individual NITRO operations:**NITRO allows you to logon to the Citrix ADC appliance to perform individual operations. You can use this option instead of creating a NITRO session (using the login object) and then using that session to perform all operations,

  To do this, you must specify the username and password in the request header of the NITRO request as follows:

  X-NITRO-USER:<username>

  X-NITRO-PASS:<password>

  Note: In such cases, make sure that the request header DOES not include the following:

  Cookie:NITRO_AUTH_TOKEN=<tokenvalue>

`*Note:`*

Mandatory parameters are marked in redand placeholder content is marked in <green>.

get (all)

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/stat/protocolicmp

Query-parameters:

args

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/stat/protocolicmp?
args=detail:<Boolean_value>,fullvalues:<Boolean_value>,ntimes:<Double_value>,logfile:<String_value

Use this query-parameter to get protocolicmp resources based on additional properties.

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:


```
{ "protocolicmp": [ {

"icmprxbytesrate":<Double_value>,

"icmptotthresholdexceeds":<Double_value>,

"icmprxechoreplyrate":<Double_value>,

"icmptottxechoreply":<Double_value>,

"icmptottxpkts":<Double_value>,

"icmptotinvalidnextmtuval":<Double_value>,

"icmptotrxecho":<Double_value>,

"icmptotdstiplookup":<Double_value>,

"icmptotnoudpconn":<Double_value>,

"icmptottxbytes":<Double_value>,

"icmptotrxbytes":<Double_value>,

"icmptotportunreachabletx":<Double_value>,

"icmpcurratethreshold":<Double_value>,

"icmptotbadchecksum":<Double_value>,

"icmptotinvalidbodylen":<Double_value>,

"icmprxechorate":<Double_value>,

"icmptotbignextmtu":<Double_value>,

"icmptotpmtudiscoverydisabled":<Double_value>,

"icmprxpktsrate":<Double_value>,
```

```
"icmptotbadpmtuipchecksum":<Double_value>,

"icmptotpmtunolink":<Double_value>,

"icmptotinvalidprotocol":<Double_value>,

"icmptotportunreachablerx":<Double_value>,

"icmptotneedfragrx":<Double_value>,

"icmptxechoreplyrate":<Double_value>,

"icmptotpktsdropped":<Double_value>,

"icmptotrxechoreply":<Double_value>,

"icmptxpktsrate":<Double_value>,

"icmptotnonfirstipfrag":<Double_value>,

"icmptotrxpkts":<Double_value>,

"icmptotnotcpconn":<Double_value>,

"icmptotinvalidtcpseqno":<Double_value>,

"icmptxbytesrate":<Double_value>

}]}
```

protocolicmpv6

Statistics for icmpv6 resource.

Properties

(click to see Operations)

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
clearstats	<String>	Read-write	Clear the statsistics / counters. Possible values = basic, full
icmpv6totrxpkts	<Double>	Read-only	ICMPv6 packets received.
icmpv6rxpktsrate	<Double>	Read-only	Rate (/s) counter for icmpv6totrxpkts
icmpv6totrxbytes	<Double>	Read-only	Bytes of ICMPv6 data received.
icmpv6rxbytesrate	<Double>	Read-only	Rate (/s) counter for icmpv6totrxbytes
icmpv6tottxpkts	<Double>	Read-only	ICMPv6 packets transmitted.
icmpv6txpktsrate	<Double>	Read-only	Rate (/s) counter for icmpv6tottxpkts
icmpv6tottxbytes	<Double>	Read-only	Bytes of ICMPv6 data transmitted.
icmpv6txbytesrate	<Double>	Read-only	Rate (/s) counter for icmpv6tottxbytes
icmpv6totrxna	<Double>	Read-only	ICMPv6 neighbor advertisement packets received. These packets are received in response to a neighbor solicitation message sent out by this node, or if the link layer address of a neighbor has changed.

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
icmpv6rxnarate	<Double>	Read-only	Rate (/s) counter for icmpv6totrxna
icmpv6totrxns	<Double>	Read-only	ICMPv6 neighbor solicitation packets received. These packets are received if the link layer address of a neighbor has changed, or in response to a neighbor solicitation message sent out by this node.
icmpv6rxnsrate	<Double>	Read-only	Rate (/s) counter for icmpv6totrxns
icmpv6totrxra	<Double>	Read-only	ICMPv6 router advertisement packets received. These are received at defined intervals or in response to a router solicitation message.
icmpv6rxrarate	<Double>	Read-only	Rate (/s) counter for icmpv6totrxra
icmpv6totrxrs	<Double>	Read-only	ICMPv6 router solicitation packets received. These could be sent by a neighboring router to initiate address resolution.
icmpv6rxrsrate	<Double>	Read-only	Rate (/s) counter for icmpv6totrxrs

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
icmpv6totrxechoreq	<Double>	Read-only	ICMPv6 Ping Echo Request packets received.
icmpv6rxechoreqrate	<Double>	Read-only	Rate (/s) counter for icmpv6totrxechoreq
icmpv6totrxechoreply	<Double>	Read-only	ICMPv6 Ping Echo Reply packets received.
icmpv6rxechoreplyrate	<Double>	Read-only	Rate (/s) counter for icmpv6totrxechoreply
icmpv6tottxna	<Double>	Read-only	ICMPv6 neighbor advertisement packets transmitted. These packets are sent in response to a neighbor solicitation packet, or if the link layer address of this node has changed.
icmpv6txnarate	<Double>	Read-only	Rate (/s) counter for icmpv6tottxna
icmpv6tottxns	<Double>	Read-only	ICMPv6 neighbor solicitation packets transmitted. These packets are sent to get the link layer addresses of neighboring nodes or to confirm that they are reachable.
icmpv6txnsrate	<Double>	Read-only	Rate (/s) counter for icmpv6tottxns

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
icmpv6tottxra	<Double>	Read-only	ICMPv6 router advertisement packets transmitted. These packets are sent at regular intervals or in response to a router solicitation packet from a neighbor.
icmpv6txrarate	<Double>	Read-only	Rate (/s) counter for icmpv6tottxra
icmpv6tottxrs	<Double>	Read-only	ICMPv6 router solicitation packets transmitted. These packets are sent to request neighboring routers to generate router advertisements immediately rather than wait for the next defined time.
icmpv6txrsrate	<Double>	Read-only	Rate (/s) counter for icmpv6tottxrs
icmpv6tottxechoreq	<Double>	Read-only	ICMPv6 Ping Echo Request packets transmitted.
icmpv6txechoreqrate	<Double>	Read-only	Rate (/s) counter for icmpv6tottxechoreq
icmpv6tottxechoreply	<Double>	Read-only	ICMP Ping Echo Reply packets transmitted.
icmpv6txechoreplyrate	<Double>	Read-only	Rate (/s) counter for icmpv6tottxechoreply

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
icmpv6errra	<Double>	Read-only	ICMPv6 router advertisement error packets received that contain an error in the header, such as an incorrect source IP address, destination IP address, or packet length.
icmpv6errna	<Double>	Read-only	ICMPv6 neighbor advertisement error packets received that contain an error in the header, such as an incorrect source IP address, destination IP address, or packet length.
icmpv6errns	<Double>	Read-only	ICMPv6 neighbor solicitation error packets received that contain an error in the header, such as an incorrect source IP address, destination IP address, or packet length.
icmpv6badchecksums	<Double>	Read-only	Packets received with an ICMPv6 checksum error.
icmpv6unspt	<Double>	Read-only	ICMPv6 packets received that are not supported by the Citrix ADC.

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
icmpv6rtthsld	<Double>	Read-only	Packets dropped because the default threshold of 100 requests per 10 milliseconds has been exceeded. This is a configurable value using the set rateControl command.

Operations

(click to see Properties)

GET (ALL)

Some options that you can use for each operations:

- **Getting warnings in response:**NITRO allows you to get warnings in an operation by specifying the "warning" query parameter as "yes". For example, to get warnings while connecting to the Citrix ADC appliance, the URL is as follows:

  http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/login?warning=yes

  If any, the warnings are displayed in the response payload with the HTTP code "209 X-NITRO-WARNING".

- **Authenticated access for individual NITRO operations:**NITRO allows you to logon to the Citrix ADC appliance to perform individual operations. You can use this option instead of creating a NITRO session (using the login object) and then using that session to perform all operations,

  To do this, you must specify the username and password in the request header of the NITRO request as follows:

  X-NITRO-USER:<username>

  X-NITRO-PASS:<password>

  Note: In such cases, make sure that the request header DOES not include the following:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue>

`*Note:` *

Mandatory parameters are marked in redand placeholder content is marked in <green>.

get (all)

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/stat/protocolicmpv6

Query-parameters:

args

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/stat/protocolicmpv6?
args=detail:<Boolean_value>,fullvalues:<Boolean_value>,ntimes:<Double_value>,logfile:<String_value

Use this query-parameter to get protocolicmpv6 resources based on additional properties.

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "protocolicmpv6": [ {

"icmpv6tottxns":<Double_value>,

"icmpv6txpktsrate":<Double_value>,

"icmpv6unspt":<Double_value>,

"icmpv6totrxechoreply":<Double_value>,
```

```
"icmpv6tottxra":<Double_value>,

"icmpv6tottxechoreply":<Double_value>,

"icmpv6badchecksums":<Double_value>,

"icmpv6txnarate":<Double_value>,

"icmpv6txrsrate":<Double_value>,

"icmpv6tottxna":<Double_value>,

"icmpv6totrxra":<Double_value>,

"icmpv6tottxrs":<Double_value>,

"icmpv6rxpktsrate":<Double_value>,

"icmpv6totrxpkts":<Double_value>,

"icmpv6totrxns":<Double_value>,

"icmpv6rxechoreplyrate":<Double_value>,

"icmpv6txechoreqrate":<Double_value>,

"icmpv6rxrsrate":<Double_value>,

"icmpv6errna":<Double_value>,

"icmpv6rxnsrate":<Double_value>,

"icmpv6rxbytesrate":<Double_value>,

"icmpv6rtthsld":<Double_value>,

"icmpv6totrxrs":<Double_value>,

"icmpv6totrxna":<Double_value>,

"icmpv6txechoreplyrate":<Double_value>,

"icmpv6totrxbytes":<Double_value>,
```

```
"icmpv6txrarate":<Double_value>,

"icmpv6errns":<Double_value>,

"icmpv6totrxechoreq":<Double_value>,

"icmpv6tottxechoreq":<Double_value>,

"icmpv6rxrarate":<Double_value>,

"icmpv6rxechoreqrate":<Double_value>,

"icmpv6tottxpkts":<Double_value>,

"icmpv6tottxbytes":<Double_value>,

"icmpv6rxnarate":<Double_value>,

"icmpv6txnsrate":<Double_value>,

"icmpv6errra":<Double_value>,

"icmpv6txbytesrate":<Double_value>

}]}
```

protocolip

Statistics for protocolip resource.

Properties

(click to see Operations)

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
clearstats	<String>	Read-write	Clear the statsistics / counters. Possible values = basic, full

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
iptotrxpkts	<Double>	Read-only	IP packets received.
iprxpktsrate	<Double>	Read-only	Rate (/s) counter for iptotrxpkts
iptotrxbytes	<Double>	Read-only	Bytes of IP data received.
iprxbytesrate	<Double>	Read-only	Rate (/s) counter for iptotrxbytes
iptottxpkts	<Double>	Read-only	IP packets transmitted.
iptxpktsrate	<Double>	Read-only	Rate (/s) counter for iptottxpkts
iptottxbytes	<Double>	Read-only	Bytes of IP data transmitted.
iptxbytesrate	<Double>	Read-only	Rate (/s) counter for iptottxbytes
iptotrxmbits	<Double>	Read-only	Megabits of IP data received.
iprxmbitsrate	<Double>	Read-only	Rate (/s) counter for iptotrxmbits
iptottxmbits	<Double>	Read-only	Megabits of IP data transmitted.
iptxmbitsrate	<Double>	Read-only	Rate (/s) counter for iptottxmbits
iptotroutedpkts	<Double>	Read-only	Total routed packets.
iproutedpktsrate	<Double>	Read-only	Rate (/s) counter for iptotroutedpkts
iptotroutedmbits	<Double>	Read-only	Total routed Mbits

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
iproutedmbitsrate	<Double>	Read-only	Rate (/s) counter for iptotroutedmbits
iptotfragments	<Double>	Read-only	IP fragments received.
iptotsuccreassembly	<Double>	Read-only	Fragmented IP packets successfully reassembled on the Citrix ADC.
iptotreassemblyattempt	<Double>	Read-only	IP packets that the Citrix ADC attempts to reassemble. If one of the fragments is missing, the whole packet is dropped.
iptotaddrlookup	<Double>	Read-only	IP address lookups performed by the Citrix ADC. When a packet is received on a non-established session, the Citrix ADC checks if the destination IP address is one of the Citrix ADC owned IP addresses.
iptotaddrlookupfail	<Double>	Read-only	IP address lookups performed by the Citrix ADC that have failed because the destination IP address of the packet does not match any of the Citrix ADC owned IP addresses.

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
iptotudpfragmentsfwd	<Double>	Read-only	UDP fragments forwarded to the client or the server.
iptottcpfragmentsfwd	<Double>	Read-only	TCP fragments forwarded to the client or the server.
iptotfragpktsgen	<Double>	Read-only	Fragmented packets created by the Citrix ADC.
iptotbadchecksums	<Double>	Read-only	Packets received with an IP checksum error.
iptotunsuccreassembly	<Double>	Read-only	Packets received that could not be reassembled. This can occur when there is a checksum failure, an identification field mismatch, or when one of the fragments is missing.
iptottoobig	<Double>	Read-only	Packets received for which the reassembled data exceeds the Ethernet packet data length of 1500 bytes.
iptotzerofragmentlen	<Double>	Read-only	Packets received with a fragment length of 0 bytes.
iptotdupfragments	<Double>	Read-only	Duplicate IP fragments received. This can occur when

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
			the acknowledgement was not received within the expected time.
iptotoutoforderfrag	<Double>	Read-only	Fragments received that are out of order.
iptotunknowndstrcvd	<Double>	Read-only	Packets received in which the destination IP address was not reachable or not owned by the Citrix ADC.
iptotbadtransport	<Double>	Read-only	Packets received in which the protocol specified in the IP header is unknown to the Citrix ADC.
iptotvipdown	<Double>	Read-only	Packets received for which the VIP is down. This can occur when all the services bound to the VIP are down or the VIP is manually disabled.
iptotfixheaderfail	<Double>	Read-only	Packets received that contain an error in one or more components of the IP header.
iptotttlexpired	<Double>	Read-only	Packets for which the time-to-live (TTL) expired during transit. These packets are dropped.

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
iptotmaxclients	<Double>	Read-only	Attempts to open a new connection to a service for which the maximum limit has been exceeded. Default value, 0, applies no limit.
iptotunknownsvcs	<Double>	Read-only	Packets received on a port or service that is not configured.
iptotlandattacks	<Double>	Read-only	Land-attack packets received. The source and destination addresses are the same.
iptotinvalidheadersz	<Double>	Read-only	Packets received in which an invalid data length is specified, or the value in the length field and the actual data length do not match. The range for the Ethernet packet data length is 0-1500 bytes.
iptotinvalidpacketsize	<Double>	Read-only	Total number of packets received by Citrix ADC with invalid IP packet size.
iptottruncatedpackets	<Double>	Read-only	Truncated IP packets received. An overflow in the routers along the

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
			path can truncate IP packets.
noniptottruncatedpackets	<Double>	Read-only	Truncated non-IP packets received.
iptotzeronexthop	<Double>	Read-only	Packets received that contain a 0 value in the next hop field. These packets are dropped.
iptotbadlens	<Double>	Read-only	Packets received with a length greater than the normal maximum transmission unit of 1514 bytes.
iptotbadmacaddrs	<Double>	Read-only	IP packets transmitted with a bad MAC address.

Operations

(click to see Properties)

GET (ALL)

Some options that you can use for each operations:

- **Getting warnings in response:**NITRO allows you to get warnings in an operation by specifying the "warning" query parameter as "yes". For example, to get warnings while connecting to the Citrix ADC appliance, the URL is as follows:

  http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/login?warning=yes

  If any, the warnings are displayed in the response payload with the HTTP code "209 X-NITRO-WARNING".

- **Authenticated access for individual NITRO operations:**NITRO allows you to logon to the Citrix ADC appliance to perform individual operations. You can use this option instead of

creating a NITRO session (using the login object) and then using that session to perform all operations,

To do this, you must specify the username and password in the request header of the NITRO request as follows:

X-NITRO-USER:<username>

X-NITRO-PASS:<password>

Note: In such cases, make sure that the request header DOES not include the following:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue>

`*Note:`*

Mandatory parameters are marked in redand placeholder content is marked in <green>.

get (all)

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/stat/protocolip

Query-parameters:

args

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/stat/protocolip?
args=detail:<Boolean_value>,fullvalues:<Boolean_value>,ntimes:<Double_value>,logfile:<String_value

Use this query-parameter to get protocolip resources based on additional properties.

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "protocolip": [ {
"iproutedpktsrate":<Double_value>,
"iptotoutoforderfrag":<Double_value>,
"noniptottruncatedpackets":<Double_value>,
"iptotmaxclients":<Double_value>,
"iptotzerofragmentlen":<Double_value>,
"iptxbytesrate":<Double_value>,
"iptotvipdown":<Double_value>,
"iptotroutedpkts":<Double_value>,
"iprxmbitsrate":<Double_value>,
"iprxpktsrate":<Double_value>,
"iptxmbitsrate":<Double_value>,
"iptotroutedmbits":<Double_value>,
"iptotinvalidheadersz":<Double_value>,
"iptotaddrlookupfail":<Double_value>,
"iptotbadtransport":<Double_value>,
"iptotfragpktsgen":<Double_value>,
"iptotsuccreassembly":<Double_value>,
"iptotrxmbits":<Double_value>,
"iptottxbytes":<Double_value>,
"iptotbadmacaddrs":<Double_value>,
```

```
"iptottcpfragmentsfwd":<Double_value>,

"iptotrxpkts":<Double_value>,

"iptotrxbytes":<Double_value>,

"iptotlandattacks":<Double_value>,

"iptotunknowndstrcvd":<Double_value>,

"iptottruncatedpackets":<Double_value>,

"iptotttlexpired":<Double_value>,

"iprxbytesrate":<Double_value>,

"iptottxmbits":<Double_value>,

"iptotbadlens":<Double_value>,

"iptotunknownsvcs":<Double_value>,

"iptotdupfragments":<Double_value>,

"iptottoobig":<Double_value>,

"iptotzeronexthop":<Double_value>,

"iptotaddrlookup":<Double_value>,

"iptotfragments":<Double_value>,

"iptotinvalidpacketsize":<Double_value>,

"iptotunsuccreassembly":<Double_value>,

"iptotreassemblyattempt":<Double_value>,

"iptottxpkts":<Double_value>,

"iptxpktsrate":<Double_value>,

"iptotbadchecksums":<Double_value>,
```

```
"iptotfixheaderfail":<Double_value>,

"iproutedmbitsrate":<Double_value>,

"iptotudpfragmentsfwd":<Double_value>

}]}
```

protocolipv6

Statistics for ip v6 resource.

Properties

(click to see Operations)

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
clearstats	<String>	Read-write	Clear the statsistics / counters. Possible values = basic, full
ipv6totrxpkts	<Double>	Read-only	IPv6 packets received.
ipv6rxpktsrate	<Double>	Read-only	Rate (/s) counter for ipv6totrxpkts
ipv6totrxbytes	<Double>	Read-only	Bytes of IPv6 data received.
ipv6rxbytesrate	<Double>	Read-only	Rate (/s) counter for ipv6totrxbytes
ipv6tottxpkts	<Double>	Read-only	IPv6 packets transmitted
ipv6txpktsrate	<Double>	Read-only	Rate (/s) counter for ipv6tottxpkts

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
ipv6tottxbytes	<Double>	Read-only	Bytes of IPv6 data transmitted.
ipv6txbytesrate	<Double>	Read-only	Rate (/s) counter for ipv6tottxbytes
ipv6totroutepkts	<Double>	Read-only	IPv6 packets routed.
ipv6routepktsrate	<Double>	Read-only	Rate (/s) counter for ipv6totroutepkts
ipv6totroutembits	<Double>	Read-only	IPv6 total Mbits.
ipv6routembitsrate	<Double>	Read-only	Rate (/s) counter for ipv6totroutembits
ipv6fragtotrxpkts	<Double>	Read-only	IPv6 fragments received.
ipv6fragrxpktsrate	<Double>	Read-only	Rate (/s) counter for ipv6fragtotrxpkts
ipv6fragtottcpreassembled	<Double>	Read-only	TCP fragments processed after reassembly.
ipv6fragtcpreassembledrate	<Double>	Read-only	Rate (/s) counter for ipv6fragtottcpreassembled
ipv6fragtotudpreassembled	<Double>	Read-only	UDP fragments processed after reassembly.
ipv6fragudpreassembledrate	<Double>	Read-only	Rate (/s) counter for ipv6fragtotudpreassembled
ipv6fragtotpktsprocessnoreass	<Double>	Read-only	IPv6 fragments processed without reassembly.

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
ipv6fragpktsprocessnoreassrate	<Double>	Read-only	Rate (/s) counter for ipv6fragtotpktsprocessnoreass
ipv6fragtotpktsforward	<Double>	Read-only	IPv6 fragments forwarded to the client or server without reassembly.
ipv6fragpktsforwardrate	<Double>	Read-only	Rate (/s) counter for ipv6fragtotpktsforward
ipv6errhdr	<Double>	Read-only	Packets received that contain an error in one or more components of the IPv6 header.
ipv6nextheadererr	<Double>	Read-only	Packets received that contain an unsupported next header. The supported next headers are TCP, ICMP, UDP, OSPF, and FRAGMENT.
ipv6landattack	<Double>	Read-only	Land-attack packets received. The source and destination addresses are the same. If not dropped, these packets can lock up the appliance.
ipv6fragpkttoobig	<Double>	Read-only	Packets received for which the reassembled data exceeds the Ethernet

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
			packet data length of 1500 bytes.
ipv6fragzerolenpkt	<Double>	Read-only	Packets received with a fragment length of 0 bytes.
ipv6toticmpnarx	<Double>	Read-only	Number of ICMPv6 NA packets received by Citrix ADC (OBSOLETE).
ipv6icmpnarxrate	<Double>	Read-only	Rate (/s) counter for ipv6toticmpnarx
ipv6toticmpnsrx	<Double>	Read-only	Number of ICMPv6 NS packets received by Citrix ADC (OBSOLETE).
ipv6icmpnsrxrate	<Double>	Read-only	Rate (/s) counter for ipv6toticmpnsrx
ipv6toticmpnatx	<Double>	Read-only	Number of ICMPv6 NA packets transmitted by Citrix ADC (OBSOLETE).
ipv6icmpnatxrate	<Double>	Read-only	Rate (/s) counter for ipv6toticmpnatx
ipv6toticmpnstx	<Double>	Read-only	Number of ICMPv6 NS packets transmitted by Citrix ADC (OBSOLETE).
ipv6icmpnstxrate	<Double>	Read-only	Rate (/s) counter for ipv6toticmpnstx
ipv6toticmprarx	<Double>	Read-only	Number of ICMPv6 RA packets received

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
			by Citrix ADC (OBSOLETE).
ipv6icmprarxrate	<Double>	Read-only	Rate (/s) counter for ipv6toticmprarx
ipv6toticmprstx	<Double>	Read-only	Number of ICMPv6 RS packets transmitted by Citrix ADC (OBSOLETE).
ipv6icmprstxrate	<Double>	Read-only	Rate (/s) counter for ipv6toticmprstx
ipv6toticmprx	<Double>	Read-only	Number of ICMPv6 packets received by Citrix ADC (OBSOLETE).
ipv6icmprxrate	<Double>	Read-only	Rate (/s) counter for ipv6toticmprx
ipv6toticmptx	<Double>	Read-only	Number of ICMPv6 packets transmitted by Citrix ADC (OBSOLETE).
ipv6icmptxrate	<Double>	Read-only	Rate (/s) counter for ipv6toticmptx
ipv6errrxhdr	<Double>	Read-only	Number of erroneous header packets received (OBSOLETE).
ipv6errrxpkt	<Double>	Read-only	Number of erroneous packets received (OBSOLETE).
ipv6badcksum	<Double>	Read-only	Number of bad checksum packets

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
			received (OBSOLETE).
icmpv6error	<Double>	Read-only	Number of erroneous ICMPv6 packets received (OBSOLETE).
icmpv6unspt	<Double>	Read-only	Number of ICMPv6 unsupported packets received (OBSOLETE).
icmpv6rtthsld	<Double>	Read-only	Number of ICMPv6 packets dropped for rate threshold exceeded (OBSOLETE).

Operations

(click to see Properties)

GET (ALL)

Some options that you can use for each operations:

- **Getting warnings in response:**NITRO allows you to get warnings in an operation by specifying the "warning" query parameter as "yes". For example, to get warnings while connecting to the Citrix ADC appliance, the URL is as follows:

  http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/login?warning=yes

  If any, the warnings are displayed in the response payload with the HTTP code "209 X-NITRO-WARNING".
- **Authenticated access for individual NITRO operations:**NITRO allows you to logon to the Citrix ADC appliance to perform individual operations. You can use this option instead of creating a NITRO session (using the login object) and then using that session to perform all operations,

To do this, you must specify the username and password in the request header of the NITRO request as follows:

X-NITRO-USER:<username>

X-NITRO-PASS:<password>

Note: In such cases, make sure that the request header DOES not include the following:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue>

`*Note:`*

Mandatory parameters are marked in redand placeholder content is marked in <green>.

get (all)

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/stat/protocolipv6

Query-parameters:

args

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/stat/protocolipv6?
args=detail:<Boolean_value>,fullvalues:<Boolean_value>,ntimes:<Double_value>,logfile:<String_value

Use this query-parameter to get protocolipv6 resources based on additional properties.

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "protocolipv6": [ {

"ipv6fragpktsforwardrate":<Double_value>,

"ipv6icmprstxrate":<Double_value>,

"ipv6fragpkttoobig":<Double_value>,

"ipv6fragtotrxpkts":<Double_value>,

"ipv6icmpnatxrate":<Double_value>,

"ipv6toticmpnsrx":<Double_value>,

"icmpv6unspt":<Double_value>,

"ipv6errrxhdr":<Double_value>,

"ipv6rxpktsrate":<Double_value>,

"ipv6icmpnstxrate":<Double_value>,

"ipv6errhdr":<Double_value>,

"ipv6rxbytesrate":<Double_value>,

"ipv6icmprxrate":<Double_value>,

"ipv6fragpktsprocessnoreassrate":<Double_value>,

"ipv6icmpnarxrate":<Double_value>,

"ipv6errrxpkt":<Double_value>,

"ipv6nextheadererr":<Double_value>,

"ipv6tottxbytes":<Double_value>,

"ipv6fragtotpktsforward":<Double_value>,

"ipv6totroutepkts":<Double_value>,
```

```
"ipv6toticmpnarx":<Double_value>,

"ipv6txpktsrate":<Double_value>,

"ipv6totrxbytes":<Double_value>,

"ipv6toticmpnatx":<Double_value>,

"ipv6toticmprstx":<Double_value>,

"ipv6txbytesrate":<Double_value>,

"icmpv6error":<Double_value>,

"ipv6fragzerolenpkt":<Double_value>,

"ipv6routembitsrate":<Double_value>,

"icmpv6rtthsld":<Double_value>,

"ipv6icmprarxrate":<Double_value>,

"ipv6toticmprx":<Double_value>,

"ipv6icmpnsrxrate":<Double_value>,

"ipv6routepktsrate":<Double_value>,

"ipv6icmptxrate":<Double_value>,

"ipv6fragtcpreassembledrate":<Double_value>,

"ipv6tottxpkts":<Double_value>,

"ipv6fragtotudpreassembled":<Double_value>,

"ipv6fragtottcpreassembled":<Double_value>,

"ipv6fragudpreassembledrate":<Double_value>,

"ipv6fragrxpktsrate":<Double_value>,

"ipv6badcksum":<Double_value>,
```

```
"ipv6totrxpkts":<Double_value>,

"ipv6fragtotpktsprocessnoreass":<Double_value>,

"ipv6totroutembits":<Double_value>,

"ipv6toticmptx":<Double_value>,

"ipv6toticmpnstx":<Double_value>,

"ipv6toticmprarx":<Double_value>,

"ipv6landattack":<Double_value>

}]}
```

protocolmptcp

Statistics for mptcp protocol resource.

Properties

(click to see Operations)

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
clearstats	<String>	Read-write	Clear the statsistics / counters. Possible values = basic, full
mptcptotmpcapsession	<Double>	Read-only	MPTCP total sessions created
mptcpmpcapsessionrate	<Double>	Read-only	Rate (/s) counter for mptcptotmpcapsession
mptcptotsfconn	<Double>	Read-only	MPTCP total Subflow connections created

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
mptcpsfconnrate	<Double>	Read-only	Rate (/s) counter for mptcptotsfconn
mptcpcurmpcapable sessions	<Double>	Read-only	The number of current mptcp sessions.
mptcpcursfconnectio ns	<Double>	Read-only	The number of current mptcp subflow connections.
mptcpcurpendingjoin	<Double>	Read-only	The number of current mptcp subflow connections in pending state.
mptcpcursesswithout sfs	<Double>	Read-only	Current Multipath TCP sessions without any subflows
mptcptotmpcapsyn	<Double>	Read-only	MPTCP total MP_CAPABLE received
mptcpmpcapsynrate	<Double>	Read-only	Rate (/s) counter for mptcptotmpcapsyn
mptcptotmpcapsteer ed	<Double>	Read-only	Total MP_CAPABLE sessions steered
mptcpmpcapsteeredr ate	<Double>	Read-only	Rate (/s) counter for mptcptotmpcapsteer ed
mptcptotconnest	<Double>	Read-only	Total MP_CAPABLE sessions created.
mptcpconnestrate	<Double>	Read-only	Rate (/s) counter for mptcptotconnest

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
mptcptotmpcapsynacksent	<Double>	Read-only	Total number of MP_CAPABLE SYN/ACKs sent.
mptcpmpcapsynacksentrate	<Double>	Read-only	Rate (/s) counter for mptcptotmpcapsynacksent
mptcptotmpcapfackrecvd	<Double>	Read-only	Total number of MP_CAPABLE Final ACKs received.
mptcpmpcapfackrecvdrate	<Double>	Read-only	Rate (/s) counter for mptcptotmpcapfackrecvd
mptcptotmpjoinsyn	<Double>	Read-only	MPTCP total MP_JOIN syn received
mptcpmpjoinsynrate	<Double>	Read-only	Rate (/s) counter for mptcptotmpjoinsyn
mptcptotmpjoinsteered	<Double>	Read-only	Total MP_JOIN subflows steered
mptcpmpjoinsteeredrate	<Double>	Read-only	Rate (/s) counter for mptcptotmpjoinsteered
mptcptotmpjoinsynacksent	<Double>	Read-only	Total MP_JOIN SYN/ACKs sent.
mptcpmpjoinsynacksentrate	<Double>	Read-only	Rate (/s) counter for mptcptotmpjoinsynacksent
mptcptotmpjoinfackrecvd	<Double>	Read-only	Total number of MP_JOIN Final ACKs

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
mptcpmpjoinfackrecvdrate	<Double>	Read-only	Rate (/s) counter for mptcptotmpjoinfackrecvd
mptcptotmpjoin4thacksent	<Double>	Read-only	Total number of Subflow final ACK from peer in 3 way handshake validated with 4th ACK.
mptcpmpjoin4thacksentrate	<Double>	Read-only	Rate (/s) counter for mptcptotmpjoin4thacksent
mptcptotestsfreplaced	<Double>	Read-only	MPTCP Total established subflows replaced due to new MP_JOIN.
mptcpestsfreplacedrate	<Double>	Read-only	Rate (/s) counter for mptcptotestsfreplaced
mptcptotpendsfreplaced	<Double>	Read-only	MPTCP Total pending subflows replaced due to new MP_JOIN.
mptcppendsfreplacedrate	<Double>	Read-only	Rate (/s) counter for mptcptotpendsfreplaced
mptcptotfreshackfrwd	<Double>	Read-only	Fresh ACK recieved on a subflow
mptcpfreshackfrwdrate	<Double>	Read-only	Rate (/s) counter for mptcptotfreshackfrwd
mptcpplainackfallback	<Double>	Read-only	MPTCP Fallback to regular tcp on

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
			receiving plain ACK for DSS.
mptcpplainackfallbackrate	<Double>	Read-only	Rate (/s) counter for mptcpplainackfallback
mptcpinfinitemaprecvd	<Double>	Read-only	MPTCP Received and set infinite map and fallen back to regular TCP.
mptcpinfinitemaprecvdrate	<Double>	Read-only	Rate (/s) counter for mptcpinfinitemaprecvd
mptcptotaddrremoved	<Double>	Read-only	Total number of addresses removed from MPTCP connection with REMOVE_ADDR option
mptcpaddrremovedrate	<Double>	Read-only	Rate (/s) counter for mptcptotaddrremoved
mptcptotdss	<Double>	Read-only	Total number of Data Sequence Signal packets.
mptcpdssrate	<Double>	Read-only	Rate (/s) counter for mptcptotdss
mptcptotrxdss	<Double>	Read-only	MPTCP Total Data Sequence Signal packets received.
mptcprxdssrate	<Double>	Read-only	Rate (/s) counter for mptcptotrxdss

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
mptcptottxdss	<Double>	Read-only	MMPTCP Total Data Sequence Signal packets sent
mptcptxdssrate	<Double>	Read-only	Rate (/s) counter for mptcptottxdss
mptcptotdssa	<Double>	Read-only	Total Data Sequence Signal packets during data transfer with DATA_ACK
mptcpdssarate	<Double>	Read-only	Rate (/s) counter for mptcptotdssa
mptcptotdssfreshack	<Double>	Read-only	MPTCP total Data Sequence Signal packets during data transfer with fresh ACK
mptcpdssfreshackrate	<Double>	Read-only	Rate (/s) counter for mptcptotdssfreshack
mptcptotdssm	<Double>	Read-only	MPTCP total data Sequence Signal packets with Data Sequence Mapping and checksum
mptcpdssmrate	<Double>	Read-only	Rate (/s) counter for mptcptotdssm
mptcptotinfinitemapfrwd	<Double>	Read-only	MPTCP received Data Sequence Signal with infinite map flag (Fallback to regular TCP).

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
mptcpinfinitemapfrwdrate	<Double>	Read-only	Rate (/s) counter for mptcptotinfinitemapfrwd
mptcptotdatalessthandatalen	<Double>	Read-only	MPTCP Map amount of data not yet received.
mptcpdatalessthandatalenrate	<Double>	Read-only	Rate (/s) counter for mptcptotdatalessthandatalen
mptcppriobackuprx	<Double>	Read-only	MPTCP Subflow used as backup path.
mptcppriobackuprxrate	<Double>	Read-only	Rate (/s) counter for mptcppriobackuprx
mptcpprioclearbackuprx	<Double>	Read-only	Subflow earlier used only as a backup subflow, changes to regular subflow with MP_PRIO option
mptcpprioclearbackuprxrate	<Double>	Read-only	Rate (/s) counter for mptcpprioclearbackuprx
mptcptottxdatafin	<Double>	Read-only	Total MPTCP connection close requests sent
mptcptxdatafinrate	<Double>	Read-only	Rate (/s) counter for mptcptottxdatafin
mptcptotrxdatafin	<Double>	Read-only	Total MPTCP connection close(DATA_FIN) requests received.

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
mptcprxdatafinrate	<Double>	Read-only	Rate (/s) counter for mptcptotrxdatafin
mptcptottxsffin	<Double>	Read-only	MPTCP total subflow close requests.
mptcptxsffinrate	<Double>	Read-only	Rate (/s) counter for mptcptottxsffin
mptcperrinvalcookie	<Double>	Read-only	MPTCP invalid cookie received on MP_CAPABLE final ACK.
mptcperrinvalcookierate	<Double>	Read-only	Rate (/s) counter for mptcperrinvalcookie
mptcperrextnflagset	<Double>	Read-only	Extension flag is set on MP_CAPABLE request.
mptcperrextnflagsetrate	<Double>	Read-only	Rate (/s) counter for mptcperrextnflagset
mptcperrresflagset	<Double>	Read-only	MPTCP One or more reserved bits are set on MP_CAPABLE request.
mptcperrresflagsetrate	<Double>	Read-only	Rate (/s) counter for mptcperrresflagset
mptcperrunknowntoken	<Double>	Read-only	MPTCP invalid token received on MP_JOIN request.
mptcperrunknowntokenrate	<Double>	Read-only	Rate (/s) counter for mptcperrunknowntoken

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
mptcperraddridexist	<Double>	Read-only	MPTCP MP_JOIN request on existing address id.
mptcperraddridexistrate	<Double>	Read-only	Rate (/s) counter for mptcperraddridexist
mptcperraddrid0	<Double>	Read-only	MPTCP MP_JOIN request on address id 0.
mptcperraddrid0rate	<Double>	Read-only	Rate (/s) counter for mptcperraddrid0
mptcperrmaxsf	<Double>	Read-only	MPTCP new MP_JOIN request after maximum configured subflows are established.
mptcperrmaxsfrate	<Double>	Read-only	Rate (/s) counter for mptcperrmaxsf
mptcperrjointhreshold	<Double>	Read-only	MPTCP Global pending MP_JOIN threshold limit is reached, new MP_JOIN request will be dropped sending RST
mptcperrjointhresholdrate	<Double>	Read-only	Rate (/s) counter for mptcperrjointhreshold
mptcperrjoinafterfallback	<Double>	Read-only	MPTCP New MP_JOIN request received after fallback to regular tcp.

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
mptcperrjoinafterfallbackrate	<Double>	Read-only	Rate (/s) counter for mptcperrjoinafterfallback
mptcperrinvalmac	<Double>	Read-only	MPTCP invalid MAC on MP_JOIN final ACK.
mptcperrinvalmacrate	<Double>	Read-only	Rate (/s) counter for mptcperrinvalmac
mptcperrinvalopts	<Double>	Read-only	MPTCP invalid mptcp option is received and is dropped.
mptcperrinvaloptsrate	<Double>	Read-only	Rate (/s) counter for mptcperrinvalopts
mptcperroptiondiscarded	<Double>	Read-only	Invalid subtype in MPTCP option field and hence discarded.
mptcperroptiondiscardedrate	<Double>	Read-only	Rate (/s) counter for mptcperroptiondiscarded
mptcperroptsnosession	<Double>	Read-only	MPTCP options sent on non existing connection/subflow PCBs.
mptcperroptsnosessionrate	<Double>	Read-only	Rate (/s) counter for mptcperroptsnosession
mptcperrinvalremaddr	<Double>	Read-only	MPTCP remove address request received on invalid/ unknown address id.

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
mptcperrinvalremaddrrate	<Double>	Read-only	Rate (/s) counter for mptcperrinvalremaddr
mptcperroptssendrst	<Double>	Read-only	MPTCP sent RST on receiving improper option field.
mptcperroptssendrstrate	<Double>	Read-only	Rate (/s) counter for mptcperroptssendrst
mptcperrremaddrself	<Double>	Read-only	MPTCP remove address request for self address.
mptcperrremaddrselfrate	<Double>	Read-only	Rate (/s) counter for mptcperrremaddrself
mptcperrrssffail	<Double>	Read-only	Add RSS filter to steer traffic to right node on established MPTCP session failed.
mptcperrrssffailrate	<Double>	Read-only	Rate (/s) counter for mptcperrrssffail
mptcperrnopayloadlenpkt	<Double>	Read-only	MPTCP Payload length not specified in packet.
mptcperrnopayloadlenpktrate	<Double>	Read-only	Rate (/s) counter for mptcperrnopayloadlenpkt
mptcperrunsupportedmssnegotiated	<Double>	Read-only	MPTCP Unsupported MSS negotiated error.

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
mptcperrunsupportedmssnegotiatedrate	<Double>	Read-only	Rate (/s) counter for mptcperrunsupportedmssnegotiated
mptcperrbadcksum	<Double>	Read-only	MPTCP checksum failed. Connection will fallback to regular tcp.
mptcperrbadcksumrate	<Double>	Read-only	Rate (/s) counter for mptcperrbadcksum
mptcperrcryptonotsupported	<Double>	Read-only	MPTCP client crypto algorithm not supported.
mptcperrcryptonotsupportedrate	<Double>	Read-only	Rate (/s) counter for mptcperrcryptonotsupported
mptcperrversionnotsupported	<Double>	Read-only	MPTCP MP_CAPABLE request from unsupported mptcp client.
mptcperrversionnotsupportedrate	<Double>	Read-only	Rate (/s) counter for mptcperrversionnotsupported
mptcpplainackrst	<Double>	Read-only	MPTCP Sent RST on receiving plain ACK for DSS.
mptcpplainackrstrate	<Double>	Read-only	Rate (/s) counter for mptcpplainackrst
mptcperrdatafinpassive	<Double>	Read-only	MPTCP Data FIN received on passive subflow

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
mptcperrdatafinpassiverate	<Double>	Read-only	Rate (/s) counter for mptcperrdatafinpassive
mptcperrfastclose	<Double>	Read-only	MPTCP FAST CLOSE sent.
mptcperrfastcloserate	<Double>	Read-only	Rate (/s) counter for mptcperrfastclose
mptcperrfastclosepassive	<Double>	Read-only	MPTCP Fast close received on passive subflow.
mptcperrfastclosepassiverate	<Double>	Read-only	Rate (/s) counter for mptcperrfastclosepassive
mptcperrfastclosekey	<Double>	Read-only	MPTCP FAST_CLOSE received with invalid key and the packet is dropped.
mptcperrfastclosekeyrate	<Double>	Read-only	Rate (/s) counter for mptcperrfastclosekey
mptcpmpfailsent	<Double>	Read-only	MPTCP Total MP_FAIL sent due to checksum failure.
mptcpmpfailsentrate	<Double>	Read-only	Rate (/s) counter for mptcpmpfailsent
mptcpmpfailrecvd	<Double>	Read-only	MPTCP Total MP_FAIL received and fallback to regular TCP.
mptcpmpfailrecvdrate	<Double>	Read-only	Rate (/s) counter for mptcpmpfailrecvd

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
mptcperrnomappktrcvd	<Double>	Read-only	MPTCP Packet received with no Data Sequence Mapping.
mptcperrnomappktrcvdrate	<Double>	Read-only	Rate (/s) counter for mptcperrnomappktrcvd
mptcptotmoredatarcvd	<Double>	Read-only	MPTCP More data received than the available Data Sequence Mapping.
mptcpmoredatarcvdrate	<Double>	Read-only	Rate (/s) counter for mptcptotmoredatarcvd
mptcperrbadmapconndrop	<Double>	Read-only	MPTCP Drop the session incase of invalid Data Sequence map.
mptcperrbadmapconndroprate	<Double>	Read-only	Rate (/s) counter for mptcperrbadmapconndrop
mptcperrdupmaprecvd	<Double>	Read-only	MPTCP Duplicate maps in Data Sequence map table.
mptcperrdupmaprecvdrate	<Double>	Read-only	Rate (/s) counter for mptcperrdupmaprecvd
mptcperrinvalidsfn	<Double>	Read-only	MPTCP subflow map doesn't exactly match MPTCP session mapping.

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
mptcperrinvalidsfnrate	<Double>	Read-only	Rate (/s) counter for mptcperrinvalidsfn
mptcperrmapexists	<Double>	Read-only	MPTCP sequence map already exists.
mptcperrmapexistsrate	<Double>	Read-only	Rate (/s) counter for mptcperrmapexists
mptcperrretxpktrcvd	<Double>	Read-only	Retransmitted Data Recevied on MPTCP session.
mptcperrretxpktrcvdrate	<Double>	Read-only	Rate (/s) counter for mptcperrretxpktrcvd
mptcperrsfsessionallocfail	<Double>	Read-only	Attaching the subflow to MPTCP session failed due to failure in allocating memory to subflow map table.
mptcperrsfsessionallocfailrate	<Double>	Read-only	Rate (/s) counter for mptcperrsfsessionallocfail
mptcperrmpcapsessionallocfail	<Double>	Read-only	Creating a MPTCP connection failed due to failure in allocating memory to MPTCP connection management structure.
mptcperrmpcapsessionallocfailrate	<Double>	Read-only	Rate (/s) counter for mptcperrmpcapsessionallocfail
mptcptotmpcapsfpcballoc	<Double>	Read-only	Allocating memory to TCP protocol control

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
			block(PCB) for subflow failed.
mptcpmpcapsfpcballocrate	<Double>	Read-only	Rate (/s) counter for mptcptotmpcapsfpcb alloc
mptcptotmpcballocfailed	<Double>	Read-only	Allocating memory to MPTCP protocol control block failed.
mptcpmpcballocfailedrate	<Double>	Read-only	Rate (/s) counter for mptcptotmpcballocfailed
mptcperrnsballocfailed	<Double>	Read-only	Failed to allocate memory to output MPTCP packet.
mptcperrnsballocfailedrate	<Double>	Read-only	Rate (/s) counter for mptcperrnsballocfailed
mptcperrnosffreensb	<Double>	Read-only	MPTCP output a packet without any subflow PCB.
mptcperrnosffreensbrate	<Double>	Read-only	Rate (/s) counter for mptcperrnosffreensb

Operations

(click to see Properties)

GET (ALL)

Some options that you can use for each operations:

- **Getting warnings in response:**NITRO allows you to get warnings in an operation by specifying the "warning" query parameter as "yes". For example, to get warnings while connecting to the Citrix ADC appliance, the URL is as follows:

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/login?warning=yes

If any, the warnings are displayed in the response payload with the HTTP code "209 X-NITRO-WARNING".

- **Authenticated access for individual NITRO operations:**NITRO allows you to logon to the Citrix ADC appliance to perform individual operations. You can use this option instead of creating a NITRO session (using the login object) and then using that session to perform all operations,

  To do this, you must specify the username and password in the request header of the NITRO request as follows:

  X-NITRO-USER:<username>

  X-NITRO-PASS:<password>

  Note: In such cases, make sure that the request header DOES not include the following:

  Cookie:NITRO_AUTH_TOKEN=<tokenvalue>

`*Note:`*

Mandatory parameters are marked in redand placeholder content is marked in <green>.

get (all)

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/stat/protocolmptcp

Query-parameters:

args

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/stat/protocolmptcp?
args=detail:<Boolean_value>,fullvalues:<Boolean_value>,ntimes:<Double_value>,logfile:<String_value

Use this query-parameter to get protocolmptcp resources based on additional properties.

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "protocolmptcp": [ {

"mptcperrinvalopts":<Double_value>,

"mptcperraddrid0rate":<Double_value>,

"mptcpplainackfallbackrate":<Double_value>,

"mptcperrbadmapconndroprate":<Double_value>,

"mptcperrrssffailrate":<Double_value>,

"mptcperrremaddrselfrate":<Double_value>,

"mptcpsfconnrate":<Double_value>,

"mptcperroptiondiscardedrate":<Double_value>,

"mptcptotmpjoinsynacksent":<Double_value>,

"mptcperrfastclose":<Double_value>,

"mptcperrremaddrself":<Double_value>,

"mptcperraddridexist":<Double_value>,

"mptcpfreshackfrwdrate":<Double_value>,

"mptcperrunsupportedmssnegotiatedrate":<Double_value>,

"mptcptotmpjoin4thacksent":<Double_value>,

"mptcperrinvalcookie":<Double_value>,
```

```
"mptcpmpjoinsteeredrate":<Double_value>,

"mptcptotaddrremoved":<Double_value>,

"mptcpplainackrstrate":<Double_value>,

"mptcperrinvalidsfnrate":<Double_value>,

"mptcpmpcapfackrecvdrate":<Double_value>,

"mptcperraddrid0":<Double_value>,

"mptcptotrxdss":<Double_value>,

"mptcperrfastclosekey":<Double_value>,

"mptcpmpfailsentrate":<Double_value>,

"mptcperrinvaloptsrate":<Double_value>,

"mptcpcurmpcapablesessions":<Double_value>,

"mptcprxdssrate":<Double_value>,

"mptcperrdatafinpassiverate":<Double_value>,

"mptcperrjoinafterfallbackrate":<Double_value>,

"mptcpconnestrate":<Double_value>,

"mptcperrcryptonotsupportedrate":<Double_value>,

"mptcperrmaxsf":<Double_value>,

"mptcperrinvalremaddr":<Double_value>,

"mptcperrinvalidsfn":<Double_value>,

"mptcptotmpcapsyn":<Double_value>,

"mptcpcursesswithoutsfs":<Double_value>,

"mptcpprioclearbackuprxrate":<Double_value>,
```

```
"mptcperrversionnotsupportedrate":<Double_value>,

"mptcpprioclearbackuprx":<Double_value>,

"mptcperrmapexistsrate":<Double_value>,

"mptcpmoredatarcvdrate":<Double_value>,

"mptcpmpfailsent":<Double_value>,

"mptcpmpfailrecvdrate":<Double_value>,

"mptcptotdssa":<Double_value>,

"mptcptotdss":<Double_value>,

"mptcperrfastclosekeyrate":<Double_value>,

"mptcperrrssffail":<Double_value>,

"mptcperrnsballocfailed":<Double_value>,

"mptcpmpjoinfackrecvdrate":<Double_value>,

"mptcpinfinitemaprecvdrate":<Double_value>,

"mptcptotmpcapsfpcballoc":<Double_value>,

"mptcperrunsupportedmssnegotiated":<Double_value>,

"mptcperrmapexists":<Double_value>,

"mptcpmpjoin4thacksentrate":<Double_value>,

"mptcperrnsballocfailedrate":<Double_value>,

"mptcppriobackuprx":<Double_value>,

"mptcpdssmrate":<Double_value>,

"mptcperroptssendrst":<Double_value>,

"mptcperrnomappktrcvdrate":<Double_value>,
```

```
"mptcpmpcballocfailedrate":<Double_value>,

"mptcpinfinitemaprecvd":<Double_value>,

"mptcperrnopayloadlenpktrate":<Double_value>,

"mptcpmpjoinsynacksentrate":<Double_value>,

"mptcperrsfsessionallocfail":<Double_value>,

"mptcperroptsnosessionrate":<Double_value>,

"mptcpcurpendingjoin":<Double_value>,

"mptcperrjoinafterfallback":<Double_value>,

"mptcperrinvalmacrate":<Double_value>,

"mptcperrretxpktrcvdrate":<Double_value>,

"mptcpdssarate":<Double_value>,

"mptcptotmpcapfackrecvd":<Double_value>,

"mptcpmpcapsessionrate":<Double_value>,

"mptcptotmpcapsynacksent":<Double_value>,

"mptcperrextnflagset":<Double_value>,

"mptcptotsfconn":<Double_value>,

"mptcptotconnest":<Double_value>,

"mptcptotmoredatarcvd":<Double_value>,

"mptcptotmpjoinsteered":<Double_value>,

"mptcppriobackuprxrate":<Double_value>,

"mptcprxdatafinrate":<Double_value>,

"mptcperrunknowntokenrate":<Double_value>,
```

```
"mptcperrfastcloserate":<Double_value>,

"mptcptotrxdatafin":<Double_value>,

"mptcpplainackrst":<Double_value>,

"mptcpaddrremovedrate":<Double_value>,

"mptcperrinvalcookierate":<Double_value>,

"mptcppendsfreplacedrate":<Double_value>,

"mptcperrfastclosepassiverate":<Double_value>,

"mptcperrnosffreensb":<Double_value>,

"mptcpdssrate":<Double_value>,

"mptcpmpcapsynrate":<Double_value>,

"mptcptxdssrate":<Double_value>,

"mptcperrbadmapconndrop":<Double_value>,

"mptcpmpcapsteeredrate":<Double_value>,

"mptcptxsffinrate":<Double_value>,

"mptcpmpfailrecvd":<Double_value>,

"mptcperrresflagsetrate":<Double_value>,

"mptcpmpcapsfpcballocrate":<Double_value>,

"mptcptotinfinitemapfrwd":<Double_value>,

"mptcptottxdss":<Double_value>,

"mptcperrdupmaprecvd":<Double_value>,

"mptcptotmpcapsteered":<Double_value>,

"mptcperrretxpktrcvd":<Double_value>,
```

```
"mptcperrnopayloadlenpkt":<Double_value>,

"mptcptotmpcapsession":<Double_value>,

"mptcpplainackfallback":<Double_value>,

"mptcpdatalessthandatalenrate":<Double_value>,

"mptcperrunknowntoken":<Double_value>,

"mptcptotmpcballocfailed":<Double_value>,

"mptcptotdatalessthandatalen":<Double_value>,

"mptcperrfastclosepassive":<Double_value>,

"mptcperrextnflagsetrate":<Double_value>,

"mptcperrinvalremaddrrate":<Double_value>,

"mptcptottxsffin":<Double_value>,

"mptcperrsfsessionallocfailrate":<Double_value>,

"mptcptotfreshackfrwd":<Double_value>,

"mptcperrjointhresholdrate":<Double_value>,

"mptcperrnosffreensbrate":<Double_value>,

"mptcperrresflagset":<Double_value>,

"mptcperrnomappktrcvd":<Double_value>,

"mptcpestsfreplacedrate":<Double_value>,

"mptcperraddridexistrate":<Double_value>,

"mptcptotmpjoinfackrecvd":<Double_value>,

"mptcpcursfconnections":<Double_value>,

"mptcpinfinitemapfrwdrate":<Double_value>,
```

```
"mptcperrjointhreshold":<Double_value>,

"mptcpmpcapsynacksentrate":<Double_value>,

"mptcperrinvalmac":<Double_value>,

"mptcptotpendsfreplaced":<Double_value>,

"mptcptotdssm":<Double_value>,

"mptcperrversionnotsupported":<Double_value>,

"mptcperrmpcapsessionallocfail":<Double_value>,

"mptcperrdupmaprecvdrate":<Double_value>,

"mptcperroptssendrstrate":<Double_value>,

"mptcpmpjoinsynrate":<Double_value>,

"mptcperroptsnosession":<Double_value>,

"mptcptottxdatafin":<Double_value>,

"mptcperrcryptonotsupported":<Double_value>,

"mptcptotestsfreplaced":<Double_value>,

"mptcperroptiondiscarded":<Double_value>,

"mptcperrbadcksumrate":<Double_value>,

"mptcperrmaxsfrate":<Double_value>,

"mptcptxdatafinrate":<Double_value>,

"mptcpdssfreshackrate":<Double_value>,

"mptcperrmpcapsessionallocfailrate":<Double_value>,

"mptcptotmpjoinsyn":<Double_value>,

"mptcperrbadcksum":<Double_value>,
```

```
"mptcperrdatafinpassive":<Double_value>,

"mptcptotdssfreshack":<Double_value>

}]}
```

protocoltcp

Statistics for tcp protocol resource.

Properties

(click to see Operations)

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
clearstats	<String>	Read-write	Clear the statsistics / counters. Possible values = basic, full
tcpactiveserverconn	<Double>	Read-only	Connections to a server currently responding to requests.
tcpcurserverconnopening	<Double>	Read-only	Server connections in the Opening state, which indicates that the handshakes are not yet complete.
tcpcurclientconnopening	<Double>	Read-only	Client connections in the Opening state, which indicates that the handshakes are not yet complete.
tcpcurclientconnestablished	<Double>	Read-only	Current client connections in the Established state,

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
			which indicates that data transfer can occur between the Citrix ADC and the client.
tcpcurserverconnestablished	<Double>	Read-only	Current server connections in the Established state, which indicates that data transfer can occur between the Citrix ADC and the server.
tcptotrxpkts	<Double>	Read-only	TCP packets received.
tcprxpktsrate	<Double>	Read-only	Rate (/s) counter for tcptotrxpkts
tcptotrxbytes	<Double>	Read-only	Bytes of TCP data received.
tcprxbytesrate	<Double>	Read-only	Rate (/s) counter for tcptotrxbytes
tcptottxpkts	<Double>	Read-only	TCP packets transmitted.
tcptxpktsrate	<Double>	Read-only	Rate (/s) counter for tcptottxpkts
tcptottxbytes	<Double>	Read-only	Bytes of TCP data transmitted.
tcptxbytesrate	<Double>	Read-only	Rate (/s) counter for tcptottxbytes
tcpcurclientconn	<Double>	Read-only	Client connections, including connections in the

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
			Opening, Established, and Closing state.
tcpcurclientconnclosing	<Double>	Read-only	Client connections in the Closing state, which indicates that the connection termination process has initiated but is not complete.
tcptotclientconnopened	<Double>	Read-only	Client connections opened by the Citrix ADC since startup (after three-way handshake). This counter is reset when the Citrix ADC is restarted.
tcpclientconnopenedrate	<Double>	Read-only	Rate (/s) counter for tcptotclientconnopened
tcpcurserverconn	<Double>	Read-only	Server connections, including connections in the Opening, Established, and Closing state.
tcpcurserverconnclosing	<Double>	Read-only	Server connections in the Closing state, which indicates that the connection termination process has initiated but is not complete.

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
tcptotserverconnopened	<Double>	Read-only	Server connections initiated by the Citrix ADC since startup. This counter is reset when the Citrix ADC is restarted.
tcpserverconnopenedrate	<Double>	Read-only	Rate (/s) counter for tcptotserverconnopened
tcpsurgequeuelen	<Double>	Read-only	Connections in the surge queue. When the Citrix ADC cannot open a connection to the server, for example when maximum connections have been reached, the Citrix ADC queues these requests.
tcpspareconn	<Double>	Read-only	Spare connections available. To save time and resources in establishing another connection for a new client, the connection on the server is not closed after completing the request from the first client and is available for serving future requests.
tcptotzombiecltconnflushed	<Double>	Read-only	Client connections that are flushed because the client

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
			has been idle for some time.
tcpzombiecltconnflushedrate	<Double>	Read-only	Rate (/s) counter for tcptotzombiecltconnflushed
tcptotzombiehalfopencltconnflushed	<Double>	Read-only	Half-opened client connections that are flushed because the three-way handshakes are not complete.
tcpzombiehalfopencltconnflushedrate	<Double>	Read-only	Rate (/s) counter for tcptotzombiehalfopencltconnflushed
tcptotzombieactivehalfclosecltconnflushed	<Double>	Read-only	Active half-closed client connections that are flushed because the client has closed the connection and there has been no activity on the connection.
tcpzombieactivehalfclosecltconnflushedrate	<Double>	Read-only	Rate (/s) counter for tcptotzombieactivehalfclosecltconnflushed
tcptotzombiepassivehalfclosecltconnflushed	<Double>	Read-only	Passive half-closed client connections that are flushed because the Citrix ADC has closed the connection and there has been no activity on the connection.

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
tcpzombiepassivehalfclosecltconnflushedrate	<Double>	Read-only	Rate (/s) counter for tcptotzombiepassivehalfclosecltconnflushed
tcptotzombiesvrconnflushed	<Double>	Read-only	Server connections that are flushed because there have been no client requests in the queue for some time.
tcpzombiesvrconnflushedrate	<Double>	Read-only	Rate (/s) counter for tcptotzombiesvrconnflushed
tcptotzombiehalfopensvrconnflushed	<Double>	Read-only	Half-opened server connections that are flushed because the three-way handshakes are not complete.
tcpzombiehalfopensvrconnflushedrate	<Double>	Read-only	Rate (/s) counter for tcptotzombiehalfopensvrconnflushed
tcptotzombieactivehalfclosesvrconnflushed	<Double>	Read-only	Active half-closed server connections that are flushed because the server has closed the connection and there has been no activity on the connection.
tcpzombieactivehalfclosesvrconnflushedrate	<Double>	Read-only	Rate (/s) counter for tcptotzombieactivehalfclosesvrconnflushed

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
tcptotzombiepassivehalfclosesrvconnflushed	<Double>	Read-only	Passive half-closed server connections that are flushed because the Citrix ADC has closed the connection and there has been no activity on the connection.
tcpzombiepassivehalfclosesrvconnflushedrate	<Double>	Read-only	Rate (/s) counter for tcptotzombiepassivehalfclosesrvconnflushed
pcbtotzombiecall	<Double>	Read-only	Times the Zombie cleanup function is called. Every time a connection is flushed, it is marked for cleanup. The Zombie cleanup function clears all these connections at predefined intervals.
pcbzombiecallrate	<Double>	Read-only	Rate (/s) counter for pcbtotzombiecall
tcptotsyn	<Double>	Read-only	SYN packets received
tcpsynrate	<Double>	Read-only	Rate (/s) counter for tcptotsyn
tcptotsynprobe	<Double>	Read-only	Probes from the Citrix ADC to a server. The Citrix ADC sends a SYN packet to the server to check its availability and

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
			expects a SYN_ACK packet from the server before a specified response timeout.
tcpsynproberate	<Double>	Read-only	Rate (/s) counter for tcptotsynprobe
tcptotsvrfin	<Double>	Read-only	FIN packets received from the server.
tcpsvrfinrate	<Double>	Read-only	Rate (/s) counter for tcptotsvrfin
tcptotcltfin	<Double>	Read-only	FIN packets received from the clients.
tcpcltfinrate	<Double>	Read-only	Rate (/s) counter for tcptotcltfin
tcpwaittosyn	<Double>	Read-only	SYN packets (packets used to initiate a TCP connection) received on connections that are in the TIME_WAIT state. Packets cannot be transferred on a connection in this state.
tcpwaittosynrate	<Double>	Read-only	Rate (/s) counter for tcpwaittosyn
tcpwaittodata	<Double>	Read-only	Bytes of data received on connections that are in the TIME_WAIT state. Data cannot be transferred on a

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
			connection that is in this state.
tcpwaittodatarate	<Double>	Read-only	Rate (/s) counter for tcpwaittodata
tcptotsynheld	<Double>	Read-only	SYN packets held on the Citrix ADC that are waiting for a server connection.
tcpsynheldrate	<Double>	Read-only	Rate (/s) counter for tcptotsynheld
tcptotsynflush	<Double>	Read-only	SYN packets flushed on the Citrix ADC because of no response from the server for three or more seconds.
tcpsynflushrate	<Double>	Read-only	Rate (/s) counter for tcptotsynflush
tcptotfinwaitclosed	<Double>	Read-only	Connections closed on the Citrix ADC because the number of connections in the TIME_WAIT state has exceeded the default value of 7000.
tcpfinwaitclosedrate	<Double>	Read-only	Rate (/s) counter for tcptotfinwaitclosed
tcperrbadchecksum	<Double>	Read-only	Packets received with a TCP checksum error.
tcperrbadchecksumrate	<Double>	Read-only	Rate (/s) counter for tcperrbadchecksum

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
tcperrdataafterfin	<Double>	Read-only	Packets received following a connection termination request. This error is usually caused by a reordering of packets during transmission.
tcperrdataafterfinrate	<Double>	Read-only	Rate (/s) counter for tcperrdataafterfin
tcperrsyninsynrcvd	<Double>	Read-only	SYN packets received on a connection that is in the SYN_RCVD state. A connection goes into the SYN_RCVD state after receiving a SYN packet.
tcperrsyninsynrcvdrate	<Double>	Read-only	Rate (/s) counter for tcperrsyninsynrcvd
tcperrsyninest	<Double>	Read-only	SYN packets received on a connection that is in the ESTABLISHED state. A SYN packet is not expected on an ESTABLISHED connection.
tcperrsyninestrate	<Double>	Read-only	Rate (/s) counter for tcperrsyninest
tcperrsynsentbadack	<Double>	Read-only	Incorrect ACK packets received on a connection that is in the SYN_SENT

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
			state. An incorrect ACK packet is the third packet in the three-way handshake that has an incorrect sequence number.
tcperrsynsentbadack rate	<Double>	Read-only	Rate (/s) counter for tcperrsynsentbadack
tcperrrst	<Double>	Read-only	Reset packets received from a client or a server.
tcperrrstrate	<Double>	Read-only	Rate (/s) counter for tcperrrst
tcperrrstnonest	<Double>	Read-only	Reset packets received on a connection that is not in the ESTABLISHED state.
tcperrrstnonestrate	<Double>	Read-only	Rate (/s) counter for tcperrrstnonest
tcperrrstoutofwindow	<Double>	Read-only	Reset packets received on a connection that is out of the current TCP window.
tcperrrstoutofwindow rate	<Double>	Read-only	Rate (/s) counter for tcperrrstoutofwindow
tcperrrstintimewait	<Double>	Read-only	Reset packets received on a connection that is in the TIME_WAIT state. Packets

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
			cannot be transferred on a connection in the TIME_WAIT state.
tcperrrstintimewaitrate	<Double>	Read-only	Rate (/s) counter for tcperrrstintimewait
tcperrsvroutoforder	<Double>	Read-only	Out of order TCP packets received from a server.
tcperrsvroutoforderrate	<Double>	Read-only	Rate (/s) counter for tcperrsvroutoforder
tcperrcltoutoforder	<Double>	Read-only	Out of order TCP packets received from a client.
tcperrcltoutoforderrate	<Double>	Read-only	Rate (/s) counter for tcperrcltoutoforder
tcperrclthole	<Double>	Read-only	TCP holes created on a client connection. When out of order packets are received from a client, a hole is created on the Citrix ADC for each group of missing packets.
tcperrcltholerate	<Double>	Read-only	Rate (/s) counter for tcperrclthole
tcperrsvrhole	<Double>	Read-only	TCP holes created on a server connection. When out of order packets are received from a server, a hole is created on the Citrix

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
			ADC for each group of missing packets.
tcperrsvrholerate	<Double>	Read-only	Rate (/s) counter for tcperrsvrhole
tcperrcookiepktseqreject	<Double>	Read-only	SYN cookie packets rejected because they contain an incorrect sequence number.
tcperrcookiepktseqrejectrate	<Double>	Read-only	Rate (/s) counter for tcperrcookiepktseqreject
tcperrcookiepktsigreject	<Double>	Read-only	SYN cookie packets rejected because they contain an incorrect signature.
tcperrcookiepktsigrejectrate	<Double>	Read-only	Rate (/s) counter for tcperrcookiepktsigreject
tcperrcookiepktseqdrop	<Double>	Read-only	SYN cookie packets dropped because the sequence number specified in the packets is outside the current window.
tcperrcookiepktseqdroprate	<Double>	Read-only	Rate (/s) counter for tcperrcookiepktseqdrop
tcperrcookiepktmssreject	<Double>	Read-only	SYN cookie packets rejected because the maximum segment size (MSS) specified

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
			in the packets is incorrect.
tcperrcookiepktmssrejectrate	<Double>	Read-only	Rate (/s) counter for tcperrcookiepktmssreject
tcperranyportfail	<Double>	Read-only	Port allocations that have failed on a mapped IP address because the maximum limit of 65536 has been exceeded.
tcperranyportfailrate	<Double>	Read-only	Rate (/s) counter for tcperranyportfail
tcperripportfail	<Double>	Read-only	Port allocations that have failed on a subnet IP address or vserver IP address because the maximum limit of 65536 has been exceeded.
tcperripportfailrate	<Double>	Read-only	Rate (/s) counter for tcperripportfail
tcperrstraypkt	<Double>	Read-only	Number of stray or misrouted packets.
tcperrstraypktrate	<Double>	Read-only	Rate (/s) counter for tcperrstraypkt
tcperrsentrst	<Double>	Read-only	Reset packets sent to a client or a server.
tcperrsentrstrate	<Double>	Read-only	Rate (/s) counter for tcperrsentrst

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
tcperrbadstateconn	<Double>	Read-only	Connections that are not in a valid TCP state.
tcperrbadstateconnrate	<Double>	Read-only	Rate (/s) counter for tcperrbadstateconn
tcperrrstthreshold	<Double>	Read-only	Reset packets dropped because the default threshold of 100 resets per 10 milliseconds has been exceeded. This is a configurable value using the set rateControl command.
tcperrrstthresholdrate	<Double>	Read-only	Rate (/s) counter for tcperrrstthreshold
tcperroutofwindowpkts	<Double>	Read-only	Packets received that are out of the current advertised window.
tcperroutofwindowpktsrate	<Double>	Read-only	Rate (/s) counter for tcperroutofwindowpkts
tcperrsyndroppedcongestion	<Double>	Read-only	SYN packets dropped because of network congestion.
tcperrsyndroppedcongestionrate	<Double>	Read-only	Rate (/s) counter for tcperrsyndroppedcongestion
tcperrcltretrasmit	<Double>	Read-only	Packets retransmitted by a client. This usually

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
			occurs because the acknowledgement from the Citrix ADC has not reached the client.
tcperrcltretrasmitrate	<Double>	Read-only	Rate (/s) counter for tcperrcltretrasmit
tcperrfullretrasmit	<Double>	Read-only	Full packets retransmitted by the client or the server.
tcperrfullretrasmitrate	<Double>	Read-only	Rate (/s) counter for tcperrfullretrasmit
tcperrsynretry	<Double>	Read-only	SYN packets resent to a server.
tcperrsynretryrate	<Double>	Read-only	Rate (/s) counter for tcperrsynretry
tcperrsyngiveup	<Double>	Read-only	Attempts to establish a connection on the Citrix ADC that timed out.
tcperrsyngiveuprate	<Double>	Read-only	Rate (/s) counter for tcperrsyngiveup
tcperrretransmit	<Double>	Read-only	TCP packets retransmitted. The Citrix ADC attempts to retransmit the packet up to seven times, after which it resets the other half of the TCP connection.
tcperrretransmitrate	<Double>	Read-only	Rate (/s) counter for tcperrretransmit

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
tcperrfirstretransmissions	<Double>	Read-only	Packets retransmitted once by the Citrix ADC.
tcperrfirstretransmissionsrate	<Double>	Read-only	Rate (/s) counter for tcperrfirstretransmissions
tcperrthirdretransmissions	<Double>	Read-only	Packets retransmitted three times by the Citrix ADC.
tcperrthirdretransmissionsrate	<Double>	Read-only	Rate (/s) counter for tcperrthirdretransmissions
tcperrfifthretransmissions	<Double>	Read-only	Packets retransmitted five times by the Citrix ADC.
tcperrfifthretransmissionsrate	<Double>	Read-only	Rate (/s) counter for tcperrfifthretransmissions
tcperrseventhretransmissions	<Double>	Read-only	Packets retransmitted seven times by the Citrix ADC. If this fails, the Citrix ADC terminates the connection.
tcperrseventhretransmissionsrate	<Double>	Read-only	Rate (/s) counter for tcperrseventhretransmissions
tcperrfastretransmissions	<Double>	Read-only	TCP packets on which the Citrix ADC performs a fast

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
			retransmission in response to three duplicate acknowledgements or a partial acknowledgement. The Citrix ADC assumes that the packet is lost and retransmits the packet before its time-out.
tcperrfastretransmissionsrate	<Double>	Read-only	Rate (/s) counter for tcperrfastretransmissions
tcperrsvrretrasmit	<Double>	Read-only	Packets retransmitted by a server. This usually occurs because the acknowledgement from the Citrix ADC has not reached the server.
tcperrsvrretrasmitrate	<Double>	Read-only	Rate (/s) counter for tcperrsvrretrasmit
tcperrpartialretrasmit	<Double>	Read-only	Partial packet retransmits by a client or server due to congestion on the connection. This usually occurs because the window advertised by the Citrix ADC is not big enough to hold the full packet.

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
tcperrpartialretrasmit rate	<Double>	Read-only	Rate (/s) counter for tcperrpartialretrasmit
tcperrfinretry	<Double>	Read-only	FIN packets resent to a server or a client.
tcperrfinretryrate	<Double>	Read-only	Rate (/s) counter for tcperrfinretry
tcperrfingiveup	<Double>	Read-only	Connections that were timed out by the Citrix ADC because of not receiving the ACK packet after retransmitting the FIN packet four times.
tcperrfingiveuprate	<Double>	Read-only	Rate (/s) counter for tcperrfingiveup
tcperrsecondretrans missions	<Double>	Read-only	Packets retransmitted twice by the Citrix ADC.
tcperrsecondretrans missionsrate	<Double>	Read-only	Rate (/s) counter for tcperrsecondretrans missions
tcperrforthretransmis sions	<Double>	Read-only	Packets retransmitted four times by the Citrix ADC.
tcperrforthretransmis sionsrate	<Double>	Read-only	Rate (/s) counter for tcperrforthretransmis sions

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
tcperrsixthretransmissions	<Double>	Read-only	Packets retransmitted six times by the Citrix ADC.
tcperrsixthretransmissionsrate	<Double>	Read-only	Rate (/s) counter for tcperrsixthretransmissions
tcperrretransmitgiveup	<Double>	Read-only	Number of times Citrix ADC terminates a connection after retransmitting the packet seven times on that connection.Retrasnmission happens when recieving end doesn't acknowledges the packet.
tcperrretransmitgiveuprate	<Double>	Read-only	Rate (/s) counter for tcperrretransmitgiveup
tcperrcipalloc	<Double>	Read-only	Number of times TCP level client header insertion failure
tcperrcipallocrate	<Double>	Read-only	Rate (/s) counter for tcperrcipalloc

Operations

(click to see Properties)

GET (ALL)

Some options that you can use for each operations:

- **Getting warnings in response:**NITRO allows you to get warnings in an operation by specifying the "warning" query parameter as "yes". For example, to get warnings while connecting to the Citrix ADC appliance, the URL is as follows:

  http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/login?warning=yes

  If any, the warnings are displayed in the response payload with the HTTP code "209 X-NITRO-WARNING".

- **Authenticated access for individual NITRO operations:**NITRO allows you to logon to the Citrix ADC appliance to perform individual operations. You can use this option instead of creating a NITRO session (using the login object) and then using that session to perform all operations,

  To do this, you must specify the username and password in the request header of the NITRO request as follows:

  X-NITRO-USER:<username>

  X-NITRO-PASS:<password>

  Note: In such cases, make sure that the request header DOES not include the following:

  Cookie:NITRO_AUTH_TOKEN=<tokenvalue>

`*Note:`*

Mandatory parameters are marked in redand placeholder content is marked in <green>.

get (all)

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/stat/protocoltcp

Query-parameters:

args

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/stat/protocoltcp?
args=detail:<Boolean_value>,fullvalues:<Boolean_value>,ntimes:<Double_value>,logfile:<String_value

Use this query-parameter to get protocoltcp resources based on additional properties.

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "protocoltcp": [ {

"tcperrfingiveuprate":<Double_value>,

"tcperrretransmit":<Double_value>,

"tcperrcookiepktmssreject":<Double_value>,

"tcptotzombiehalfopencltconnflushed":<Double_value>,

"tcperrsentrstrate":<Double_value>,

"tcperrcltoutoforder":<Double_value>,

"tcpsurgequeuelen":<Double_value>,

"tcperrcookiepktseqrejectrate":<Double_value>,

"tcperrrstoutofwindowrate":<Double_value>,

"tcptotrxpkts":<Double_value>,

"tcpcurclientconnopening":<Double_value>,

"tcperrcookiepktseqreject":<Double_value>,

"tcperrforthretransmissionsrate":<Double_value>,

"tcperrrst":<Double_value>,
```

```
"tcpsvrfinrate":<Double_value>,

"tcptxbytesrate":<Double_value>,

"tcptotsyn":<Double_value>,

"tcpserverconnopenedrate":<Double_value>,

"tcperrsyninsynrcvdrate":<Double_value>,

"tcperrpartialretrasmitrate":<Double_value>,

"tcpzombieactivehalfclosecltconnflushedrate":<Double_value>,

"tcperrfinretry":<Double_value>,

"tcperrsvroutoforderrate":<Double_value>,

"pcbzombiecallrate":<Double_value>,

"tcperrfirstretransmissionsrate":<Double_value>,

"tcperrstraypkt":<Double_value>,

"tcperrsyninsynrcvd":<Double_value>,

"tcprxbytesrate":<Double_value>,

"tcpcurserverconnestablished":<Double_value>,

"tcperroutofwindowpktsrate":<Double_value>,

"tcperranyportfailrate":<Double_value>,

"tcpsynproberate":<Double_value>,

"tcpzombiesvrconnflushedrate":<Double_value>,

"tcpcurclientconn":<Double_value>,

"tcperrsyninestrate":<Double_value>,

"tcpcurserverconn":<Double_value>,
```

```
"tcperripportfailrate":<Double_value>,

"tcperrcookiepktsigrejectrate":<Double_value>,

"tcperrforthretransmissions":<Double_value>,

"tcpzombiehalfopensvrconnflushedrate":<Double_value>,

"tcperrsixthretransmissions":<Double_value>,

"tcperrbadchecksum":<Double_value>,

"tcpfinwaitclosedrate":<Double_value>,

"tcpsynflushrate":<Double_value>,

"tcperrcipallocrate":<Double_value>,

"tcperrrstthreshold":<Double_value>,

"tcptotzombieactivehalfclosecltconnflushed":<Double_value>,

"tcperrsynsentbadackrate":<Double_value>,

"tcpwaittodatarate":<Double_value>,

"tcperrseventhretransmissionsrate":<Double_value>,

"tcperrretransmitrate":<Double_value>,

"tcperrfullretrasmitrate":<Double_value>,

"tcptotsynflush":<Double_value>,

"tcperrsixthretransmissionsrate":<Double_value>,

"tcperrfastretransmissions":<Double_value>,

"tcperrsecondretransmissionsrate":<Double_value>,

"tcpsynheldrate":<Double_value>,

"tcpcltfinrate":<Double_value>,
```

```
"tcpzombiepassivehalfclosecltconnflushedrate":<Double_value>,

"tcperrrstnonestrate":<Double_value>,

"tcperrcltretrasmit":<Double_value>,

"tcptotsynprobe":<Double_value>,

"tcptotfinwaitclosed":<Double_value>,

"tcpzombiepassivehalfclosesrvconnflushedrate":<Double_value>,

"tcperrsynsentbadack":<Double_value>,

"tcptotsynheld":<Double_value>,

"tcperrrstintimewaitrate":<Double_value>,

"tcperrrstintimewait":<Double_value>,

"tcperrthirdretransmissions":<Double_value>,

"tcptotrxbytes":<Double_value>,

"tcperrsvrretrasmit":<Double_value>,

"tcperrfastretransmissionsrate":<Double_value>,

"tcptotzombiepassivehalfclosecltconnflushed":<Double_value>,

"tcperrcltholerate":<Double_value>,

"tcperrsvroutoforder":<Double_value>,

"tcperrstraypktrate":<Double_value>,

"tcptotclientconnopened":<Double_value>,

"tcprxpktsrate":<Double_value>,

"tcperrfinretryrate":<Double_value>,

"tcptotserverconnopened":<Double_value>,
```

```
"tcperrretransmitgiveup":<Double_value>,

"tcperrthirdretransmissionsrate":<Double_value>,

"tcperrsyndroppedcongestion":<Double_value>,

"tcpwaittosyn":<Double_value>,

"tcperrfingiveup":<Double_value>,

"tcptotsvrfin":<Double_value>,

"tcperrcltretrasmitrate":<Double_value>,

"tcperrcookiepktmssrejectrate":<Double_value>,

"tcperrsvrretrasmitrate":<Double_value>,

"tcpcurserverconnclosing":<Double_value>,

"tcperrrstrate":<Double_value>,

"tcperrfifthretransmissionsrate":<Double_value>,

"tcperrsvrhole":<Double_value>,

"tcperrdataafterfin":<Double_value>,

"tcperrfirstretransmissions":<Double_value>,

"tcperrsyngiveup":<Double_value>,

"tcptotzombiesvrconnflushed":<Double_value>,

"tcperrsynretryrate":<Double_value>,

"tcperroutofwindowpkts":<Double_value>,

"tcperrcookiepktsigreject":<Double_value>,

"tcperrcookiepktseqdrop":<Double_value>,

"tcperrbadstateconnrate":<Double_value>,
```

```
"tcperrsyndroppedcongestionrate":<Double_value>,

"tcperrsvrholerate":<Double_value>,

"tcperrcltoutoforderrate":<Double_value>,

"tcperrdataafterfinrate":<Double_value>,

"tcperrsyninest":<Double_value>,

"tcptotzombiepassivehalfclosesrvconnflushed":<Double_value>,

"tcperrsecondretransmissions":<Double_value>,

"tcperrbadstateconn":<Double_value>,

"tcpspareconn":<Double_value>,

"tcperrretransmitgiveuprate":<Double_value>,

"tcptotcltfin":<Double_value>,

"tcptottxpkts":<Double_value>,

"tcpsynrate":<Double_value>,

"tcpclientconnopenedrate":<Double_value>,

"tcperranyportfail":<Double_value>,

"tcperrrstnonest":<Double_value>,

"tcpwaittosynrate":<Double_value>,

"tcperripportfail":<Double_value>,

"tcperrpartialretrasmit":<Double_value>,

"tcperrclthole":<Double_value>,

"tcpzombieactivehalfclosesvrconnflushedrate":<Double_value>,

"tcpcurclientconnestablished":<Double_value>,
```

```
"tcpwaittodata":<Double_value>,

"tcperrsyngiveuprate":<Double_value>,

"tcperrsentrst":<Double_value>,

"tcperrcookiepktseqdroprate":<Double_value>,

"tcperrbadchecksumrate":<Double_value>,

"tcperrseventhretransmissions":<Double_value>,

"tcperrrstoutofwindow":<Double_value>,

"tcpcurserverconnopening":<Double_value>,

"tcperrfullretrasmit":<Double_value>,

"tcptotzombieactivehalfclosesvrconnflushed":<Double_value>,

"tcperrcipalloc":<Double_value>,

"tcptotzombiecltconnflushed":<Double_value>,

"tcptotzombiehalfopensvrconnflushed":<Double_value>,

"pcbtotzombiecall":<Double_value>,

"tcpactiveserverconn":<Double_value>,

"tcperrsynretry":<Double_value>,

"tcptottxbytes":<Double_value>,

"tcperrfifthretransmissions":<Double_value>,

"tcperrrstthresholdrate":<Double_value>,

"tcpcurclientconnclosing":<Double_value>,

"tcpzombiehalfopencltconnflushedrate":<Double_value>,

"tcpzombiecltconnflushedrate":<Double_value>,
```

```
"tcptxpktsrate":<Double_value>

}]}
```

protocoludp

Statistics for UDP Protocol resource.

Properties

(click to see Operations)

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
clearstats	<String>	Read-write	Clear the statsistics / counters. Possible values = basic, full
udptotrxpkts	<Double>	Read-only	Total number of UDP packets received.
udprxpktsrate	<Double>	Read-only	Rate (/s) counter for udptotrxpkts
udptotrxbytes	<Double>	Read-only	Total number of UDP data received in bytes.
udprxbytesrate	<Double>	Read-only	Rate (/s) counter for udptotrxbytes
udptottxpkts	<Double>	Read-only	Total number of UDP packets transmitted.
udptxpktsrate	<Double>	Read-only	Rate (/s) counter for udptottxpkts
udptottxbytes	<Double>	Read-only	Total number of UDP data transmitted in bytes.

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
udptxbytesrate	<Double>	Read-only	Rate (/s) counter for udptottxbytes
udpcurratethreshold	<Double>	Read-only	Limit for UDP packets handled every 10 milliseconds. Default value, 0, applies no limit. This is a configurable value using the set rateControl command.
udptotunknownsvcpkts	<Double>	Read-only	Stray UDP packets dropped due to no configured listening service.
udpbadchecksum	<Double>	Read-only	Packets received with a UDP checksum error.
udpcurratethresholdexceeds	<Double>	Read-only	Number of times the UDP rate threshold is exceeded. If this counter continuously increases, first make sure the UDP packets received are genuine. If they are, increase the current rate threshold. This is a configurable value using the set rateControl command.

Operations

(click to see Properties)

GET (ALL)

Some options that you can use for each operations:

- **Getting warnings in response:**NITRO allows you to get warnings in an operation by specifying the "warning" query parameter as "yes". For example, to get warnings while connecting to the Citrix ADC appliance, the URL is as follows:

  http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/login?warning=yes

  If any, the warnings are displayed in the response payload with the HTTP code "209 X-NITRO-WARNING".

- **Authenticated access for individual NITRO operations:**NITRO allows you to logon to the Citrix ADC appliance to perform individual operations. You can use this option instead of creating a NITRO session (using the login object) and then using that session to perform all operations,

  To do this, you must specify the username and password in the request header of the NITRO request as follows:

  X-NITRO-USER:<username>

  X-NITRO-PASS:<password>

  Note: In such cases, make sure that the request header DOES not include the following:

  Cookie:NITRO_AUTH_TOKEN=<tokenvalue>

`*Note:`*

Mandatory parameters are marked in redand placeholder content is marked in <green>.

get (all)

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/stat/protocoludp

Query-parameters:

args

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/stat/protocoludp?
args=detail:<Boolean_value>,fullvalues:<Boolean_value>,ntimes:<Double_value>,logfile:<String_value

Use this query-parameter to get protocoludp resources based on additional properties.

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "protocoludp": [ {

"udptxpktsrate":<Double_value>,

"udpcurratethreshold":<Double_value>,

"udptotrxpkts":<Double_value>,

"udptottxpkts":<Double_value>,

"udptotrxbytes":<Double_value>,

"udptxbytesrate":<Double_value>,

"udprxpktsrate":<Double_value>,

"udpbadchecksum":<Double_value>,

"udptottxbytes":<Double_value>,

"udptotunknownsvcpkts":<Double_value>,

"udprxbytesrate":<Double_value>,

"udpcurratethresholdexceeds":<Double_value>
```

```
}]}
```

qos

Statistics for qos.

Properties

(click to see Operations)

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
name	<String>	Read-write	.
clearstats	<String>	Read-write	Clear the statsistics / counters. Possible values = basic, full
snmpqosqos_packets_received	<Double>	Read-only	Receive direction packets processed by QoS
snmpqosqos_packets_receivedrate	<Double>	Read-only	Rate (/s) counter for snmpqosqos_packets_received
snmpqosqos_packets_sent	<Double>	Read-only	Send direction packets processed by QoS
snmpqosqos_packets_sentrate	<Double>	Read-only	Rate (/s) counter for snmpqosqos_packets_sent
snmpqosqos_packets_bypassed	<Double>	Read-only	Packets bypassing QoS

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
snmpqosqos_packets_bypassedrate	<Double>	Read-only	Rate (/s) counter for snmpqosqos_packets_bypassed
snmpqosqos_packets_dropped	<Double>	Read-only	Total packets dropped
snmpqosqos_packets_droppedrate	<Double>	Read-only	Rate (/s) counter for snmpqosqos_packets_dropped
snmpqosqos_bytes_rx	<Double>	Read-only	Received bytes processed by QoS
snmpqosqos_bytes_rxrate	<Double>	Read-only	Rate (/s) counter for snmpqosqos_bytes_rx
snmpqosqos_bytes_tx	<Double>	Read-only	Sent bytes processed by QoS
snmpqosqos_bytes_txrate	<Double>	Read-only	Rate (/s) counter for snmpqosqos_bytes_tx
snmpqosqos_lazy_bytes	<Double>	Read-only	QoS lazy byte optimization rate
snmpqosqos_lazy_bytesrate	<Double>	Read-only	Rate (/s) counter for snmpqosqos_lazy_bytes
snmpqosqos_real_bytes	<Double>	Read-only	QoS actual bytes scheduled
snmpqosqos_real_bytesrate	<Double>	Read-only	Rate (/s) counter for snmpqosqos_real_bytes
snmpqosqos_packets_filtered	<Double>	Read-only	Total packets filtered by QoS

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
snmpqosqos_packets_filteredrate	<Double>	Read-only	Rate (/s) counter for snmpqosqos_packets_filtered
snmpqosqos_packets_classified	<Double>	Read-only	Total packets classified by QoS
snmpqosqos_packets_classifiedrate	<Double>	Read-only	Rate (/s) counter for snmpqosqos_packets_classified
snmpqosqos_flows	<Double>	Read-only	New QoS flows
snmpqosqos_flowsrate	<Double>	Read-only	Rate (/s) counter for snmpqosqos_flows
snmpqosqos_flow_recycles	<Double>	Read-only	Recycled QoS flows
snmpqosqos_flow_recyclesrate	<Double>	Read-only	Rate (/s) counter for snmpqosqos_flow_recycles
snmpqosqos_session_recycle_failure	<Double>	Read-only	QoS Flow Recycle failures
snmpqosqos_session_recycle_failurerate	<Double>	Read-only	Rate (/s) counter for snmpqosqos_session_recycle_failure
snmpqosqos_sessions_ignored	<Double>	Read-only	Sessions manually ignored
snmpqosqos_sessions_ignoredrate	<Double>	Read-only	Rate (/s) counter for snmpqosqos_sessions_ignored
snmpqosqos_sessions_consumed	<Double>	Read-only	sessions manually consumed

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
snmpqosqos_sessions_consumedrate	<Double>	Read-only	Rate (/s) counter for snmpqosqos_sessions_consumed
snmpqosqos_actions_created	<Double>	Read-only	Uneque qos action objects created
snmpqosqos_actions_createdrate	<Double>	Read-only	Rate (/s) counter for snmpqosqos_actions_created
snmpqosqos_policy_reeval	<Double>	Read-only	Policies re-evaluated due to cli change
snmpqosqos_policy_reevalrate	<Double>	Read-only	Rate (/s) counter for snmpqosqos_policy_reeval
snmpqosqos_cfy_tcp_unknown	<Double>	Read-only	Connections unable to be classified beyond TCP
snmpqosqos_cfy_tcp_unknownrate	<Double>	Read-only	Rate (/s) counter for snmpqosqos_cfy_tcp_unknown
snmpqosqos_cfy_udp_unknown	<Double>	Read-only	Connections unable to be classified beyond UDP
snmpqosqos_cfy_udp_unknownrate	<Double>	Read-only	Rate (/s) counter for snmpqosqos_cfy_udp_unknown
snmpqosqos_sch_leafs	<Double>	Read-only	Scheduler leaf nodes constructed
snmpqosqos_sch_leafsrate	<Double>	Read-only	Rate (/s) counter for snmpqosqos_sch_leafs

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
snmpqosqos_session_mem	<Double>	Read-only	Session memory allocated
snmpqosqos_session_memrate	<Double>	Read-only	Rate (/s) counter for snmpqosqos_session_mem
snmpqosqos_sch_virtual_packets	<Double>	Read-only	Scheduler virtual packets constructed
snmpqosqos_sch_virtual_packetsrate	<Double>	Read-only	Rate (/s) counter for snmpqosqos_sch_virtual_packets
snmpqosqos_sch_virtual_bytes_accepted	<Double>	Read-only	Scheduler bytes accepted
snmpqosqos_sch_virtual_bytes_acceptedrate	<Double>	Read-only	Rate (/s) counter for snmpqosqos_sch_virtual_bytes_accepted
snmpqosqos_sch_leaf_recycle_failures	<Double>	Read-only	Scheduler Failures to recycle QoS flows
snmpqosqos_sch_leaf_recycle_failuresrate	<Double>	Read-only	Rate (/s) counter for snmpqosqos_sch_leaf_recycle_failures
snmpqosqos_sch_node_regulated_count	<Double>	Read-only	Scheduler Regulated node count
snmpqosqos_sch_node_regulated_countrate	<Double>	Read-only	Rate (/s) counter for snmpqosqos_sch_node_regulated_count
snmpqosqos_sch_sessions_created	<Double>	Read-only	Scheduler session classes constructed
snmpqosqos_sch_sessions_createdrate	<Double>	Read-only	Rate (/s) counter for snmpqosqos_sch_sessions_created

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
snmpqosqos_sch_sessions_deleted	<Double>	Read-only	Scheduler session classes constructed
snmpqosqos_sch_sessions_deletedrate	<Double>	Read-only	Rate (/s) counter for snmpqosqos_sch_sessions_deleted
snmpqosqos_sch_sdrr_nodes	<Double>	Read-only	Scheduler sdrr nodes constructed
snmpqosqos_sch_sdrr_nodesrate	<Double>	Read-only	Rate (/s) counter for snmpqosqos_sch_sdrr_nodes
snmpqosqos_sch_session_conns	<Double>	Read-only	Scheduler session connections created
snmpqosqos_sch_session_connsrate	<Double>	Read-only	Rate (/s) counter for snmpqosqos_sch_session_conns
snmpqosqos_sch_session_conns_removed	<Double>	Read-only	Scheduler session connections removed
snmpqosqos_sch_session_conns_removedrate	<Double>	Read-only	Rate (/s) counter for snmpqosqos_sch_session_conns_removed
snmpqosqos_sch_sessions_regulated_count	<Double>	Read-only	Scheduler regulated sessions count
snmpqosqos_sch_sessions_regulated_countrate	<Double>	Read-only	Rate (/s) counter for snmpqosqos_sch_sessions_regulated_count
snmpqosqos_sch_sessions_byte_count	<Double>	Read-only	Scheduler session bytes total

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
snmpqosqos_sch_sessions_byte_countrate	<Double>	Read-only	Rate (/s) counter for snmpqosqos_sch_sessions_byte_count
snmpqosqos_sch_regulated_count	<Double>	Read-only	Scheduler regulated node count
snmpqosqos_sch_regulated_countrate	<Double>	Read-only	Rate (/s) counter for snmpqosqos_sch_regulated_count
snmpqosqos_sch_links_created	<Double>	Read-only	Scheduler links created
snmpqosqos_sch_links_createdrate	<Double>	Read-only	Rate (/s) counter for snmpqosqos_sch_links_created
snmpqosqos_sch_links_deleted	<Double>	Read-only	Scheduler links deleted
snmpqosqos_sch_links_deletedrate	<Double>	Read-only	Rate (/s) counter for snmpqosqos_sch_links_deleted
snmpqosqos_sch_links_updated	<Double>	Read-only	Scheduler links updated
snmpqosqos_sch_links_updatedrate	<Double>	Read-only	Rate (/s) counter for snmpqosqos_sch_links_updated
snmpqosqos_sch_poll_count	<Double>	Read-only	Scheduler calls to poll_libqos
snmpqosqos_sch_poll_countrate	<Double>	Read-only	Rate (/s) counter for snmpqosqos_sch_poll_count
snmpqosqos_sch_peer_msgs	<Double>	Read-only	Scheduler peer messages received

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
snmpqosqos_sch_peer_msgsrate	<Double>	Read-only	Rate (/s) counter for snmpqosqos_sch_peer_msgs
snmpqosqos_error_ipc	<Double>	Read-only	IPC failed for QoS messages.
snmpqosqos_error_ipcrate	<Double>	Read-only	Rate (/s) counter for snmpqosqos_error_ipc
snmpqosqos_flow_mem	<Double>	Read-only	Flow memory allocated
snmpqosqos_flow_memrate	<Double>	Read-only	Rate (/s) counter for snmpqosqos_flow_mem
snmpqosqos_flows_available	<Double>	Read-only	Flows free list size
snmpqosqos_flows_availablerate	<Double>	Read-only	Rate (/s) counter for snmpqosqos_flows_available
snmpqosqos_recycle_failed_backlog	<Double>	Read-only	Recycle failed due to backlog
snmpqosqos_recycle_failed_backlograte	<Double>	Read-only	Rate (/s) counter for snmpqosqos_recycle_failed_backlog
snmpqosqos_recycle_failed_session	<Double>	Read-only	Recycle failed due to session attachment
snmpqosqos_recycle_failed_sessionrate	<Double>	Read-only	Rate (/s) counter for snmpqosqos_recycle_failed_session
snmpqosqos_error_create_action_failed	<Double>	Read-only	Failed attempts to create actions

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
snmpqosqos_error_create_action_failedrate	<Double>	Read-only	Rate (/s) counter for snmpqosqos_error_create_action_failed
snmpqosqos_error_modify_action_failed	<Double>	Read-only	Failed attempts to modify actions
snmpqosqos_error_modify_action_failedrate	<Double>	Read-only	Rate (/s) counter for snmpqosqos_error_modify_action_failed
snmpqosqos_error_remove_action_failed	<Double>	Read-only	Failed attempts to remove actions
snmpqosqos_error_remove_action_failed rate	<Double>	Read-only	Rate (/s) counter for snmpqosqos_error_remove_action_failed
snmpqosqos_error_cli_unknown	<Double>	Read-only	Internal CLI error
snmpqosqos_error_cli_unknownrate	<Double>	Read-only	Rate (/s) counter for snmpqosqos_error_cli_unknown
snmpqosqos_error_rename_not_implemented	<Double>	Read-only	qos action rename not yet implemented
snmpqosqos_error_rename_not_implementedrate	<Double>	Read-only	Rate (/s) counter for snmpqosqos_error_rename_not_implemented
snmpqosqos_error_remove_policy_failed	<Double>	Read-only	Failed attempts to remove qos policy
snmpqosqos_error_remove_policy_failedrate	<Double>	Read-only	Rate (/s) counter for snmpqosqos_error_remove_policy_failed

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
snmpqosqos_error_create_policy_failed	<Double>	Read-only	Failed attempts to create qos policy
snmpqosqos_error_create_policy_failedrate	<Double>	Read-only	Rate (/s) counter for snmpqosqos_error_create_policy_failed
snmpqosqos_error_libqos_api_failures	<Double>	Read-only	Libqos api failures
snmpqosqos_error_libqos_api_failuresrate	<Double>	Read-only	Rate (/s) counter for snmpqosqos_error_libqos_api_failures
snmpqosqos_error_api_ses_invalidpcb	<Double>	Read-only	Libqos api qos_session_add_pcb/natpcb() failed for reason QS_EINVALIDPCB
snmpqosqos_error_api_ses_invalidpcbrate	<Double>	Read-only	Rate (/s) counter for snmpqosqos_error_api_ses_invalidpcb
snmpqosqos_error_api_ses_notready	<Double>	Read-only	Libqos api qos_session_add_pcb/natpcb() failed for reason QS_ENOTREADY
snmpqosqos_error_api_ses_notreadyrate	<Double>	Read-only	Rate (/s) counter for snmpqosqos_error_api_ses_notready
snmpqosqos_error_api_ses_add_insession	<Double>	Read-only	Libqos api qos_session_add_pcb/natpcb() failed for reason QS_EINSESSION

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
snmpqosqos_error_api_ses_add_insessionrate	<Double>	Read-only	Rate (/s) counter for snmpqosqos_error_api_ses_add_insession
snmpqosqos_error_api_ses_add_other	<Double>	Read-only	Libqos api qos_session_add_pcb/natpcb() failed
snmpqosqos_error_api_ses_add_otherrate	<Double>	Read-only	Rate (/s) counter for snmpqosqos_error_api_ses_add_other
snmpqosqos_error_api_ses_rem_notinsession	<Double>	Read-only	Libqos api qos_session_rem_pcb/natpcb() failed for reason QS_ENOTINSESSION
snmpqosqos_error_api_ses_rem_notinsessionrate	<Double>	Read-only	Rate (/s) counter for snmpqosqos_error_api_ses_rem_notinsession
snmpqosqos_error_api_ses_rem_other	<Double>	Read-only	Libqos api qos_session_rem_pcb/natpcb() failed
snmpqosqos_error_api_ses_rem_otherrate	<Double>	Read-only	Rate (/s) counter for snmpqosqos_error_api_ses_rem_other
snmpqosqos_error_api_ses_del	<Double>	Read-only	Libqos api qos_session_delete faled
snmpqosqos_error_api_ses_delrate	<Double>	Read-only	Rate (/s) counter for snmpqosqos_error_api_ses_del

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
snmpqosqos_error_no_flows	<Double>	Read-only	Libqos out of flow memory
snmpqosqos_error_no_flowsrate	<Double>	Read-only	Rate (/s) counter for snmpqosqos_error_no_flows
snmpqosqos_error_no_sessions	<Double>	Read-only	Libqos out of session memory
snmpqosqos_error_no_sessionsrate	<Double>	Read-only	Rate (/s) counter for snmpqosqos_error_no_sessions

Operations

(click to see Properties)

GET (ALL)

Some options that you can use for each operations:

- **Getting warnings in response:**NITRO allows you to get warnings in an operation by specifying the "warning" query parameter as "yes". For example, to get warnings while connecting to the Citrix ADC appliance, the URL is as follows:

  http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/login?warning=yes

  If any, the warnings are displayed in the response payload with the HTTP code "209 X-NITRO-WARNING".

- **Authenticated access for individual NITRO operations:**NITRO allows you to logon to the Citrix ADC appliance to perform individual operations. You can use this option instead of creating a NITRO session (using the login object) and then using that session to perform all operations,

  To do this, you must specify the username and password in the request header of the NITRO request as follows:

  X-NITRO-USER:<username>

  X-NITRO-PASS:<password>

Note: In such cases, make sure that the request header DOES not include the following:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue>

`*Note:` *

Mandatory parameters are marked in redand placeholder content is marked in <green>.

get (all)

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/stat/qos

Query-parameters:

args

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/stat/qos?
args=name:<String_value>,detail:<Boolean_value>,fullvalues:<Boolean_value>,ntimes:<Double_value>

Use this query-parameter to get qos resources based on additional properties.

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "qos": [ {

"snmpqosqos_cfy_udp_unknownrate":<Double_value>,

"snmpqosqos_flow_memrate":<Double_value>,
```

```
"snmpqosqos_sch_sessions_createdrate":<Double_value>,

"snmpqosqos_sch_peer_msgs":<Double_value>,

"snmpqosqos_packets_classified":<Double_value>,

"snmpqosqos_sch_session_conns_removed":<Double_value>,

"snmpqosqos_sch_regulated_countrate":<Double_value>,

"snmpqosqos_actions_createdrate":<Double_value>,

"snmpqosqos_error_no_sessions":<Double_value>,

"snmpqosqos_flows_available":<Double_value>,

"snmpqosqos_sch_virtual_bytes_acceptedrate":<Double_value>,

"snmpqosqos_flowsrate":<Double_value>,

"snmpqosqos_packets_droppedrate":<Double_value>,

"snmpqosqos_lazy_bytesrate":<Double_value>,

"snmpqosqos_error_remove_policy_failed":<Double_value>,

"snmpqosqos_sch_links_deleted":<Double_value>,

"snmpqosqos_recycle_failed_backlograte":<Double_value>,

"snmpqosqos_real_bytesrate":<Double_value>,

"snmpqosqos_session_recycle_failurerate":<Double_value>,

"snmpqosqos_cfy_tcp_unknown":<Double_value>,

"snmpqosqos_error_rename_not_implementedrate":<Double_value>,

"snmpqosqos_sessions_ignoredrate":<Double_value>,

"snmpqosqos_bytes_rx":<Double_value>,

"snmpqosqos_error_remove_action_failed":<Double_value>,
```

```
"snmpqosqos_packets_filtered":<Double_value>,

"snmpqosqos_error_api_ses_add_insessionrate":<Double_value>,

"snmpqosqos_error_api_ses_notready":<Double_value>,

"snmpqosqos_recycle_failed_backlog":<Double_value>,

"snmpqosqos_sch_session_connsrate":<Double_value>,

"snmpqosqos_error_create_action_failed":<Double_value>,

"snmpqosqos_sch_virtual_packets":<Double_value>,

"snmpqosqos_packets_sent":<Double_value>,

"snmpqosqos_session_memrate":<Double_value>,

"snmpqosqos_error_api_ses_invalidpcb":<Double_value>,

"snmpqosqos_error_rename_not_implemented":<Double_value>,

"snmpqosqos_sch_links_createdrate":<Double_value>,

"snmpqosqos_bytes_rxrate":<Double_value>,

"snmpqosqos_error_api_ses_invalidpcbrate":<Double_value>,

"snmpqosqos_sch_leaf_recycle_failures":<Double_value>,

"snmpqosqos_error_modify_action_failed":<Double_value>,

"snmpqosqos_error_create_policy_failed":<Double_value>,

"snmpqosqos_packets_receivedrate":<Double_value>,

"snmpqosqos_sch_links_deletedrate":<Double_value>,

"snmpqosqos_policy_reeval":<Double_value>,

"snmpqosqos_real_bytes":<Double_value>,

"snmpqosqos_error_api_ses_add_insession":<Double_value>,
```

```
"snmpqosqos_error_api_ses_add_other":<Double_value>,

"snmpqosqos_sch_sessions_deleted":<Double_value>,

"snmpqosqos_bytes_tx":<Double_value>,

"snmpqosqos_packets_classifiedrate":<Double_value>,

"snmpqosqos_error_libqos_api_failures":<Double_value>,

"snmpqosqos_error_create_action_failedrate":<Double_value>,

"snmpqosqos_error_api_ses_del":<Double_value>,

"snmpqosqos_packets_filteredrate":<Double_value>,

"snmpqosqos_flows_availablerate":<Double_value>,

"snmpqosqos_session_recycle_failure":<Double_value>,

"snmpqosqos_sch_virtual_packetsrate":<Double_value>,

"snmpqosqos_bytes_txrate":<Double_value>,

"snmpqosqos_packets_bypassed":<Double_value>,

"snmpqosqos_cfy_udp_unknown":<Double_value>,

"snmpqosqos_packets_received":<Double_value>,

"snmpqosqos_error_create_policy_failedrate":<Double_value>,

"snmpqosqos_error_api_ses_delrate":<Double_value>,

"snmpqosqos_flow_mem":<Double_value>,

"snmpqosqos_session_mem":<Double_value>,

"snmpqosqos_recycle_failed_sessionrate":<Double_value>,

"snmpqosqos_sch_poll_count":<Double_value>,

"snmpqosqos_lazy_bytes":<Double_value>,
```

```
"snmpqosqos_error_remove_action_failedrate":<Double_value>,

"snmpqosqos_sch_sessions_regulated_count":<Double_value>,

"snmpqosqos_error_api_ses_add_otherrate":<Double_value>,

"snmpqosqos_error_api_ses_rem_otherrate":<Double_value>,

"snmpqosqos_recycle_failed_session":<Double_value>,

"snmpqosqos_sch_sessions_byte_countrate":<Double_value>,

"snmpqosqos_sch_links_updated":<Double_value>,

"snmpqosqos_sch_sdrr_nodes":<Double_value>,

"snmpqosqos_sch_peer_msgsrate":<Double_value>,

"snmpqosqos_sch_session_conns":<Double_value>,

"snmpqosqos_error_libqos_api_failuresrate":<Double_value>,

"snmpqosqos_error_cli_unknownrate":<Double_value>,

"snmpqosqos_actions_created":<Double_value>,

"snmpqosqos_sch_virtual_bytes_accepted":<Double_value>,

"snmpqosqos_sessions_consumed":<Double_value>,

"snmpqosqos_error_remove_policy_failedrate":<Double_value>,

"snmpqosqos_error_cli_unknown":<Double_value>,

"snmpqosqos_flow_recycles":<Double_value>,

"snmpqosqos_sch_sessions_deletedrate":<Double_value>,

"snmpqosqos_sch_poll_countrate":<Double_value>,

"snmpqosqos_sch_node_regulated_countrate":<Double_value>,

"snmpqosqos_error_no_sessionsrate":<Double_value>,
```

```
"snmpqosqos_error_api_ses_rem_notinsession":<Double_value>,

"snmpqosqos_flows":<Double_value>,

"snmpqosqos_sch_sessions_regulated_countrate":<Double_value>,

"snmpqosqos_sch_session_conns_removedrate":<Double_value>,

"snmpqosqos_sch_sessions_byte_count":<Double_value>,

"snmpqosqos_flow_recyclesrate":<Double_value>,

"snmpqosqos_sch_leafs":<Double_value>,

"snmpqosqos_error_api_ses_rem_other":<Double_value>,

"snmpqosqos_policy_reevalrate":<Double_value>,

"snmpqosqos_sch_links_created":<Double_value>,

"snmpqosqos_error_api_ses_notreadyrate":<Double_value>,

"snmpqosqos_error_no_flowsrate":<Double_value>,

"snmpqosqos_error_ipcrate":<Double_value>,

"snmpqosqos_error_modify_action_failedrate":<Double_value>,

"snmpqosqos_sch_regulated_count":<Double_value>,

"snmpqosqos_sch_sessions_created":<Double_value>,

"snmpqosqos_error_no_flows":<Double_value>,

"snmpqosqos_error_api_ses_rem_notinsessionrate":<Double_value>,

"snmpqosqos_sch_leafsrate":<Double_value>,

"snmpqosqos_sch_node_regulated_count":<Double_value>,

"snmpqosqos_error_ipc":<Double_value>,

"snmpqosqos_sch_leaf_recycle_failuresrate":<Double_value>,
```

```
"snmpqosqos_packets_dropped":<Double_value>,

"snmpqosqos_sessions_consumedrate":<Double_value>,

"snmpqosqos_sessions_ignored":<Double_value>,

"snmpqosqos_cfy_tcp_unknownrate":<Double_value>,

"snmpqosqos_sch_sdrr_nodesrate":<Double_value>,

"snmpqosqos_sch_links_updatedrate":<Double_value>,

"snmpqosqos_packets_sentrate":<Double_value>,

"snmpqosqos_packets_bypassedrate":<Double_value>

}]}
```

Responder

Responder statistics.

NAME	DESCRIPTION
responderpolicy	Statistics for responder policy resource.
responderpolicylabel	Statistics for responder policy label resource.

responderpolicy

Statistics for responder policy resource.

Properties

(click to see Operations)

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
name	<String>	Read-write	Name of the responder policy for

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
			which to show detailed statistics. Minimum length = 1
clearstats	<String>	Read-write	Clear the statsistics / counters. Possible values = basic, full
pipolicyhits	<Double>	Read-only	Number of hits on the policy
pipolicyhitsrate	<Double>	Read-only	Rate (/s) counter for pipolicyhits
pipolicyundefhits	<Double>	Read-only	Number of undef hits on the policy
pipolicyundefhitsrate	<Double>	Read-only	Rate (/s) counter for pipolicyundefhits

Operations

(click to see Properties)

GET (ALL)| GET

Some options that you can use for each operations:

- **Getting warnings in response:**NITRO allows you to get warnings in an operation by specifying the "warning" query parameter as "yes". For example, to get warnings while connecting to the Citrix ADC appliance, the URL is as follows:

  http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/login?warning=yes

  If any, the warnings are displayed in the response payload with the HTTP code "209 X-NITRO-WARNING".

- **Authenticated access for individual NITRO operations:**NITRO allows you to logon to the Citrix ADC appliance to perform individual operations. You can use this option instead of

creating a NITRO session (using the login object) and then using that session to perform all operations,

To do this, you must specify the username and password in the request header of the NITRO request as follows:

X-NITRO-USER:<username>

X-NITRO-PASS:<password>

Note: In such cases, make sure that the request header DOES not include the following:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue>

`*Note:`*

Mandatory parameters are marked in redand placeholder content is marked in <green>.

get (all)

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/stat/responderpolicy

Query-parameters:

args

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/stat/responderpolicy?**args=name:<String_value>,detail:<Boolean_value>,fullvalues:<Boolean_value>,ntimes:<Double_value>**

Use this query-parameter to get responderpolicy resources based on additional properties.

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "responderpolicy": [ {

"name":<String_value>,

"pipolicyundefhitsrate":<Double_value>,

"pipolicyundefhits":<Double_value>,

"pipolicyhitsrate":<Double_value>,

"pipolicyhits":<Double_value>

}]}
```

get

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/stat/responderpolicy/name_value><String>

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "responderpolicy": [ {

"name":<String_value>,

"pipolicyundefhitsrate":<Double_value>,
```

```
"pipolicyundefhits":<Double_value>,

"pipolicyhitsrate":<Double_value>,

"pipolicyhits":<Double_value>

}]}
```

responderpolicylabel

Statistics for responder policy label resource.

Properties

(click to see Operations)

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
labelname	<String>	Read-write	Name of the responder policy label. Minimum length = 1
clearstats	<String>	Read-write	Clear the statsistics / counters. Possible values = basic, full
pipolicylabelhits	<Double>	Read-only	Number of times policy label was invoked.
pipolicylabelhitsrate	<Double>	Read-only	Rate (/s) counter for pipolicylabelhits

Operations

(click to see Properties)

GET (ALL)| GET

Some options that you can use for each operations:

- **Getting warnings in response:**NITRO allows you to get warnings in an operation by specifying the "warning" query parameter as "yes". For example, to get warnings while connecting to the Citrix ADC appliance, the URL is as follows:

  http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/login?warning=yes

  If any, the warnings are displayed in the response payload with the HTTP code "209 X-NITRO-WARNING".

- **Authenticated access for individual NITRO operations:**NITRO allows you to logon to the Citrix ADC appliance to perform individual operations. You can use this option instead of creating a NITRO session (using the login object) and then using that session to perform all operations,

  To do this, you must specify the username and password in the request header of the NITRO request as follows:

  X-NITRO-USER:<username>

  X-NITRO-PASS:<password>

  Note: In such cases, make sure that the request header DOES not include the following:

  Cookie:NITRO_AUTH_TOKEN=<tokenvalue>

`*Note:`*

Mandatory parameters are marked in redand placeholder content is marked in <green>.

get (all)

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/stat/responderpolicylabel

Query-parameters:

args

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/stat/responderpolicylabel?
args=labelname:<String_value>,detail:<Boolean_value>,fullvalues:<Boolean_value>,ntimes:<Double_va

Use this query-parameter to get responderpolicylabel resources based on additional properties.

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "responderpolicylabel": [ {

"labelname":<String_value>,

"pipolicylabelhitsrate":<Double_value>,

"pipolicylabelhits":<Double_value>

}]}
```

get

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/stat/responderpolicylabel/labelname_value><String>

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "responderpolicylabel": [ {

"labelname":<String_value>,

"pipolicylabelhitsrate":<Double_value>,

"pipolicylabelhits":<Double_value>

}]}
```

Rewrite

Rewrite statistics.

NAME	DESCRIPTION
rewritepolicy	Statistics for rewrite policy resource.
rewritepolicylabel	Statistics for rewrite policy label resource.

rewritepolicy

Statistics for rewrite policy resource.

Properties

(click to see Operations)

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
name	<String>	Read-write	Name of the rewrite policy. Minimum length = 1
clearstats	<String>	Read-write	Clear the statsistics / counters.

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
			Possible values = basic, full
pipolicyhits	<Double>	Read-only	Number of hits on the policy
pipolicyhitsrate	<Double>	Read-only	Rate (/s) counter for pipolicyhits
pipolicyundefhits	<Double>	Read-only	Number of undef hits on the policy
pipolicyundefhitsrate	<Double>	Read-only	Rate (/s) counter for pipolicyundefhits

Operations

(click to see Properties)

GET (ALL)| GET

Some options that you can use for each operations:

- **Getting warnings in response:**NITRO allows you to get warnings in an operation by specifying the "warning" query parameter as "yes". For example, to get warnings while connecting to the Citrix ADC appliance, the URL is as follows:

  http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/login?warning=yes

  If any, the warnings are displayed in the response payload with the HTTP code "209 X-NITRO-WARNING".

- **Authenticated access for individual NITRO operations:**NITRO allows you to logon to the Citrix ADC appliance to perform individual operations. You can use this option instead of creating a NITRO session (using the login object) and then using that session to perform all operations,

  To do this, you must specify the username and password in the request header of the NITRO request as follows:

  X-NITRO-USER:<username>

X-NITRO-PASS:<password>

Note: In such cases, make sure that the request header DOES not include the following:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue>

`*Note:`*

Mandatory parameters are marked in redand placeholder content is marked in <green>.

get (all)

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/stat/rewritepolicy

Query-parameters:

args

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/stat/rewritepolicy?
args=name:<String_value>,detail:<Boolean_value>,fullvalues:<Boolean_value>,ntimes:<Double_value>

Use this query-parameter to get rewritepolicy resources based on additional properties.

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "rewritepolicy": [ {

"name":<String_value>,

"pipolicyundefhitsrate":<Double_value>,
```

```
"pipolicyundefhits":<Double_value>,

"pipolicyhitsrate":<Double_value>,

"pipolicyhits":<Double_value>

}]}
```

get

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/stat/rewritepolicy/name_value><String>

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "rewritepolicy": [ {

"name":<String_value>,

"pipolicyundefhitsrate":<Double_value>,

"pipolicyundefhits":<Double_value>,

"pipolicyhitsrate":<Double_value>,

"pipolicyhits":<Double_value>
```

```
}]}
```

rewritepolicylabel

Statistics for rewrite policy label resource.

Properties

(click to see Operations)

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
labelname	<String>	Read-write	Name of the rewrite policy label. Minimum length = 1
clearstats	<String>	Read-write	Clear the statsistics / counters. Possible values = basic, full
pipolicylabelhits	<Double>	Read-only	Number of times policy label was invoked.
pipolicylabelhitsrate	<Double>	Read-only	Rate (/s) counter for pipolicylabelhits

Operations

(click to see Properties)

GET (ALL)| GET

Some options that you can use for each operations:

- **Getting warnings in response:**NITRO allows you to get warnings in an operation by specifying the "warning" query parameter as "yes". For example, to get warnings while connecting to the Citrix ADC appliance, the URL is as follows:

  http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/login?warning=yes

If any, the warnings are displayed in the response payload with the HTTP code "209 X-NITRO-WARNING".

- **Authenticated access for individual NITRO operations:**NITRO allows you to logon to the Citrix ADC appliance to perform individual operations. You can use this option instead of creating a NITRO session (using the login object) and then using that session to perform all operations,

  To do this, you must specify the username and password in the request header of the NITRO request as follows:

  X-NITRO-USER:<username>

  X-NITRO-PASS:<password>

  Note: In such cases, make sure that the request header DOES not include the following:

  Cookie:NITRO_AUTH_TOKEN=<tokenvalue>

`*Note:`*

Mandatory parameters are marked in redand placeholder content is marked in <green>.

get (all)

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/stat/rewritepolicylabel

Query-parameters:

args

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/stat/rewritepolicylabel?
args=labelname:<String_value>,detail:<Boolean_value>,fullvalues:<Boolean_value>,ntimes:<Double_va

Use this query-parameter to get rewritepolicylabel resources based on additional properties.

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "rewritepolicylabel": [ {

"labelname":<String_value>,

"pipolicylabelhitsrate":<Double_value>,

"pipolicylabelhits":<Double_value>

}]}
```

get

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/stat/rewritepolicylabel/labelname_value><String>

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "rewritepolicylabel": [ {

"labelname":<String_value>,

"pipolicylabelhitsrate":<Double_value>,

"pipolicylabelhits":<Double_value>

}]}
```

snmp

Statistics for snmp.

Properties

(click to see Operations)

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
clearstats	<String>	Read-write	Clear the statsistics / counters. Possible values = basic, full
snmptotrxpkts	<Double>	Read-only	SNMP packets received.
snmprxpktsrate	<Double>	Read-only	Rate (/s) counter for snmptotrxpkts
snmptottxpkts	<Double>	Read-only	SNMP packets transmitted.
snmptxpktsrate	<Double>	Read-only	Rate (/s) counter for snmptottxpkts
snmptotgetreqs	<Double>	Read-only	SNMP Get-Request PDUs that have

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
			been accepted and processed.
snmpgetreqsrate	<Double>	Read-only	Rate (/s) counter for snmptotgetreqs
snmptotgetnextreqs	<Double>	Read-only	SNMP Get-Next PDUs that have been accepted and processed.
snmpgetnextreqsrate	<Double>	Read-only	Rate (/s) counter for snmptotgetnextreqs
snmptotgetbulkreqs	<Double>	Read-only	SNMP Get-Bulk PDUs that have been accepted and processed.
snmpgetbulkreqsrate	<Double>	Read-only	Rate (/s) counter for snmptotgetbulkreqs
snmptotresponses	<Double>	Read-only	SNMP Get-Response PDUs that have been generated by the Citrix ADC.
snmpresponsesrate	<Double>	Read-only	Rate (/s) counter for snmptotresponses
snmptottraps	<Double>	Read-only	SNMP Trap PDUs that have been generated by the Citrix ADC.
snmptoterrreqdroppe d	<Double>	Read-only	SNMP requests dropped.
snmptotparseerrs	<Double>	Read-only	Number of ASN.1 or BER errors encountered when

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
			decoding received SNMP Messages.
snmptotbadversions	<Double>	Read-only	Number of SNMP messages received, which were for an unsupported SNMP version.
snmptotbadcommname	<Double>	Read-only	SNMP messages received, which used an SNMP community name not known to the Citrix ADC.
snmptotbadcommuse	<Double>	Read-only	The total number of SNMP Messages received that represented an SNMP operation which was not allowed by the SNMP community named in the Message.
snmpunsupportedsecuritylevel	<Double>	Read-only	SNMP packets that were dropped because they requested a security level that was unknown to the Citrix ADC or otherwise unavailable.
snmpnotintimewindow	<Double>	Read-only	SNMP packets that were dropped because they appeared outside of the authoritative

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
			SNMP engine's window.
snmpunknownusername	<Double>	Read-only	SNMP packets that were dropped because they referenced a user that was not known to the SNMP engine.
snmpunknownengineids	<Double>	Read-only	SNMP packets that were dropped because they referenced an SNMP engine ID that was not known to the Citrix ADC.
snmpwrongdigests	<Double>	Read-only	SNMP packets that were dropped because they did not contain the expected digest value.
snmpdecryptionerrors	<Double>	Read-only	SNMP packets that were dropped because they could not be decrypted.

Operations

(click to see Properties)

GET (ALL)

Some options that you can use for each operations:

- **Getting warnings in response:**NITRO allows you to get warnings in an operation by specifying the "warning" query parameter as "yes". For example, to get warnings while connecting to the Citrix ADC appliance, the URL is as follows:

  http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/login?warning=yes

If any, the warnings are displayed in the response payload with the HTTP code "209 X-NITRO-WARNING".

- **Authenticated access for individual NITRO operations:**NITRO allows you to logon to the Citrix ADC appliance to perform individual operations. You can use this option instead of creating a NITRO session (using the login object) and then using that session to perform all operations,

  To do this, you must specify the username and password in the request header of the NITRO request as follows:

  X-NITRO-USER:<username>

  X-NITRO-PASS:<password>

  Note: In such cases, make sure that the request header DOES not include the following:

  Cookie:NITRO_AUTH_TOKEN=<tokenvalue>

`*Note:`*

Mandatory parameters are marked in redand placeholder content is marked in <green>.

get (all)

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/stat/snmp

Query-parameters:

args

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/stat/snmp?
args=detail:<Boolean_value>,fullvalues:<Boolean_value>,ntimes:<Double_value>,logfile:<String_value

Use this query-parameter to get snmp resources based on additional properties.

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "snmp": [ {

"snmpdecryptionerrors":<Double_value>,

"snmptotresponses":<Double_value>,

"snmptotbadcommuse":<Double_value>,

"snmptoterrreqdropped":<Double_value>,

"snmpgetnextreqsrate":<Double_value>,

"snmptotrxpkts":<Double_value>,

"snmptottxpkts":<Double_value>,

"snmptotparseerrs":<Double_value>,

"snmptottraps":<Double_value>,

"snmptotbadversions":<Double_value>,

"snmptxpktsrate":<Double_value>,

"snmpresponsesrate":<Double_value>,

"snmpgetreqsrate":<Double_value>,

"snmptotbadcommname":<Double_value>,

"snmptotgetnextreqs":<Double_value>,

"snmpunknownengineids":<Double_value>,
```

```
"snmpwrongdigests":<Double_value>,

"snmpgetbulkreqsrate":<Double_value>,

"snmpnotintimewindow":<Double_value>,

"snmptotgetbulkreqs":<Double_value>,

"snmpunknownusername":<Double_value>,

"snmpunsupportedsecuritylevel":<Double_value>,

"snmptotgetreqs":<Double_value>,

"snmprxpktsrate":<Double_value>

}]}
```

Spillover

Spillover policies and actions.

NAME	DESCRIPTION
spilloverpolicy	Statistics for Spillover policy resource.

spilloverpolicy

Statistics for Spillover policy resource.

Properties

(click to see Operations)

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
name	<String>	Read-write	Name of the spillover policy for which to show detailed statistics.

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
			Minimum length = 1
clearstats	<String>	Read-write	Clear the statsistics / counters. Possible values = basic, full
pipolicyhits	<Double>	Read-only	Number of hits on the policy
pipolicyhitsrate	<Double>	Read-only	Rate (/s) counter for pipolicyhits
pipolicyundefhits	<Double>	Read-only	Number of undef hits on the policy
pipolicyundefhitsrate	<Double>	Read-only	Rate (/s) counter for pipolicyundefhits

Operations

(click to see Properties)

GET (ALL)| GET

Some options that you can use for each operations:

- **Getting warnings in response:**NITRO allows you to get warnings in an operation by specifying the "warning" query parameter as "yes". For example, to get warnings while connecting to the Citrix ADC appliance, the URL is as follows:

  http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/login?warning=yes

  If any, the warnings are displayed in the response payload with the HTTP code "209 X-NITRO-WARNING".
- **Authenticated access for individual NITRO operations:**NITRO allows you to logon to the Citrix ADC appliance to perform individual operations. You can use this option instead of creating a NITRO session (using the login object) and then using that session to perform all operations,

To do this, you must specify the username and password in the request header of the NITRO request as follows:

X-NITRO-USER:<username>

X-NITRO-PASS:<password>

Note: In such cases, make sure that the request header DOES not include the following:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue>

`*Note:`*

Mandatory parameters are marked in redand placeholder content is marked in <green>.

get (all)

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/stat/spilloverpolicy

Query-parameters:

args

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/stat/spilloverpolicy?
args=name:<String_value>,detail:<Boolean_value>,fullvalues:<Boolean_value>,ntimes:<Double_value>

Use this query-parameter to get spilloverpolicy resources based on additional properties.

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "spilloverpolicy": [ {

"name":<String_value>,

"pipolicyundefhitsrate":<Double_value>,

"pipolicyundefhits":<Double_value>,

"pipolicyhitsrate":<Double_value>,

"pipolicyhits":<Double_value>

}]}
```

get

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/stat/spilloverpolicy/name_value><String>

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "spilloverpolicy": [ {

"name":<String_value>,

"pipolicyundefhitsrate":<Double_value>,
```

```
"pipolicyundefhits":<Double_value>,

"pipolicyhitsrate":<Double_value>,

"pipolicyhits":<Double_value>

}]}
```

ssl

Statistics for ssl.

Properties

(click to see Operations)

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
clearstats	<String>	Read-write	Clear the statsistics / counters. Possible values = basic, full
ssltothwdecbesecondary	<Double>	Read-only	Number of bytes decrypted on the back-end in hardware on secondary card.
sslhwdecbesecondaryrate	<Double>	Read-only	Rate (/s) counter for ssltothwdecbesecondary
ssltothwdecfesecondary	<Double>	Read-only	Number of bytes decrypted on the front-end in hardware on secondary card.

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
sslhwdecfesecondaryrate	<Double>	Read-only	Rate (/s) counter for ssltothwdecfesecondary
ssltothwencbesecondary	<Double>	Read-only	Number of bytes encrypted on the back-end in hardware on secondary card.
sslhwencbesecondaryrate	<Double>	Read-only	Rate (/s) counter for ssltothwencbesecondary
ssltothwencfesecondary	<Double>	Read-only	Number of bytes encrypted on the front-end in hardware on secondary card.
sslhwencfesecondaryrate	<Double>	Read-only	Rate (/s) counter for ssltothwencfesecondary
ssltotdechwsecondary	<Double>	Read-only	Number of bytes decrypted in hardware on secondary card.
ssldechwsecondaryrate	<Double>	Read-only	Rate (/s) counter for ssltotdechwsecondary
ssltotenchwsecondary	<Double>	Read-only	Number of bytes encrypted in hardware on secondary card.
sslenchwsecondaryrate	<Double>	Read-only	Rate (/s) counter for ssltotenchwsecondary

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
sslcryptoutilizationstat2nd	<Double>	Read-only	Utilization of the hardware crypto resource. Only valid values are 0-100%. Only works on N3FIPS+N3 hybrid platform.
sslcryptoutilizationstat	<Double>	Read-only	Utilization of the hardware crypto resource. Only valid values are 0-100%. Only works on platforms with Nitrox-3 chips.
sslnumcardsupsecondary	<Double>	Read-only	Number of Add on SSL cards that are UP.
sslcardssecondary	<Double>	Read-only	Number of secondary SSL crypto cards present on the Citrix ADC.
sslnumcardsup	<Double>	Read-only	Number of SSL cards that are UP. If the number of cards UP is lower than a threshold, a failover is initiated.
sslcardstatus	<Double>	Read-only	Status of the SSL card(s). The value should be interpreted in binary form, with each set bit indicates a card as UP.

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
sslcards	<Double>	Read-only	Number of SSL crypto cards present on the Citrix ADC.
sslenginestatus	<Double>	Read-only	State of the SSL Engine (1=UP/ 0=DOWN). This state is decided based on SSL Feature/License status and minimum number of cards UP.
ssltotsessions	<Double>	Read-only	Number of SSL sessions on the Citrix ADC.
sslsessionsrate	<Double>	Read-only	Rate (/s) counter for ssltotsessions
ssltottransactions	<Double>	Read-only	Number of SSL transactions on the Citrix ADC
ssltransactionsrate	<Double>	Read-only	Rate (/s) counter for ssltottransactions
ssltotsslv2transactions	<Double>	Read-only	Number of SSLv2 transactions on the Citrix ADC.
sslsslv2transactionsrate	<Double>	Read-only	Rate (/s) counter for ssltotsslv2transactions
ssltotsslv3transactions	<Double>	Read-only	Total number of SSLv3 transactions on the Citrix ADC.

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
sslsslv3transactionsrate	<Double>	Read-only	Rate (/s) counter for ssltotsslv3transactions
ssltottlsv1transactions	<Double>	Read-only	Number of TLSv1 transactions on the Citrix ADC.
ssltlsv1transactionsrate	<Double>	Read-only	Rate (/s) counter for ssltottlsv1transactions
ssltottlsv11transactions	<Double>	Read-only	Number of TLSv1.1 transactions on the Citrix ADC.
ssltlsv11transactionsrate	<Double>	Read-only	Rate (/s) counter for ssltottlsv11transactions
ssltottlsv12transactions	<Double>	Read-only	Number of TLSv1.2 transactions on the Citrix ADC.
ssltlsv12transactionsrate	<Double>	Read-only	Rate (/s) counter for ssltottlsv12transactions
ssltottlsv13transactions	<Double>	Read-only	Number of TLSv1.3 transactions on the Citrix ADC.
ssltlsv13transactionsrate	<Double>	Read-only	Rate (/s) counter for ssltottlsv13transactions
ssltotdtlsv1transactions	<Double>	Read-only	Number of DTLSv1 transactions on the Citrix ADC.

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
ssldtlsv1transactionsrate	<Double>	Read-only	Rate (/s) counter for ssltotdtlsv1transactions
ssltotsslv2sessions	<Double>	Read-only	Number of SSLv2 sessions on the Citrix ADC.
sslsslv2sessionsrate	<Double>	Read-only	Rate (/s) counter for ssltotsslv2sessions
ssltotsslv3sessions	<Double>	Read-only	Number of SSLv3 sessions on the Citrix ADC.
sslsslv3sessionsrate	<Double>	Read-only	Rate (/s) counter for ssltotsslv3sessions
ssltottlsv1sessions	<Double>	Read-only	Number of TLSv1 sessions on the Citrix ADC.
ssltlsv1sessionsrate	<Double>	Read-only	Rate (/s) counter for ssltottlsv1sessions
ssltottlsv11sessions	<Double>	Read-only	Number of TLSv1.1 sessions on the Citrix ADC.
ssltlsv11sessionsrate	<Double>	Read-only	Rate (/s) counter for ssltottlsv11sessions
ssltottlsv12sessions	<Double>	Read-only	Number of TLSv1.2 sessions on the Citrix ADC.
ssltlsv12sessionsrate	<Double>	Read-only	Rate (/s) counter for ssltottlsv12sessions

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
ssltottlsv13sessions	<Double>	Read-only	Number of TLSv1.3 sessions on the Citrix ADC.
ssltlsv13sessionsrate	<Double>	Read-only	Rate (/s) counter for ssltottlsv13sessions
ssltotdtlsv1sessions	<Double>	Read-only	Number of DTLSv1 sessions on the Citrix ADC.
ssldtlsv1sessionsrate	<Double>	Read-only	Rate (/s) counter for ssltotdtlsv1sessions
ssltotnewsessions	<Double>	Read-only	Number of new SSL sessions created on the Citrix ADC.
sslnewsessionsrate	<Double>	Read-only	Rate (/s) counter for ssltotnewsessions
ssltotsessionmiss	<Double>	Read-only	Number of SSL session reuse misses on the Citrix ADC.
sslsessionmissrate	<Double>	Read-only	Rate (/s) counter for ssltotsessionmiss
ssltotsessionhits	<Double>	Read-only	Number of SSL session reuse hits on the Citrix ADC.
sslsessionhitsrate	<Double>	Read-only	Rate (/s) counter for ssltotsessionhits
sslbetotsessions	<Double>	Read-only	Number of back-end SSL sessions on the Citrix ADC.

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
sslbesessionsrate	<Double>	Read-only	Rate (/s) counter for sslbetotsessions
sslbetotsslv3sessions	<Double>	Read-only	Number of back-end SSLv3 sessions on the Citrix ADC.
sslbesslv3sessionsrate	<Double>	Read-only	Rate (/s) counter for sslbetotsslv3sessions
sslbetottlsv1sessions	<Double>	Read-only	Number of back-end TLSv1 sessions on the Citrix ADC.
sslbetlsv1sessionsrate	<Double>	Read-only	Rate (/s) counter for sslbetottlsv1sessions
sslbetottlsv11sessions	<Double>	Read-only	Number of back-end TLSv1.1 sessions on the Citrix ADC.
sslbetlsv11sessionsrate	<Double>	Read-only	Rate (/s) counter for sslbetottlsv11sessions
sslbetottlsv12sessions	<Double>	Read-only	Number of back-end TLSv1.2 sessions on the Citrix ADC.
sslbetlsv12sessionsrate	<Double>	Read-only	Rate (/s) counter for sslbetottlsv12sessions
sslbetotdtlsv1sessions	<Double>	Read-only	Number of back-end DTLSv1 sessions on the Citrix ADC.
sslbedtlsv1sessionsrate	<Double>	Read-only	Rate (/s) counter for sslbetotdtlsv1sessions

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
sslbetotsessionmultiplexattempts	<Double>	Read-only	Number of back-end SSL session multiplex attempts on the Citrix ADC.
sslbesessionmultiplexattemptsrate	<Double>	Read-only	Rate (/s) counter for sslbetotsessionmultiplexattempts
sslbetotsessionmultiplexattemptsuccess	<Double>	Read-only	Number of back-end SSL session multiplex successes on the Citrix ADC.
sslbesessionmultiplexattemptsuccessrate	<Double>	Read-only	Rate (/s) counter for sslbetotsessionmultiplexattemptsuccess
sslbetotsessionmultiplexattemptfails	<Double>	Read-only	Number of back-end SSL session multiplex failures on the Citrix ADC.
sslbesessionmultiplexattemptfailsrate	<Double>	Read-only	Rate (/s) counter for sslbetotsessionmultiplexattemptfails
ssltotenc	<Double>	Read-only	Number of bytes encrypted on the Citrix ADC.
sslencrate	<Double>	Read-only	Rate (/s) counter for ssltotenc
ssltotdec	<Double>	Read-only	Number of bytes decrypted on the Citrix ADC.
ssldecrate	<Double>	Read-only	Rate (/s) counter for ssltotdec

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
ssltotsslserverinrecords	<Double>	Read-only	Number server in record on the Citrix ADC.
sslsslserverinrecordsrate	<Double>	Read-only	Rate (/s) counter for ssltotsslserverinrecords
ssltotrenegsessions	<Double>	Read-only	Number of SSL session renegotiations on the Citrix ADC.
sslrenegsessionsrate	<Double>	Read-only	Rate (/s) counter for ssltotrenegsessions
ssltotsslv3renegsessions	<Double>	Read-only	Number of session renegotiations done on SSLv3.
sslsslv3renegsessionsrate	<Double>	Read-only	Rate (/s) counter for ssltotsslv3renegsessions
ssltottlsv1renegsessions	<Double>	Read-only	Number of SSL session renegotiations done on TLSv1.
ssltlsv1renegsessionsrate	<Double>	Read-only	Rate (/s) counter for ssltottlsv1renegsessions
ssltottlsv11renegsessions	<Double>	Read-only	Number of SSL session renegotiations done on TLSv1.1.
ssltlsv11renegsessionsrate	<Double>	Read-only	Rate (/s) counter for ssltottlsv11renegsessions

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
ssltottlsv12renegsessions	<Double>	Read-only	Number of SSL session renegotiations done on TLSv1.2.
ssltlsv12renegsessionsrate	<Double>	Read-only	Rate (/s) counter for ssltottlsv12renegsessions
ssltotdtlsv1renegsessions	<Double>	Read-only	Number of SSL session renegotiations done on DTLSv1.
ssldtlsv1renegsessionsrate	<Double>	Read-only	Rate (/s) counter for ssltotdtlsv1renegsessions
ssltotrsa512keyexchanges	<Double>	Read-only	Number of RSA 512-bit key exchanges on the Citrix ADC.
sslrsa512keyexchangesrate	<Double>	Read-only	Rate (/s) counter for ssltotrsa512keyexchanges
ssltotrsa1024keyexchanges	<Double>	Read-only	Number of RSA 1024-bit key exchanges on the Citrix ADC.
sslrsa1024keyexchangesrate	<Double>	Read-only	Rate (/s) counter for ssltotrsa1024keyexchanges
ssltotrsa2048keyexchanges	<Double>	Read-only	Number of RSA 2048-bit key exchanges on the Citrix ADC.

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
sslrsa2048keyexchangesrate	<Double>	Read-only	Rate (/s) counter for ssltotrsa2048keyexchanges
ssltotrsa4096keyexchanges	<Double>	Read-only	Number of RSA 4096-bit key exchanges on the Citrix ADC.
sslrsa4096keyexchangesrate	<Double>	Read-only	Rate (/s) counter for ssltotrsa4096keyexchanges
ssltotdh512keyexchanges	<Double>	Read-only	Number of Diffie-Helman 512-bit key exchanges on the Citrix ADC.
ssldh512keyexchangesrate	<Double>	Read-only	Rate (/s) counter for ssltotdh512keyexchanges
ssltotdh1024keyexchanges	<Double>	Read-only	Number of Diffie-Helman 1024-bit key exchanges on the Citrix ADC.
ssldh1024keyexchangesrate	<Double>	Read-only	Rate (/s) counter for ssltotdh1024keyexchanges
ssltotdh2048keyexchanges	<Double>	Read-only	Number of Diffie-Helman 2048-bit key exchanges on the Citrix ADC.
ssldh2048keyexchangesrate	<Double>	Read-only	Rate (/s) counter for ssltotdh2048keyexchanges

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
ssltotdh4096keyexchanges	<Double>	Read-only	Number of Diffie-Helman 4096-bit key exchanges on the Citrix ADC.
ssldh4096keyexchangesrate	<Double>	Read-only	Rate (/s) counter for ssltotdh4096keyexchanges
ssltotecdhe521keyexchanges	<Double>	Read-only	Number of 521 Elliptical Curve Diffie-Helman on the Citrix ADC.
sslecdhe521keyexchangesrate	<Double>	Read-only	Rate (/s) counter for ssltotecdhe521keyexchanges
ssltotecdhe384keyexchanges	<Double>	Read-only	Number of 384 Elliptical Curve Diffie-Helman on the Citrix ADC.
sslecdhe384keyexchangesrate	<Double>	Read-only	Rate (/s) counter for ssltotecdhe384keyexchanges
ssltotecdhe256keyexchanges	<Double>	Read-only	Number of 256 Elliptical Curve Diffie-Helman on the Citrix ADC.
sslecdhe256keyexchangesrate	<Double>	Read-only	Rate (/s) counter for ssltotecdhe256keyexchanges
ssltotecdhe224keyexchanges	<Double>	Read-only	Number of 224 Elliptical Curve Diffie-Helman on the Citrix ADC.

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
sslecdhe224keyexchangesrate	<Double>	Read-only	Rate (/s) counter for ssltotecdhe224keyexchanges
ssltot40bitrc4ciphers	<Double>	Read-only	Number of RC4 40-bit cipher encryptions on the Citrix ADC.
ssl40bitrc4ciphersrate	<Double>	Read-only	Rate (/s) counter for ssltot40bitrc4ciphers
ssltot56bitrc4ciphers	<Double>	Read-only	Number of RC4 56-bit cipher encryptions on the Citrix ADC.
ssl56bitrc4ciphersrate	<Double>	Read-only	Rate (/s) counter for ssltot56bitrc4ciphers
ssltot64bitrc4ciphers	<Double>	Read-only	Number of RC4 64-bit cipher encryptions on the Citrix ADC.
ssl64bitrc4ciphersrate	<Double>	Read-only	Rate (/s) counter for ssltot64bitrc4ciphers
ssltot128bitrc4ciphers	<Double>	Read-only	Number of RC4 128-bit cipher encryptions on the Citrix ADC.
ssl128bitrc4ciphersrate	<Double>	Read-only	Rate (/s) counter for ssltot128bitrc4ciphers
ssltot40bitdesciphers	<Double>	Read-only	Number of DES 40-bit cipher encryptions on the Citrix ADC.
ssl40bitdesciphersrate	<Double>	Read-only	Rate (/s) counter for ssltot40bitdesciphers

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
ssltot56bitdesciphers	<Double>	Read-only	Number of DES 56-bit cipher encryptions on the Citrix ADC.
ssl56bitdesciphersrate	<Double>	Read-only	Rate (/s) counter for ssltot56bitdesciphers
ssltot168bit3desciphers	<Double>	Read-only	Number of DES 168-bit cipher encryptions on the Citrix ADC.
ssl168bit3desciphersrate	<Double>	Read-only	Rate (/s) counter for ssltot168bit3desciphers
ssltotcipheraes128	<Double>	Read-only	Number of AES 128-bit cipher encryptions on the Citrix ADC.
sslcipheraes128rate	<Double>	Read-only	Rate (/s) counter for ssltotcipheraes128
ssltotcipheraes256	<Double>	Read-only	Number of AES 256-bit cipher encryptions on the Citrix ADC.
sslcipheraes256rate	<Double>	Read-only	Rate (/s) counter for ssltotcipheraes256
ssltot40bitrc2ciphers	<Double>	Read-only	Number of RC2 40-bit cipher encryptions on the Citrix ADC.
ssl40bitrc2ciphersrate	<Double>	Read-only	Rate (/s) counter for ssltot40bitrc2ciphers
ssltot56bitrc2ciphers	<Double>	Read-only	Number of RC2 56-bit cipher encryptions on the Citrix ADC.

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
ssl56bitrc2ciphersrate	<Double>	Read-only	Rate (/s) counter for ssltot56bitrc2ciphers
ssltot128bitrc2ciphers	<Double>	Read-only	Number of RC2 128-bit cipher encryptions on the Citrix ADC.
ssl128bitrc2ciphersrate	<Double>	Read-only	Rate (/s) counter for ssltot128bitrc2ciphers
ssltot128bitaesgcmciphers	<Double>	Read-only	Number of AEC-GCM 128-bit cipher encryptions on the Citrix ADC.
ssl128bitaesgcmciphersrate	<Double>	Read-only	Rate (/s) counter for ssltot128bitaesgcmciphers
ssltot256bitaesgcmciphers	<Double>	Read-only	Number of AEC-GCM 256-bit cipher encryptions on the Citrix ADC.
ssl256bitaesgcmciphersrate	<Double>	Read-only	Rate (/s) counter for ssltot256bitaesgcmciphers
ssltotnullciphers	<Double>	Read-only	Number of Null cipher encryptions on the Citrix ADC.
sslnullciphersrate	<Double>	Read-only	Rate (/s) counter for ssltotnullciphers
ssltotmd5mac	<Double>	Read-only	Number of MD5 hashes on the Citrix ADC.

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
sslmd5macrate	<Double>	Read-only	Rate (/s) counter for ssltotmd5mac
ssltotshamac	<Double>	Read-only	Number of SHA hashes on the Citrix ADC.
sslshamacrate	<Double>	Read-only	Rate (/s) counter for ssltotshamac
ssltotsha256mac	<Double>	Read-only	Number of SHA256 hashes on the Citrix ADC.
sslsha256macrate	<Double>	Read-only	Rate (/s) counter for ssltotsha256mac
ssltotsha384mac	<Double>	Read-only	Number of SHA384 hashes on the Citrix ADC.
sslsha384macrate	<Double>	Read-only	Rate (/s) counter for ssltotsha384mac
ssltotsslv2handshakes	<Double>	Read-only	Number of handshakes on SSLv2 on the Citrix ADC.
sslsslv2handshakesrate	<Double>	Read-only	Rate (/s) counter for ssltotsslv2handshakes
ssltotsslv3handshakes	<Double>	Read-only	Number of handshakes on SSLv3 on the Citrix ADC.
sslsslv3handshakesrate	<Double>	Read-only	Rate (/s) counter for ssltotsslv3handshakes

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
ssltottlsv1handshakes	<Double>	Read-only	Number of SSL handshakes on TLSv1 on the Citrix ADC.
ssltlsv1handshakesrate	<Double>	Read-only	Rate (/s) counter for ssltottlsv1handshakes
ssltottlsv11handshakes	<Double>	Read-only	Number of SSL handshakes on TLSv1.1 on the Citrix ADC.
ssltlsv11handshakesrate	<Double>	Read-only	Rate (/s) counter for ssltottlsv11handshakes
ssltottlsv12handshakes	<Double>	Read-only	Number of SSL handshakes on TLSv1.2 on the Citrix ADC.
ssltlsv12handshakesrate	<Double>	Read-only	Rate (/s) counter for ssltottlsv12handshakes
ssltottlsv13handshakes	<Double>	Read-only	Number of SSL handshakes on TLSv1.3 on the Citrix ADC.
ssltlsv13handshakesrate	<Double>	Read-only	Rate (/s) counter for ssltottlsv13handshakes
ssltotdtlsv1handshakes	<Double>	Read-only	Number of SSL handshakes on DTLSv1 on the Citrix ADC.

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
ssldtlsv1handshakes rate	<Double>	Read-only	Rate (/s) counter for ssltotdtlsv1handshakes
ssltotsslv2clientauthentications	<Double>	Read-only	Number of client authentications done on SSLv2.
sslsslv2clientauthenticationsrate	<Double>	Read-only	Rate (/s) counter for ssltotsslv2clientauthentications
ssltotsslv3clientauthentications	<Double>	Read-only	Number of client authentications done on SSLv3.
sslsslv3clientauthenticationsrate	<Double>	Read-only	Rate (/s) counter for ssltotsslv3clientauthentications
ssltottlsv1clientauthentications	<Double>	Read-only	Number of client authentications done on TLSv1.
ssltlsv1clientauthenticationsrate	<Double>	Read-only	Rate (/s) counter for ssltottlsv1clientauthentications
ssltottlsv11clientauthentications	<Double>	Read-only	Number of client authentications done on TLSv1.1.
ssltlsv11clientauthenticationsrate	<Double>	Read-only	Rate (/s) counter for ssltottlsv11clientauthentications
ssltottlsv12clientauthentications	<Double>	Read-only	Number of client authentications done on TLSv1.2.

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
ssltlsv12clientauthenticationsrate	<Double>	Read-only	Rate (/s) counter for ssltottlsv12clientauthentications
ssltottlsv13clientauthentications	<Double>	Read-only	Number of client authentications done on TLSv1.3.
ssltlsv13clientauthenticationsrate	<Double>	Read-only	Rate (/s) counter for ssltottlsv13clientauthentications
ssltotdtlsv1clientauthentications	<Double>	Read-only	Number of client authentications done on DTLSv1.
ssldtlsv1clientauthenticationsrate	<Double>	Read-only	Rate (/s) counter for ssltotdtlsv1clientauthentications
ssltotrsaauthorizations	<Double>	Read-only	Number of RSA authentications on the Citrix ADC.
sslrsaauthorizationsrate	<Double>	Read-only	Rate (/s) counter for ssltotrsaauthorizations
ssltotdhauthorizations	<Double>	Read-only	Number of Diffie-Helman authentications on the Citrix ADC.
ssldhauthorizationsrate	<Double>	Read-only	Rate (/s) counter for ssltotdhauthorizations
ssltotdssauthorizations	<Double>	Read-only	Total number of times DSS authorization is used on the Citrix ADC.

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
ssldssauthorizationsrate	<Double>	Read-only	Rate (/s) counter for ssltotdssauthorizations
ssltotecdsaauthorizations	<Double>	Read-only	Total number of times ECDSA authorization is used on the Citrix ADC.
sslecdsaauthorizationsrate	<Double>	Read-only	Rate (/s) counter for ssltotecdsaauthorizations
ssltotnullauthorizations	<Double>	Read-only	Number of Null authentications on the Citrix ADC.
sslnullauthorizationsrate	<Double>	Read-only	Rate (/s) counter for ssltotnullauthorizations
ssltotbkendsessionrenegotiate	<Double>	Read-only	Number of back-end SSL session renegotiations on the Citrix ADC.
sslbkendsessionrenegotiaterate	<Double>	Read-only	Rate (/s) counter for ssltotbkendsessionrenegotiate
ssltotbkendsslv3renego	<Double>	Read-only	Number of back-end SSLv3 session renegotiations on the Citrix ADC.
sslbkendsslv3renegorate	<Double>	Read-only	Rate (/s) counter for ssltotbkendsslv3renego
ssltotbkendtlsvlrenego	<Double>	Read-only	Number of back-end TLSv1 session

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
			renegotiations on the Citrix ADC.
sslbkendtlsvlrenegorate	<Double>	Read-only	Rate (/s) counter for ssltotbkendtlsvlrenego
ssltotbkendtlsv11renego	<Double>	Read-only	Number of back-end TLSv1.1 session renegotiations on the Citrix ADC.
sslbkendtlsv11renegorate	<Double>	Read-only	Rate (/s) counter for ssltotbkendtlsv11renego
ssltotbkendtlsv12renego	<Double>	Read-only	Number of back-end TLSv1.2 session renegotiations on the Citrix ADC.
sslbkendtlsv12renegorate	<Double>	Read-only	Rate (/s) counter for ssltotbkendtlsv12renego
ssltotbkenddtlsvlrenego	<Double>	Read-only	Number of back-end DTLSv1 session renegotiations on the Citrix ADC.
sslbkenddtlsvlrenegorate	<Double>	Read-only	Rate (/s) counter for ssltotbkenddtlsvlrenego
sslbetotrsa512keyexchanges	<Double>	Read-only	Number of back-end RSA 512-bit key exchanges on the Citrix ADC.
sslbersa512keyexchangesrate	<Double>	Read-only	Rate (/s) counter for sslbetotrsa512keyexchanges

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
sslbetotrsa1024keyexchanges	<Double>	Read-only	Number of back-end RSA 1024-bit key exchanges on the Citrix ADC.
sslbersa1024keyexchangesrate	<Double>	Read-only	Rate (/s) counter for sslbetotrsa1024keyexchanges
sslbetotrsa2048keyexchanges	<Double>	Read-only	Number of back-end RSA 2048-bit key exchanges on the Citrix ADC.
sslbersa2048keyexchangesrate	<Double>	Read-only	Rate (/s) counter for sslbetotrsa2048keyexchanges
sslbetotrsa4096keyexchanges	<Double>	Read-only	Number of back-end RSA 4096-bit key exchanges on the Citrix ADC.
sslbersa4096keyexchangesrate	<Double>	Read-only	Rate (/s) counter for sslbetotrsa4096keyexchanges
sslbetotdh512keyexchanges	<Double>	Read-only	Number of back-end DH 512-bit key exchanges on the Citrix ADC.
sslbedh512keyexchangesrate	<Double>	Read-only	Rate (/s) counter for sslbetotdh512keyexchanges
sslbetotdh1024keyexchanges	<Double>	Read-only	Number of back-end DH 1024-bit key exchanges on the Citrix ADC.

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
sslbedh1024keyexchangesrate	<Double>	Read-only	Rate (/s) counter for sslbetotdh1024keyexchanges
sslbetotdh2048keyexchanges	<Double>	Read-only	Number of back-end DH 2048-bit key exchanges on the Citrix ADC.
sslbedh2048keyexchangesrate	<Double>	Read-only	Rate (/s) counter for sslbetotdh2048keyexchanges
sslbetotdh4096keyexchanges	<Double>	Read-only	Number of back-end DH 4096-bit key exchanges on the Citrix ADC.
sslbedh4096keyexchangesrate	<Double>	Read-only	Rate (/s) counter for sslbetotdh4096keyexchanges
sslbetotecdhecurve521	<Double>	Read-only	Number of back-end ECDHE 521 curve Key exchanges on the Citrix ADC.
sslbeecdhecurve521rate	<Double>	Read-only	Rate (/s) counter for sslbetotecdhecurve521
sslbetotecdhecurve384	<Double>	Read-only	Number of back-end ECDHE 384 curve Key exchanges on the Citrix ADC.
sslbeecdhecurve384rate	<Double>	Read-only	Rate (/s) counter for sslbetotecdhecurve384

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
sslbetotecdhecurve256	<Double>	Read-only	Number of back-end ECDHE 256 curve Key exchanges on the Citrix ADC.
sslbeecdhecurve256rate	<Double>	Read-only	Rate (/s) counter for sslbetotecdhecurve256
sslbetotecdhecurve224	<Double>	Read-only	Number of back-end ECDHE 224 curve Key exchanges on the Citrix ADC.
sslbeecdhecurve224rate	<Double>	Read-only	Rate (/s) counter for sslbetotecdhecurve224
sslbetot40bitrc4ciphers	<Double>	Read-only	Number of back-end RC4 40-bit cipher encryptions on the Citrix ADC.
sslbe40bitrc4ciphersrate	<Double>	Read-only	Rate (/s) counter for sslbetot40bitrc4ciphers
sslbetot56bitrc4ciphers	<Double>	Read-only	Number of back-end RC4 56-bit cipher encryptions on the Citrix ADC.
sslbe56bitrc4ciphersrate	<Double>	Read-only	Rate (/s) counter for sslbetot56bitrc4ciphers
sslbetot64bitrc4ciphers	<Double>	Read-only	Number of back-end RC4 64-bit cipher encryptions on the Citrix ADC.

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
sslbe64bitrc4ciphersrate	<Double>	Read-only	Rate (/s) counter for sslbetot64bitrc4ciphers
sslbetot128bitrc4ciphers	<Double>	Read-only	Number of back-end RC4 128-bit cipher encryptions on the Citrix ADC.
sslbe128bitrc4ciphersrate	<Double>	Read-only	Rate (/s) counter for sslbetot128bitrc4ciphers
sslbetot40bitdesciphers	<Double>	Read-only	Number of back-end DES 40-bit cipher encryptions on the Citrix ADC.
sslbe40bitdesciphersrate	<Double>	Read-only	Rate (/s) counter for sslbetot40bitdesciphers
sslbetot56bitdesciphers	<Double>	Read-only	Number of back-end DES 56-bit cipher encryptions on the Citrix ADC.
sslbe56bitdesciphersrate	<Double>	Read-only	Rate (/s) counter for sslbetot56bitdesciphers
sslbetot168bit3desciphers	<Double>	Read-only	Number of back-end 3DES 168-bit cipher encryptions on the Citrix ADC.
sslbe168bit3desciphersrate	<Double>	Read-only	Rate (/s) counter for sslbetot168bit3desciphers

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
ssltotbkendcipheraes128	<Double>	Read-only	Back-end AES 128-bit cipher encryptions on the Citrix ADC.
sslbkendcipheraes128rate	<Double>	Read-only	Rate (/s) counter for ssltotbkendcipheraes128
ssltotbkendcipheraes256	<Double>	Read-only	Back-end AES 256-bit cipher encryptions on the Citrix ADC.
sslbkendcipheraes256rate	<Double>	Read-only	Rate (/s) counter for ssltotbkendcipheraes256
sslbetot40bitrc2ciphers	<Double>	Read-only	Number of back-end RC2 40-bit cipher encryptions on the Citrix ADC.
sslbe40bitrc2ciphersrate	<Double>	Read-only	Rate (/s) counter for sslbetot40bitrc2ciphers
sslbetot56bitrc2ciphers	<Double>	Read-only	Number of back-end RC2 56-bit cipher encryptions on the Citrix ADC.
sslbe56bitrc2ciphersrate	<Double>	Read-only	Rate (/s) counter for sslbetot56bitrc2ciphers
sslbetot128bitrc2ciphers	<Double>	Read-only	Number of back-end RC2 128-bit cipher encryptions on the Citrix ADC.

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
sslbe128bitrc2ciphersrate	<Double>	Read-only	Rate (/s) counter for sslbetot128bitrc2ciphers
ssltotbkendcipheraesgcm128	<Double>	Read-only	Back-end AES-GCM 128-bit cipher encryptions on the Citrix ADC.
sslbkendcipheraesgcm128rate	<Double>	Read-only	Rate (/s) counter for ssltotbkendcipheraesgcm128
ssltotbkendcipheraesgcm256	<Double>	Read-only	Back-end AES-GCM 256-bit cipher encryptions on the Citrix ADC .
sslbkendcipheraesgcm256rate	<Double>	Read-only	Rate (/s) counter for ssltotbkendcipheraesgcm256
sslbetotnullciphers	<Double>	Read-only	Number of back-end null cipher encryptions on the Citrix ADC.
sslbenullciphersrate	<Double>	Read-only	Rate (/s) counter for sslbetotnullciphers
sslbetotmd5mac	<Double>	Read-only	Number of back-end MD5 hashes on the Citrix ADC.
sslbemd5macrate	<Double>	Read-only	Rate (/s) counter for sslbetotmd5mac
sslbetotshamac	<Double>	Read-only	Number of back-end SHA hashes on the Citrix ADC.

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
sslbeshamacrate	<Double>	Read-only	Rate (/s) counter for sslbetotshamac
sslbetotsha256mac	<Double>	Read-only	Number of back-end SHA256 hashes on the Citrix ADC.
sslbesha256macrate	<Double>	Read-only	Rate (/s) counter for sslbetotsha256mac
sslbetotsha384mac	<Double>	Read-only	Number of back-end SHA384 hashes on the Citrix ADC.
sslbesha384macrate	<Double>	Read-only	Rate (/s) counter for sslbetotsha384mac
sslbetotsslv3handshakes	<Double>	Read-only	Number of back-end SSLv3 handshakes on the Citrix ADC.
sslbesslv3handshakesrate	<Double>	Read-only	Rate (/s) counter for sslbetotsslv3handshakes
sslbetottlsv1handshakes	<Double>	Read-only	Number of back-end TLSv1 handshakes on the Citrix ADC.
sslbetlsv1handshakesrate	<Double>	Read-only	Rate (/s) counter for sslbetottlsv1handshakes
sslbetottlsv11handshakes	<Double>	Read-only	Number of back-end TLSv1.1 handshakes on the Citrix ADC.
sslbetlsv11handshakesrate	<Double>	Read-only	Rate (/s) counter for sslbetottlsv11handshakes

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
sslbetottlsv12handshakes	<Double>	Read-only	Number of back-end TLSv1.2 handshakes on the Citrix ADC.
sslbetlsv12handshakesrate	<Double>	Read-only	Rate (/s) counter for sslbetottlsv12handshakes
sslbetotdtlsv1handshakes	<Double>	Read-only	Number of back-end DTLSv1 handshakes on the Citrix ADC.
sslbedtlsv1handshakesrate	<Double>	Read-only	Rate (/s) counter for sslbetotdtlsv1handshakes
sslbetotsslv3clientauthentications	<Double>	Read-only	Number of back-end SSLv3 client authentications on the Citrix ADC.
sslbesslv3clientauthenticationsrate	<Double>	Read-only	Rate (/s) counter for sslbetotsslv3clientauthentications
sslbetottlsv1clientauthentications	<Double>	Read-only	Number of back-end TLSv1 client authentications on the Citrix ADC.
sslbetlsv1clientauthenticationsrate	<Double>	Read-only	Rate (/s) counter for sslbetottlsv1clientauthentications
sslbetottlsv11clientauthentications	<Double>	Read-only	Number of back-end TLSv1.1 client authentications on the Citrix ADC.

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
sslbetlsv11clientauthenticationsrate	<Double>	Read-only	Rate (/s) counter for sslbetottlsv11clientauthentications
sslbetottlsv12clientauthentications	<Double>	Read-only	Number of back-end TLSv1.2 client authentications on the Citrix ADC.
sslbetlsv12clientauthenticationsrate	<Double>	Read-only	Rate (/s) counter for sslbetottlsv12clientauthentications
sslbetotdtlsv1clientauthentications	<Double>	Read-only	Number of back-end DTLSv1 client authentications on the Citrix ADC.
sslbedtlsv1clientauthenticationsrate	<Double>	Read-only	Rate (/s) counter for sslbetotdtlsv1clientauthentications
sslbetotrsaauthorizations	<Double>	Read-only	Number of back-end RSA authentications on the Citrix ADC.
sslbersaauthorizationsrate	<Double>	Read-only	Rate (/s) counter for sslbetotrsaauthorizations
sslbetotdhauthorizations	<Double>	Read-only	Number of back-end DH authentications on the Citrix ADC.
sslbedhauthorizationsrate	<Double>	Read-only	Rate (/s) counter for sslbetotdhauthorizations
sslbetotdssauthorizations	<Double>	Read-only	Number of back-end DSS authentications on the Citrix ADC.

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
sslbedssauthorizationsrate	<Double>	Read-only	Rate (/s) counter for sslbetotdssauthorizations
sslbetotecdsaauthorizations	<Double>	Read-only	Number of back-end ECDSA authentications on the Citrix ADC.
sslbeecdsaauthorizationsrate	<Double>	Read-only	Rate (/s) counter for sslbetotecdsaauthorizations
sslbetotnullauthorizations	<Double>	Read-only	Number of back-end null authentications on the Citrix ADC.
sslbenullauthorizationsrate	<Double>	Read-only	Rate (/s) counter for sslbetotnullauthorizations
ssltotoffloadrsakeyexchanges	<Double>	Read-only	Number of RSA key exchanges offloaded to the cryptography card.
sslloffloadrsakeyexchangesrate	<Double>	Read-only	Rate (/s) counter for ssltotoffloadrsakeyexchanges
ssltotoffloadsignrsa	<Double>	Read-only	Number of RSA sign operations offloaded to the cryptography card.
sslloffloadsignrsarate	<Double>	Read-only	Rate (/s) counter for ssltotoffloadsignrsa
ssltotoffloaddhkeyexchanges	<Double>	Read-only	Number of DH key exchanges offloaded

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
			to the cryptography card.
ssloffloaddhkeyexchangesrate	<Double>	Read-only	Rate (/s) counter for ssltotoffloaddhkeyexchanges
ssltotoffloadbulkrc4	<Double>	Read-only	Number of RC4 encryptions offloaded to the cryptography card.
ssloffloadbulkrc4rate	<Double>	Read-only	Rate (/s) counter for ssltotoffloadbulkrc4
ssltotoffloadbulkdes	<Double>	Read-only	Number of DES encryptions offloaded to the cryptography card.
ssloffloadbulkdesrate	<Double>	Read-only	Rate (/s) counter for ssltotoffloadbulkdes
ssltotoffloadbulkaes	<Double>	Read-only	Number of AES encryptions offloaded to the cryptography card.
ssloffloadbulkaesrate	<Double>	Read-only	Rate (/s) counter for ssltotoffloadbulkaes
ssltotoffloadbulkaesgcm128	<Double>	Read-only	Number of AES-GCM 128-bit encryptions offloaded to the cryptography card.
ssloffloadbulkaesgcm128rate	<Double>	Read-only	Rate (/s) counter for ssltotoffloadbulkaesgcm128

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
ssltotoffloadbulkaesgcm256	<Double>	Read-only	Number of AES-GCM 256-bit encryptions offloaded to the cryptography card.
sslloffloadbulkaesgcm256rate	<Double>	Read-only	Rate (/s) counter for ssltotoffloadbulkaesgcm256
ssltotenchw	<Double>	Read-only	Number of bytes encrypted in hardware.
sslenchwrate	<Double>	Read-only	Rate (/s) counter for ssltotenchw
ssltotencsw	<Double>	Read-only	Number of bytes encrypted in software.
sslencswrate	<Double>	Read-only	Rate (/s) counter for ssltotencsw
ssltotencfe	<Double>	Read-only	Number of bytes encrypted on the front-end.
sslencferate	<Double>	Read-only	Rate (/s) counter for ssltotencfe
ssltothwencfe	<Double>	Read-only	Number of bytes encrypted in hardware on the front-end.
sslhwencferate	<Double>	Read-only	Rate (/s) counter for ssltothwencfe
ssltotswencfe	<Double>	Read-only	Number of bytes encrypted in

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
			software on the front-end.
sslswencferate	<Double>	Read-only	Rate (/s) counter for ssltotswencfe
ssltotencbe	<Double>	Read-only	Number of bytes encrypted on the back-end.
sslencberate	<Double>	Read-only	Rate (/s) counter for ssltotencbe
ssltothwencbe	<Double>	Read-only	Number of bytes encrypted in hardware on the back-end.
sslhwencberate	<Double>	Read-only	Rate (/s) counter for ssltothwencbe
ssltotswencbe	<Double>	Read-only	Number of bytes encrypted in software on the back-end.
sslswencberate	<Double>	Read-only	Rate (/s) counter for ssltotswencbe
ssltotdechw	<Double>	Read-only	Number of bytes decrypted in hardware.
ssldechwrate	<Double>	Read-only	Rate (/s) counter for ssltotdechw
ssltotdecsw	<Double>	Read-only	Number of bytes decrypted in software.
ssldecswrate	<Double>	Read-only	Rate (/s) counter for ssltotdecsw

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
ssltotdecfe	<Double>	Read-only	Number of bytes decrypted on the front-end.
ssldecferate	<Double>	Read-only	Rate (/s) counter for ssltotdecfe
ssltothwdecfe	<Double>	Read-only	Number of bytes decrypted in hardware on the front-end.
sslhwdecferate	<Double>	Read-only	Rate (/s) counter for ssltothwdecfe
ssltotswdecfe	<Double>	Read-only	Number of bytes decrypted in software on the front-end.
sslswdecferate	<Double>	Read-only	Rate (/s) counter for ssltotswdecfe
ssltotdecbe	<Double>	Read-only	Number of bytes decrypted on the back-end.
ssldecberate	<Double>	Read-only	Rate (/s) counter for ssltotdecbe
ssltothwdecbe	<Double>	Read-only	Number of bytes decrypted in hardware on the back-end.
sslhwdecberate	<Double>	Read-only	Rate (/s) counter for ssltothwdecbe
ssltotswdecbe	<Double>	Read-only	Number of bytes decrypted in

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
			software on the back-end
sslswdecberate	<Double>	Read-only	Rate (/s) counter for ssltotswdecbe
sslcursslinfospcbinusecount	<Double>	Read-only	Number of SPCB in use.
sslcursslinfospcbinusecountrate	<Double>	Read-only	Rate (/s) counter for sslcursslinfospcbinusecount
sslcursessions	<Double>	Read-only	Number of active SSL sessions on the Citrix ADC.
sslcursessionsrate	<Double>	Read-only	Rate (/s) counter for sslcursessions
sslcurqsize	<Double>	Read-only	Current queue size
sslcurqsizerate	<Double>	Read-only	Rate (/s) counter for sslcurqsize
sslcursslinfonscardinqcount	<Double>	Read-only	Number of current SSL card InQ count.
sslcursslinfonscardinqcountrate	<Double>	Read-only	Rate (/s) counter for sslcursslinfonscardinqcount
sslcursslinfocardinblkq	<Double>	Read-only	Number of current SSL card In BulkQ count.
sslcursslinfocardinblkqrate	<Double>	Read-only	Rate (/s) counter for sslcursslinfocardinblkq

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
sslcursslinfocardinkeyq	<Double>	Read-only	Number of current SSL card In KeyQ count.
sslcursslinfocardinkeyqrate	<Double>	Read-only	Rate (/s) counter for sslcursslinfocardinkeyq
sslbemaxmultiplexedsessions	<Double>	Read-only	Number of back-end SSL sessions reused on the Citrix ADC.
sslbemaxmultiplexedsessionsrate	<Double>	Read-only	Rate (/s) counter for sslbemaxmultiplexedsessions
ssltot128bitideaciphers	<Double>	Read-only	Number of IDEA 128-bit cipher encryptions on the Citrix ADC.
ssl128bitideaciphersrate	<Double>	Read-only	Rate (/s) counter for ssltot128bitideaciphers
sslbetot128bitideaciphers	<Double>	Read-only	Number of back-end IDEA 128-bit cipher encryptions on the Citrix ADC.
sslbe128bitideaciphersrate	<Double>	Read-only	Rate (/s) counter for sslbetot128bitideaciphers

Operations

(click to see Properties)

GET (ALL)

Some options that you can use for each operations:

- **Getting warnings in response:**NITRO allows you to get warnings in an operation by specifying the "warning" query parameter as "yes". For example, to get warnings while connecting to the Citrix ADC appliance, the URL is as follows:

  http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/login?warning=yes

  If any, the warnings are displayed in the response payload with the HTTP code "209 X-NITRO-WARNING".

- **Authenticated access for individual NITRO operations:**NITRO allows you to logon to the Citrix ADC appliance to perform individual operations. You can use this option instead of creating a NITRO session (using the login object) and then using that session to perform all operations,

  To do this, you must specify the username and password in the request header of the NITRO request as follows:

  X-NITRO-USER:<username>

  X-NITRO-PASS:<password>

  Note: In such cases, make sure that the request header DOES not include the following:

  Cookie:NITRO_AUTH_TOKEN=<tokenvalue>

`*Note:`*

Mandatory parameters are marked in redand placeholder content is marked in <green>.

get (all)

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/stat/ssl

Query-parameters:

args

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/stat/ssl?
args=detail:<Boolean_value>,fullvalues:<Boolean_value>,ntimes:<Double_value>,logfile:<String_value

Use this query-parameter to get ssl resources based on additional properties.

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "ssl": [ {

"ssltotdtlsv1renegsessions":<Double_value>,

"ssldtlsv1handshakesrate":<Double_value>,

"ssltlsv11handshakesrate":<Double_value>,

"ssldh512keyexchangesrate":<Double_value>,

"sslbetotsha384mac":<Double_value>,

"sslbetottlsv1handshakes":<Double_value>,

"sslbkenddtlsvlrenegorate":<Double_value>,

"sslswdecberate":<Double_value>,

"ssltottransactions":<Double_value>,

"sslbetot40bitrc4ciphers":<Double_value>,

"sslnumcardsup":<Double_value>,

"ssltothwencbesecondary":<Double_value>,

"sslbetlsv1clientauthenticationsrate":<Double_value>,

"sslrenegsessionsrate":<Double_value>,
```

```
"ssltottlsv11handshakes":<Double_value>,

"sslnumcardsupsecondary":<Double_value>,

"sslbetotsessions":<Double_value>,

"sslbetot64bitrc4ciphers":<Double_value>,

"sslbetotrsa4096keyexchanges":<Double_value>,

"sslbersa2048keyexchangesrate":<Double_value>,

"sslbetottlsv1clientauthentications":<Double_value>,

"ssltothwencbe":<Double_value>,

"sslbetotdh512keyexchanges":<Double_value>,

"ssltotrsaauthorizations":<Double_value>,

"sslecdhe384keyexchangesrate":<Double_value>,

"sslcipheraes128rate":<Double_value>,

"ssltotsslv2sessions":<Double_value>,

"sslhwencbesecondaryrate":<Double_value>,

"ssltothwdecfe":<Double_value>,

"ssloffloadbulkrc4rate":<Double_value>,

"ssltotbkendtlsv11renego":<Double_value>,

"sslbe128bitrc4ciphersrate":<Double_value>,

"sslbetotdh4096keyexchanges":<Double_value>,

"ssltotdecsw":<Double_value>,

"sslbedtlsv1clientauthenticationsrate":<Double_value>,

"ssl128bitrc2ciphersrate":<Double_value>,
```

```
"ssloffloadbulkaesrate":<Double_value>,

"sslrsa512keyexchangesrate":<Double_value>,

"sslbe56bitrc2ciphersrate":<Double_value>,

"ssltotrsa4096keyexchanges":<Double_value>,

"sslbetotsslv3clientauthentications":<Double_value>,

"sslbeecdhecurve256rate":<Double_value>,

"ssltotsslv2clientauthentications":<Double_value>,

"sslbetlsv11sessionsrate":<Double_value>,

"sslcurqsizerate":<Double_value>,

"sslbkendtlsv11renegorate":<Double_value>,

"ssltotdtlsv1clientauthentications":<Double_value>,

"ssldh1024keyexchangesrate":<Double_value>,

"ssltotmd5mac":<Double_value>,

"sslbetotnullauthorizations":<Double_value>,

"sslbesessionmultiplexattemptsrate":<Double_value>,

"sslsslv2handshakesrate":<Double_value>,

"sslbetlsv11handshakesrate":<Double_value>,

"ssldssauthorizationsrate":<Double_value>,

"sslbemaxmultiplexedsessionsrate":<Double_value>,

"sslcursslinfonscardinqcountrate":<Double_value>,

"sslbkendsessionrenegotiaterate":<Double_value>,

"sslbetotrsa512keyexchanges":<Double_value>,
```

```
"ssltotenchwsecondary":<Double_value>,

"ssltlsv1renegsessionsrate":<Double_value>,

"ssltotecdsaauthorizations":<Double_value>,

"ssltot168bit3desciphers":<Double_value>,

"sslbesha256macrate":<Double_value>,

"sslbeecdsaauthorizationsrate":<Double_value>,

"ssltottlsv1handshakes":<Double_value>,

"sslbetot56bitdesciphers":<Double_value>,

"sslbetottlsv12clientauthentications":<Double_value>,

"sslbe128bitrc2ciphersrate":<Double_value>,

"ssltottlsv12handshakes":<Double_value>,

"sslrsa2048keyexchangesrate":<Double_value>,

"ssltotsessionhits":<Double_value>,

"sslbkendcipheraesgcm256rate":<Double_value>,

"ssl40bitdesciphersrate":<Double_value>,

"ssltotencfe":<Double_value>,

"sslbesha384macrate":<Double_value>,

"ssl40bitrc4ciphersrate":<Double_value>,

"sslbe56bitrc4ciphersrate":<Double_value>,

"ssltottlsv11renegsessions":<Double_value>,

"ssltottlsv13handshakes":<Double_value>,

"ssldecferate":<Double_value>,
```

```
"ssltottlsv12sessions":<Double_value>,

"sslbedtlsv1handshakesrate":<Double_value>,

"sslbetotsha256mac":<Double_value>,

"sslsha384macrate":<Double_value>,

"ssltotrsa2048keyexchanges":<Double_value>,

"sslbetotsessionmultiplexattemptfails":<Double_value>,

"ssltotoffloadbulkdes":<Double_value>,

"ssltotnullciphers":<Double_value>,

"ssltlsv12sessionsrate":<Double_value>,

"ssltothwdecbe":<Double_value>,

"ssltotoffloadsignrsa":<Double_value>,

"sslswencferate":<Double_value>,

"sslbetotdtlsv1sessions":<Double_value>,

"ssltottlsv13clientauthentications":<Double_value>,

"sslbeecdhecurve224rate":<Double_value>,

"sslbetotshamac":<Double_value>,

"ssltotcipheraes128":<Double_value>,

"sslbetottlsv11handshakes":<Double_value>,

"ssltotbkendsslv3renego":<Double_value>,

"ssltot128bitrc4ciphers":<Double_value>,

"sslbetotdssauthorizations":<Double_value>,

"sslbetot56bitrc4ciphers":<Double_value>,
```

```
"sslbetot40bitdesciphers":<Double_value>,

"sslbetottlsv12handshakes":<Double_value>,

"ssltotsslv3renegsessions":<Double_value>,

"sslbersaauthorizationsrate":<Double_value>,

"ssltot128bitaesgcmciphers":<Double_value>,

"ssltotsslv2transactions":<Double_value>,

"sslbedh1024keyexchangesrate":<Double_value>,

"ssl64bitrc4ciphersrate":<Double_value>,

"sslenchwsecondaryrate":<Double_value>,

"sslsslv3clientauthenticationsrate":<Double_value>,

"sslcursslinfocardinkeyq":<Double_value>,

"sslbetotecdsaauthorizations":<Double_value>,

"ssltotdtlsv1transactions":<Double_value>,

"sslbetottlsv11sessions":<Double_value>,

"sslcardssecondary":<Double_value>,

"sslsessionmissrate":<Double_value>,

"ssltotbkenddtlsvlrenego":<Double_value>,

"ssltotdh4096keyexchanges":<Double_value>,

"sslbetotdhauthorizations":<Double_value>,

"sslbemd5macrate":<Double_value>,

"sslbetlsv12handshakesrate":<Double_value>,

"sslsslserverinrecordsrate":<Double_value>,
```

```
"sslecdhe256keyexchangesrate":<Double_value>,

"sslbetotmd5mac":<Double_value>,

"ssldhauthorizationsrate":<Double_value>,

"ssloffloadbulkaesgcm128rate":<Double_value>,

"sslshamacrate":<Double_value>,

"sslhwdecberate":<Double_value>,

"sslnullauthorizationsrate":<Double_value>,

"sslcursessions":<Double_value>,

"sslbetot128bitrc2ciphers":<Double_value>,

"sslsslv3sessionsrate":<Double_value>,

"ssltotdh1024keyexchanges":<Double_value>,

"ssltotbkendsessionrenegotiate":<Double_value>,

"sslbkendtlsv12renegorate":<Double_value>,

"ssldtlsv1sessionsrate":<Double_value>,

"sslsessionhitsrate":<Double_value>,

"sslcurqsize":<Double_value>,

"sslsslv3renegsessionsrate":<Double_value>,

"sslcryptoutilizationstat2nd":<Double_value>,

"ssltotdtlsv1handshakes":<Double_value>,

"ssldechwrate":<Double_value>,

"ssltotsslv3clientauthentications":<Double_value>,

"ssldecrate":<Double_value>,
```

```
"ssltlsv11clientauthenticationsrate":<Double_value>,

"ssldh2048keyexchangesrate":<Double_value>,

"sslbesslv3sessionsrate":<Double_value>,

"ssltotoffloadbulkaesgcm256":<Double_value>,

"ssl40bitrc2ciphersrate":<Double_value>,

"sslbe40bitdesciphersrate":<Double_value>,

"sslbetotsessionmultiplexattemptsuccess":<Double_value>,

"sslloffloadrsakeyexchangesrate":<Double_value>,

"sslbetotdh2048keyexchanges":<Double_value>,

"sslbesessionmultiplexattemptsuccessrate":<Double_value>,

"sslbe40bitrc4ciphersrate":<Double_value>,

"ssltotoffloadbulkaes":<Double_value>,

"sslbetlsv12clientauthenticationsrate":<Double_value>,

"sslbetot168bit3desciphers":<Double_value>,

"ssltottlsv12renegsessions":<Double_value>,

"ssltotrenegsessions":<Double_value>,

"sslcursslinfonscardinqcount":<Double_value>,

"sslbkendsslv3renegorate":<Double_value>,

"sslsslv3transactionsrate":<Double_value>,

"sslecdhe521keyexchangesrate":<Double_value>,

"ssldtlsv1clientauthenticationsrate":<Double_value>,

"ssltottlsv11clientauthentications":<Double_value>,
```

```
"sslsslv2clientauthenticationsrate":<Double_value>,

"sslbetotrsa2048keyexchanges":<Double_value>,

"sslrsaauthorizationsrate":<Double_value>,

"ssltot40bitrc2ciphers":<Double_value>,

"ssltotbkendcipheraesgcm128":<Double_value>,

"ssltot56bitrc2ciphers":<Double_value>,

"ssl128bitrc4ciphersrate":<Double_value>,

"ssltothwencfe":<Double_value>,

"ssltot40bitdesciphers":<Double_value>,

"ssldechwsecondaryrate":<Double_value>,

"sslbeshamacrate":<Double_value>,

"sslbkendcipheraes128rate":<Double_value>,

"ssltottlsv1transactions":<Double_value>,

"sslcursslinfocardinkeyqrate":<Double_value>,

"ssltotrsa512keyexchanges":<Double_value>,

"sslbetotnullciphers":<Double_value>,

"ssltlsv12renegsessionsrate":<Double_value>,

"sslbetotecdhecurve224":<Double_value>,

"ssltotencsw":<Double_value>,

"ssltothwdecbesecondary":<Double_value>,

"sslencrate":<Double_value>,

"ssltottlsv1renegsessions":<Double_value>,
```

```
"ssltotbkendcipheraes128":<Double_value>,

"sslbedh4096keyexchangesrate":<Double_value>,

"ssltlsv11transactionsrate":<Double_value>,

"sslbedtlsv1sessionsrate":<Double_value>,

"ssltlsv12clientauthenticationsrate":<Double_value>,

"ssltotcipheraes256":<Double_value>,

"ssltotsslv3transactions":<Double_value>,

"ssltotdecbe":<Double_value>,

"ssloffloadbulkdesrate":<Double_value>,

"ssltotbkendtlsv12renego":<Double_value>,

"sslbesessionsrate":<Double_value>,

"ssltot128bitideaciphers":<Double_value>,

"ssltotdec":<Double_value>,

"ssldtlsv1renegsessionsrate":<Double_value>,

"ssltotsslserverinrecords":<Double_value>,

"ssl56bitdesciphersrate":<Double_value>,

"sslbetot56bitrc2ciphers":<Double_value>,

"sslbkendcipheraes256rate":<Double_value>,

"sslbetotrsaauthorizations":<Double_value>,

"sslbedh2048keyexchangesrate":<Double_value>,

"sslbe168bit3desciphersrate":<Double_value>,

"sslbetottlsv11clientauthentications":<Double_value>,
```

```
"sslsslv2transactionsrate":<Double_value>,

"sslrsa1024keyexchangesrate":<Double_value>,

"sslhwdecbesecondaryrate":<Double_value>,

"sslbedh512keyexchangesrate":<Double_value>,

"ssltottlsv1clientauthentications":<Double_value>,

"ssldtlsv1transactionsrate":<Double_value>,

"sslbetot40bitrc2ciphers":<Double_value>,

"ssl168bit3desciphersrate":<Double_value>,

"ssltotswencfe":<Double_value>,

"ssltot64bitrc4ciphers":<Double_value>,

"ssldecswrate":<Double_value>,

"ssltot256bitaesgcmciphers":<Double_value>,

"ssltotdhauthorizations":<Double_value>,

"ssltotsessionmiss":<Double_value>,

"ssltotdecfe":<Double_value>,

"sslbesslv3handshakesrate":<Double_value>,

"sslencswrate":<Double_value>,

"sslbeecdhecurve521rate":<Double_value>,

"sslhwencferate":<Double_value>,

"sslbesessionmultiplexattemptfailsrate":<Double_value>,

"sslbe128bitideaciphersrate":<Double_value>,

"ssltotoffloaddhkeyexchanges":<Double_value>,
```

```
"ssltotecdhe224keyexchanges":<Double_value>,

"ssltlsv1clientauthenticationsrate":<Double_value>,

"ssltottlsv13sessions":<Double_value>,

"ssltot56bitrc4ciphers":<Double_value>,

"sslbe40bitrc2ciphersrate":<Double_value>,

"sslenchwrate":<Double_value>,

"sslbetlsv1handshakesrate":<Double_value>,

"sslbersa4096keyexchangesrate":<Double_value>,

"ssltotsslv3handshakes":<Double_value>,

"sslbenullciphersrate":<Double_value>,

"sslcursslinfospcbinusecountrate":<Double_value>,

"sslsessionsrate":<Double_value>,

"ssltottlsv12transactions":<Double_value>,

"ssltottlsv11sessions":<Double_value>,

"ssl56bitrc4ciphersrate":<Double_value>,

"ssldecberate":<Double_value>,

"sslbedssauthorizationsrate":<Double_value>,

"ssltot56bitdesciphers":<Double_value>,

"ssltothwencfesecondary":<Double_value>,

"sslbkendtlsvlrenegorate":<Double_value>,

"ssltotbkendcipheraesgcm256":<Double_value>,

"sslbenullauthorizationsrate":<Double_value>,
```

```
"ssltotenchw":<Double_value>,

"ssltottlsv13transactions":<Double_value>,

"ssltotdechwsecondary":<Double_value>,

"ssltlsv11sessionsrate":<Double_value>,

"sslbetotecdhecurve256":<Double_value>,

"ssltotdh512keyexchanges":<Double_value>,

"ssltlsv1sessionsrate":<Double_value>,

"ssltotencbe":<Double_value>,

"sslcryptoutilizationstat":<Double_value>,

"ssltotdtlsv1sessions":<Double_value>,

"ssltlsv13sessionsrate":<Double_value>,

"ssltotecdhe384keyexchanges":<Double_value>,

"ssltlsv13handshakesrate":<Double_value>,

"sslcipheraes256rate":<Double_value>,

"ssltotsslv2handshakes":<Double_value>,

"ssldh4096keyexchangesrate":<Double_value>,

"sslhwdecfesecondaryrate":<Double_value>,

"ssltotenc":<Double_value>,

"sslbetotsslv3sessions":<Double_value>,

"sslencberate":<Double_value>,

"ssltottlsv11transactions":<Double_value>,

"sslbemaxmultiplexedsessions":<Double_value>,
```

```
"ssl128bitaesgcmciphersrate":<Double_value>,

"sslbetotecdhecurve384":<Double_value>,

"ssltotsessions":<Double_value>,

"ssloffloadbulkaesgcm256rate":<Double_value>,

"ssltot40bitrc4ciphers":<Double_value>,

"ssltotecdhe256keyexchanges":<Double_value>,

"sslcardstatus":<Double_value>,

"sslnewsessionsrate":<Double_value>,

"sslbetot128bitideaciphers":<Double_value>,

"ssltotdechw":<Double_value>,

"ssltransactionsrate":<Double_value>,

"sslbkendcipheraesgcm128rate":<Double_value>,

"sslhwdecferate":<Double_value>,

"ssltotbkendtlsvlrenego":<Double_value>,

"sslbetotsslv3handshakes":<Double_value>,

"sslbetlsv1sessionsrate":<Double_value>,

"ssltlsv12transactionsrate":<Double_value>,

"sslcursslinfocardinblkq":<Double_value>,

"sslbetlsv12sessionsrate":<Double_value>,

"sslbetlsv11clientauthenticationsrate":<Double_value>,

"sslsslv3handshakesrate":<Double_value>,

"ssltotsha384mac":<Double_value>,
```

```
"sslbetotdtlsv1clientauthentications":<Double_value>,

"sslbetotsessionmultiplexattempts":<Double_value>,

"ssltlsv13transactionsrate":<Double_value>,

"ssltotdssauthorizations":<Double_value>,

"ssltotshamac":<Double_value>,

"ssltlsv13clientauthenticationsrate":<Double_value>,

"sslrsa4096keyexchangesrate":<Double_value>,

"ssltotswdecfe":<Double_value>,

"ssltotsha256mac":<Double_value>,

"ssltotrsa1024keyexchanges":<Double_value>,

"ssltotnullauthorizations":<Double_value>,

"sslhwencfesecondaryrate":<Double_value>,

"ssltotoffloadrsakeyexchanges":<Double_value>,

"sslecdsaauthorizationsrate":<Double_value>,

"sslbedhauthorizationsrate":<Double_value>,

"ssltotecdhe521keyexchanges":<Double_value>,

"sslmd5macrate":<Double_value>,

"sslswdecferate":<Double_value>,

"sslhwencberate":<Double_value>,

"sslcards":<Double_value>,

"ssltotoffloadbulkaesgcm128":<Double_value>,

"sslnullciphersrate":<Double_value>,
```

```
"sslbetot128bitrc4ciphers":<Double_value>,

"ssltotsslv3sessions":<Double_value>,

"ssltotdh2048keyexchanges":<Double_value>,

"ssltlsv1handshakesrate":<Double_value>,

"sslbetotdh1024keyexchanges":<Double_value>,

"ssltotswdecbe":<Double_value>,

"ssltotnewsessions":<Double_value>,

"sslecdhe224keyexchangesrate":<Double_value>,

"ssltottlsv1sessions":<Double_value>,

"ssltot128bitrc2ciphers":<Double_value>,

"sslbe64bitrc4ciphersrate":<Double_value>,

"sslbetotrsa1024keyexchanges":<Double_value>,

"sslbetottlsv1sessions":<Double_value>,

"sslbersa1024keyexchangesrate":<Double_value>,

"ssltotbkendcipheraes256":<Double_value>,

"sslenginestatus":<Double_value>,

"sslcursslinfocardinblkqrate":<Double_value>,

"ssltotswencbe":<Double_value>,

"sslbersa512keyexchangesrate":<Double_value>,

"sslbetottlsv12sessions":<Double_value>,

"ssltothwdecfesecondary":<Double_value>,

"ssltlsv1transactionsrate":<Double_value>,
```

```
"sslbetotecdhecurve521":<Double_value>,

"sslbe56bitdesciphersrate":<Double_value>,

"ssltlsv11renegsessionsrate":<Double_value>,

"ssltlsv12handshakesrate":<Double_value>,

"sslbeecdhecurve384rate":<Double_value>,

"sslcursessionsrate":<Double_value>,

"ssl56bitrc2ciphersrate":<Double_value>,

"sslbetotdtlsv1handshakes":<Double_value>,

"ssloffloaddhkeyexchangesrate":<Double_value>,

"sslsha256macrate":<Double_value>,

"sslsslv2sessionsrate":<Double_value>,

"sslswencberate":<Double_value>,

"sslencferate":<Double_value>,

"ssloffloadsignrsarate":<Double_value>,

"ssl256bitaesgcmciphersrate":<Double_value>,

"ssltottlsv12clientauthentications":<Double_value>,

"sslbesslv3clientauthenticationsrate":<Double_value>,

"ssltotoffloadbulkrc4":<Double_value>,

"sslcursslinfospcbinusecount":<Double_value>,

"ssl128bitideaciphersrate":<Double_value>

}]}
```

sslvserver

Statistics for SSL virtual server resource.

Properties

(click to see Operations)

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
vservername	<String>	Read-write	Name of the SSL virtual server for which to show detailed statistics. Minimum length = 1
clearstats	<String>	Read-write	Clear the statsistics / counters. Possible values = basic, full
vslbhealth	<Double>	Read-only	Health of the vserver. This gives percentage of UP services bound to this vserver.
primaryipaddress	<String>	Read-only	IP address of the vserver
primaryport	<Integer>	Read-only	The port on which the service is running.
type	<String>	Read-only	Protocol associated with the vserver
state	<String>	Read-only	Current state of the server. There are seven possible values: UP(7),

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
			DOWN(1), UNKNOWN(2), BUSY(3), OFS(Out of Service)(4), TROFS(Transition Out of Service)(5), TROFS_DOWN(Down When going Out of Service)(8)
actsvcs	<Double>	Read-only	number of ACTIVE services bound to a vserver
ssltotclientauthsuccess	<Double>	Read-only	Number of successful client authentication on this vserver
sslclientauthsuccessrate	<Double>	Read-only	Rate (/s) counter for ssltotclientauthsuccess
ssltotclientauthfailure	<Double>	Read-only	Number of failure client authentication on this vserver
sslclientauthfailurerate	<Double>	Read-only	Rate (/s) counter for ssltotclientauthfailure
sslctxtotencbytes	<Double>	Read-only	Number of encrypted bytes per SSL vserver
sslctxencbytesrate	<Double>	Read-only	Rate (/s) counter for sslctxtotencbytes
sslctxtotdecbytes	<Double>	Read-only	Number of decrypted bytes per SSL vserver

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
sslctxdecbytesrate	<Double>	Read-only	Rate (/s) counter for sslctxtotdecbytes
sslctxtothwencbytes	<Double>	Read-only	Number of hardware encrypted bytes per SSL vserver
sslctxhwencbytesrate	<Double>	Read-only	Rate (/s) counter for sslctxtothwencbytes
sslctxtothwdec_bytes	<Double>	Read-only	Number of hw decrypted bytes per SSL vserver
sslctxhwdec_bytesrate	<Double>	Read-only	Rate (/s) counter for sslctxtothwdec_bytes
sslctxtotsessionnew	<Double>	Read-only	Number of new sessions created
sslctxsessionnewrate	<Double>	Read-only	Rate (/s) counter for sslctxtotsessionnew
sslctxtotsessionhits	<Double>	Read-only	Number of session hits
sslctxsessionhitsrate	<Double>	Read-only	Rate (/s) counter for sslctxtotsessionhits

Operations

(click to see Properties)

GET (ALL)| GET

Some options that you can use for each operations:

- **Getting warnings in response:**NITRO allows you to get warnings in an operation by specifying the "warning" query parameter as "yes". For example, to get warnings while connecting to the Citrix ADC appliance, the URL is as follows:

  http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/login?warning=yes

If any, the warnings are displayed in the response payload with the HTTP code "209 X-NITRO-WARNING".

- **Authenticated access for individual NITRO operations:**NITRO allows you to logon to the Citrix ADC appliance to perform individual operations. You can use this option instead of creating a NITRO session (using the login object) and then using that session to perform all operations,

  To do this, you must specify the username and password in the request header of the NITRO request as follows:

  X-NITRO-USER:<username>

  X-NITRO-PASS:<password>

  Note: In such cases, make sure that the request header DOES not include the following:

  Cookie:NITRO_AUTH_TOKEN=<tokenvalue>

`*Note:`*

Mandatory parameters are marked in redand placeholder content is marked in <green>.

get (all)

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/stat/sslvserver

Query-parameters:

args

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/stat/sslvserver?
args=vservername:<String_value>,detail:<Boolean_value>,fullvalues:<Boolean_value>,ntimes:<Double

Use this query-parameter to get sslvserver resources based on additional properties.

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "sslvserver": [ {

"vservername":<String_value>,

"sslclientauthsuccessrate":<Double_value>,

"sslctxtothwdec_bytes":<Double_value>,

"sslctxtotencbytes":<Double_value>,

"sslctxtotsessionnew":<Double_value>,

"ssltotclientauthfailure":<Double_value>,

"sslctxdecbytesrate":<Double_value>,

"sslctxsessionnewrate":<Double_value>,

"sslclientauthfailurerate":<Double_value>,

"type":<String_value>,

"sslctxencbytesrate":<Double_value>,

"sslctxhwencbytesrate":<Double_value>,

"sslctxtotdecbytes":<Double_value>,

"primaryipaddress":<String_value>,

"sslctxtotsessionhits":<Double_value>,

"sslctxtothwencbytes":<Double_value>,
```

```
"sslctxsessionhitsrate":<Double_value>,

"state":<String_value>,

"vslbhealth":<Double_value>,

"actsvcs":<Double_value>,

"sslctxhwdec_bytesrate":<Double_value>,

"primaryport":<Integer_value>,

"ssltotclientauthsuccess":<Double_value>

}]}
```

get

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/stat/sslvserver/vservername_value><String>

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "sslvserver": [ {

"vservername":<String_value>,
```

```
"sslclientauthsuccessrate":<Double_value>,

"sslctxtothwdec_bytes":<Double_value>,

"sslctxtotencbytes":<Double_value>,

"sslctxtotsessionnew":<Double_value>,

"ssltotclientauthfailure":<Double_value>,

"sslctxdecbytesrate":<Double_value>,

"sslctxsessionnewrate":<Double_value>,

"sslclientauthfailurerate":<Double_value>,

"type":<String_value>,

"sslctxencbytesrate":<Double_value>,

"sslctxhwencbytesrate":<Double_value>,

"sslctxtotdecbytes":<Double_value>,

"primaryipaddress":<String_value>,

"sslctxtotsessionhits":<Double_value>,

"sslctxtothwencbytes":<Double_value>,

"sslctxsessionhitsrate":<Double_value>,

"state":<String_value>,

"vslbhealth":<Double_value>,

"actsvcs":<Double_value>,

"sslctxhwdec_bytesrate":<Double_value>,

"primaryport":<Integer_value>,

"ssltotclientauthsuccess":<Double_value>
```

```
}]}
```

SSL VPN

Virtual Private Network statistics.

NAME	DESCRIPTION
vpn	Statistics for vpn.
vpnurlpolicy	Statistics for VPN url policy resource.
vpnvserver	Statistics for VPN virtual server resource.

vpn

Statistics for vpn.

Properties

(click to see Operations)

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
clearstats	<String>	Read-write	Clear the statsistics / counters. Possible values = basic, full
indexhtmlhit	<Double>	Read-only	Number of requests for VPN login page.
indexhtmlnoserved	<Double>	Read-only	Number of failures to display VPN login page.
cfghtmlserved	<Double>	Read-only	Number of client configuration requests received by VPN server.

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
cfghtmlservedrate	<Double>	Read-only	Rate (/s) counter for cfghtmlserved
dnsreqhit	<Double>	Read-only	Number of DNS queries resolved by VPN server.
dnsreqhitrate	<Double>	Read-only	Rate (/s) counter for dnsreqhit
winsrequesthit	<Double>	Read-only	Number of WINS queries resolved by VPN server.
winsrequesthitrate	<Double>	Read-only	Rate (/s) counter for winsrequesthit
csrequesthit	<Double>	Read-only	Number of SSL VPN tunnels formed between VPN server and client.
csrequesthitrate	<Double>	Read-only	Rate (/s) counter for csrequesthit
csnonhttpprobehit	<Double>	Read-only	Number of probes from VPN to back-end non-HTTP servers that have been accessed by the VPN client.
csnonhttpprobehitrate	<Double>	Read-only	Rate (/s) counter for csnonhttpprobehit
cshttpprobehit	<Double>	Read-only	Number of probes from VPN to back-end HTTP servers that have been accessed by the VPN client.

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
cshttpprobehitrate	<Double>	Read-only	Rate (/s) counter for cshttpprobehit
totalcsconnsucc	<Double>	Read-only	Number of successful probes to all back-end servers.
csconnsuccrate	<Double>	Read-only	Rate (/s) counter for totalcsconnsucc
totalfsrequest	<Double>	Read-only	Number of file system requests received by VPN server.
fsrequestrate	<Double>	Read-only	Rate (/s) counter for totalfsrequest
vpnlicensefail	<Double>	Read-only	Number of users not able to login because of license unavailability.
iipdisabledmipused	<Double>	Read-only	Number of times SNIP is used as IIP is disabled.
iipdisabledmipusedrate	<Double>	Read-only	Rate (/s) counter for iipdisabledmipused
iipfailedmipused	<Double>	Read-only	Number of times SNIP is used as IIP assignment failed.
iipfailedmipusedrate	<Double>	Read-only	Rate (/s) counter for iipfailedmipused
iipspillovermipused	<Double>	Read-only	Number of times SNIP is used on IIP Spillover.

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
iipspillovermipusedrate	<Double>	Read-only	Rate (/s) counter for iipspillovermipused
iipdisabledmipdisabled	<Double>	Read-only	Both IIP and SNIP is disabled.
iipdisabledmipdisabledrate	<Double>	Read-only	Rate (/s) counter for iipdisabledmipdisabled
iipfailedmipdisabled	<Double>	Read-only	Number of times IIP assignment failed and SNIP is disabled.
iipfailedmipdisabledrate	<Double>	Read-only	Rate (/s) counter for iipfailedmipdisabled
socksmethreqrcvd	<Double>	Read-only	Number of received SOCKS method request.
socksmethreqrcvdrate	<Double>	Read-only	Rate (/s) counter for socksmethreqrcvd
socksmethreqsent	<Double>	Read-only	Number of sent SOCKS method request.
socksmethreqsentrate	<Double>	Read-only	Rate (/s) counter for socksmethreqsent
socksmethresprcvd	<Double>	Read-only	Number of received SOCKS method response.
socksmethresprcvdrate	<Double>	Read-only	Rate (/s) counter for socksmethresprcvd

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
socksmethrespsent	<Double>	Read-only	Number of sent SOCKS method response.
socksmethrespsentrate	<Double>	Read-only	Rate (/s) counter for socksmethrespsent
socksconnreqrcvd	<Double>	Read-only	Number of received SOCKS connect request.
socksconnreqrcvdrate	<Double>	Read-only	Rate (/s) counter for socksconnreqrcvd
socksconnreqsent	<Double>	Read-only	Number of sent SOCKS connect request.
socksconnreqsentrate	<Double>	Read-only	Rate (/s) counter for socksconnreqsent
socksconnresprcvd	<Double>	Read-only	Number of received SOCKS connect response.
socksconnresprcvdrate	<Double>	Read-only	Rate (/s) counter for socksconnresprcvd
socksconnrespsent	<Double>	Read-only	Number of sent SOCKS connect response.
socksconnrespsentrate	<Double>	Read-only	Rate (/s) counter for socksconnrespsent
socksservererror	<Double>	Read-only	Number of SOCKS server error.
socksservererrorrate	<Double>	Read-only	Rate (/s) counter for socksservererror

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
socksclienterror	<Double>	Read-only	Number of SOCKS client error.
socksclienterrorrate	<Double>	Read-only	Rate (/s) counter for socksclienterror
staconnsuccess	<Double>	Read-only	Number of STA connection success.
staconnsuccessrate	<Double>	Read-only	Rate (/s) counter for staconnsuccess
staconnfailure	<Double>	Read-only	Number of STA connection failure.
staconnfailurerate	<Double>	Read-only	Rate (/s) counter for staconnfailure
cpsconnsuccess	<Double>	Read-only	Number of CPS connection success.
cpsconnsuccessrate	<Double>	Read-only	Rate (/s) counter for cpsconnsuccess
cpsconnfailure	<Double>	Read-only	Number of CPS connection failure.
cpsconnfailurerate	<Double>	Read-only	Rate (/s) counter for cpsconnfailure
starequestsent	<Double>	Read-only	Number of STA request sent.
starequestsentrate	<Double>	Read-only	Rate (/s) counter for starequestsent
staresponserecvd	<Double>	Read-only	Number of STA response received.
staresponserecvdrate	<Double>	Read-only	Rate (/s) counter for staresponserecvd

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
icalicensefailure	<Double>	Read-only	Number of ICA license failure.
icalicensefailurerate	<Double>	Read-only	Rate (/s) counter for icalicensefailure
stamonsent	<Double>	Read-only	Number of STA monitor requests sent.
stamonsentrate	<Double>	Read-only	Rate (/s) counter for stamonsent
stamonrcvd	<Double>	Read-only	Number of STA monitor responses recieved.
stamonrcvdrate	<Double>	Read-only	Rate (/s) counter for stamonrcvd
stamonsucc	<Double>	Read-only	Number of STA monitor successful responses.
stamonsuccrate	<Double>	Read-only	Rate (/s) counter for stamonsucc
stamonfail	<Double>	Read-only	Number of STA monitor failed responses.
stamonfailrate	<Double>	Read-only	Rate (/s) counter for stamonfail
csgptktvalidatenotstarted	<Double>	Read-only	Total number of STA server lookup failures for auth-id in primary ticket

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
csgptktvalidatenotstartedrate	<Double>	Read-only	Rate (/s) counter for csgptktvalidatenotstarted
csgrtktvalidatenotstarted	<Double>	Read-only	Total number of STA server lookup failures for auth-id in redundant ticket
csgrtktvalidatenotstartedrate	<Double>	Read-only	Rate (/s) counter for csgrtktvalidatenotstarted

Operations

(click to see Properties)

GET (ALL)

Some options that you can use for each operations:

- **Getting warnings in response:**NITRO allows you to get warnings in an operation by specifying the "warning" query parameter as "yes". For example, to get warnings while connecting to the Citrix ADC appliance, the URL is as follows:

  http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/login?warning=yes

  If any, the warnings are displayed in the response payload with the HTTP code "209 X-NITRO-WARNING".

- **Authenticated access for individual NITRO operations:**NITRO allows you to logon to the Citrix ADC appliance to perform individual operations. You can use this option instead of creating a NITRO session (using the login object) and then using that session to perform all operations,

  To do this, you must specify the username and password in the request header of the NITRO request as follows:

  X-NITRO-USER:<username>

  X-NITRO-PASS:<password>

Note: In such cases, make sure that the request header DOES not include the following:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue>

`*Note:` *

Mandatory parameters are marked in redand placeholder content is marked in <green>.

get (all)

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/stat/vpn

Query-parameters:

args

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/stat/vpn?
args=detail:<Boolean_value>,fullvalues:<Boolean_value>,ntimes:<Double_value>,logfile:<String_value

Use this query-parameter to get vpn resources based on additional properties.

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "vpn": [ {

"staconnfailurerate":<Double_value>,

"totalcsconnsucc":<Double_value>,

"csnonhttpprobehitrate":<Double_value>,
```

```
"csgrtktvalidatenotstarted":<Double_value>,

"socksmethreqrcvd":<Double_value>,

"iipspillovermipusedrate":<Double_value>,

"iipfailedmipdisabled":<Double_value>,

"stamonsucc":<Double_value>,

"cshttpprobehitrate":<Double_value>,

"socksconnrespsentrate":<Double_value>,

"starequestsent":<Double_value>,

"socksservererrorrate":<Double_value>,

"csgrtktvalidatenotstartedrate":<Double_value>,

"winsrequesthitrate":<Double_value>,

"socksservererror":<Double_value>,

"socksmethresprcvdrate":<Double_value>,

"vpnlicensefail":<Double_value>,

"csgptktvalidatenotstarted":<Double_value>,

"socksconnreqrcvd":<Double_value>,

"totalfsrequest":<Double_value>,

"socksmethreqrcvdrate":<Double_value>,

"socksmethreqsentrate":<Double_value>,

"stamonrcvd":<Double_value>,

"cpsconnfailurerate":<Double_value>,

"iipdisabledmipdisabled":<Double_value>,
```

```
"cshttpprobehit":<Double_value>,

"iipspillovermipused":<Double_value>,

"stamonrcvdrate":<Double_value>,

"cpsconnsuccess":<Double_value>,

"stamonfailrate":<Double_value>,

"iipdisabledmipusedrate":<Double_value>,

"socksmethrespsentrate":<Double_value>,

"socksmethreqsent":<Double_value>,

"stamonsentrate":<Double_value>,

"icalicensefailurerate":<Double_value>,

"stamonsent":<Double_value>,

"staconnfailure":<Double_value>,

"socksconnreqrcvdrate":<Double_value>,

"socksclienterrorrate":<Double_value>,

"csgptktvalidatenotstartedrate":<Double_value>,

"csrequesthit":<Double_value>,

"winsrequesthit":<Double_value>,

"dnsreqhitrate":<Double_value>,

"socksconnresprcvd":<Double_value>,

"socksmethrespsent":<Double_value>,

"csconnsuccrate":<Double_value>,

"stamonsuccrate":<Double_value>,
```

```
"socksconnreqsent":<Double_value>,

"dnsreqhit":<Double_value>,

"staconnsuccessrate":<Double_value>,

"iipfailedmipusedrate":<Double_value>,

"staconnsuccess":<Double_value>,

"cfghtmlservedrate":<Double_value>,

"stamonfail":<Double_value>,

"staresponserecvdrate":<Double_value>,

"cpsconnfailure":<Double_value>,

"csrequesthitrate":<Double_value>,

"csnonhttpprobehit":<Double_value>,

"iipfailedmipdisabledrate":<Double_value>,

"cpsconnsuccessrate":<Double_value>,

"socksconnresprcvdrate":<Double_value>,

"indexhtmlnoserved":<Double_value>,

"fsrequestrate":<Double_value>,

"indexhtmlhit":<Double_value>,

"cfghtmlserved":<Double_value>,

"icalicensefailure":<Double_value>,

"staresponserecvd":<Double_value>,

"socksclienterror":<Double_value>,

"iipdisabledmipdisabledrate":<Double_value>,
```

```
"socksconnrespsent":<Double_value>,

"socksmethresprcvd":<Double_value>,

"iipdisabledmipused":<Double_value>,

"iipfailedmipused":<Double_value>,

"socksconnreqsentrate":<Double_value>,

"starequestsentrate":<Double_value>

}]}
```

vpnvserver

Statistics for VPN virtual server resource.

Properties

(click to see Operations)

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
name	<String>	Read-write	Name of the virtual server for which to show detailed statistics. Minimum length = 1
clearstats	<String>	Read-write	Clear the statsistics / counters. Possible values = basic, full
primaryipaddress	<String>	Read-only	The IP address on which the service is running.

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
primaryport	<Integer>	Read-only	The port on which the service is running.
type	<String>	Read-only	Protocol associated with the vserver
state	<String>	Read-only	Current state of the server. There are seven possible values: UP(7), DOWN(1), UNKNOWN(2), BUSY(3), OFS(Out of Service)(4), TROFS(Transition Out of Service)(5), TROFS_DOWN(Down When going Out of Service)(8)
totalrequests	<Double>	Read-only	Total number of requests received on this service or virtual server. (This applies to HTTP/SSL services and servers.)
requestsrate	<Double>	Read-only	Rate (/s) counter for totalrequests
totalresponses	<Double>	Read-only	Number of responses received on this service or virtual server. (This applies to HTTP/SSL services and servers.)

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
responsesrate	<Double>	Read-only	Rate (/s) counter for totalresponses
totalrequestbytes	<Double>	Read-only	Total number of request bytes received on this service or virtual server.
requestbytesrate	<Double>	Read-only	Rate (/s) counter for totalrequestbytes
totalresponsebytes	<Double>	Read-only	Number of response bytes received by this service or virtual server.
responsebytesrate	<Double>	Read-only	Rate (/s) counter for totalresponsebytes

Operations

(click to see Properties)

GET (ALL)| GET

Some options that you can use for each operations:

- **Getting warnings in response:**NITRO allows you to get warnings in an operation by specifying the "warning" query parameter as "yes". For example, to get warnings while connecting to the Citrix ADC appliance, the URL is as follows:

  http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/login?warning=yes

  If any, the warnings are displayed in the response payload with the HTTP code "209 X-NITRO-WARNING".

- **Authenticated access for individual NITRO operations:**NITRO allows you to logon to the Citrix ADC appliance to perform individual operations. You can use this option instead of creating a NITRO session (using the login object) and then using that session to perform all operations,

To do this, you must specify the username and password in the request header of the NITRO request as follows:

X-NITRO-USER:<username>

X-NITRO-PASS:<password>

Note: In such cases, make sure that the request header DOES not include the following:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue>

`*Note:`*

Mandatory parameters are marked in redand placeholder content is marked in <green>.

get (all)

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/stat/vpnvserver

Query-parameters:

args

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/stat/vpnvserver?
args=name:<String_value>,detail:<Boolean_value>,fullvalues:<Boolean_value>,ntimes:<Double_value>

Use this query-parameter to get vpnvserver resources based on additional properties.

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "vpnvserver": [ {

"name":<String_value>,

"totalrequests":<Double_value>,

"responsebytesrate":<Double_value>,

"totalresponses":<Double_value>,

"requestbytesrate":<Double_value>,

"type":<String_value>,

"primaryipaddress":<String_value>,

"responsesrate":<Double_value>,

"totalrequestbytes":<Double_value>,

"state":<String_value>,

"totalresponsebytes":<Double_value>,

"primaryport":<Integer_value>,

"requestsrate":<Double_value>

}]}
```

get

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/stat/vpnvserver/name_value><String>

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "vpnvserver": [ {

"name":<String_value>,

"totalrequests":<Double_value>,

"responsebytesrate":<Double_value>,

"totalresponses":<Double_value>,

"requestbytesrate":<Double_value>,

"type":<String_value>,

"primaryipaddress":<String_value>,

"responsesrate":<Double_value>,

"totalrequestbytes":<Double_value>,

"state":<String_value>,

"totalresponsebytes":<Double_value>,

"primaryport":<Integer_value>,

"requestsrate":<Double_value>

}]}
```

Stream

Stream statistics.

NAME	DESCRIPTION
streamidentifier	Statistics for identifier resource.

streamidentifier

Statistics for identifier resource.

Properties

(click to see Operations)

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
name	<String>	Read-write	Name of the stream identifier. Minimum length = 1
pattern	<String[]>	Read-write	Values on which grouping is performed are displayed in the output as row titles. If grouping is performed on two or more fields, their values are separated by a question mark in the row title. For example, consider a selector that contains the expressions HTTP.REQ.URL and CLIENT.IP.SRC (in that order), on an appliance that has accumulated

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
			records of a number of requests for two URLs, example.com/page1.html and example.com/page2.html, from two client IP addresses, 192.0.2.10 and 192.0.2.11. With a pattern of ? ?, the appliance performs grouping on both fields and displays statistics for the following: * Requests for example.com/abc.html from 192.0.2.10, with a row title of example.com/abc.html? 192.0.2.10. * Requests for example.com/abc.html from 192.0.2.11, with a row title of example.com/abc.html? 192.0.2.11.

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
			* Requests for example.com/def.html from 192.0.2.10, with a row title of example.com/def.html? 192.0.2.10. * Requests for example.com/def.html from 192.0.2.11, with a row title of example.com/def.html? 192.0.2.11. With a pattern of * ?, the appliance performs grouping on only the client IP address values and displays statistics for the following requests: * All requests from 192.0.2.10, with the IP address as the row title. * All requests from 192.0.2.11, with the IP address as the row title.

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
			With a pattern of ? *, the appliance performs grouping on only the URL values and displays statistics for the following requests: * All requests for example.com/abc.html, with the URL as the row title. * All requests for example.com/def.html, with the URL as the row title. With a pattern of * *, the appliance displays one set of collective statistics for all the requests received, with no row title. With a pattern of example.com/abc.html ?, the appliance displays statistics for requests for example.com/abc.html from each unique client IP address.

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
			With a pattern of * 192.0.2.11, the appliance displays statistics for all requests from 192.0.2.11.
clearstats	<String>	Read-write	Clear the statsistics / counters. Possible values = basic, full
sortby	<String>	Read-write	use this argument to sort by specific key. Possible values =
sortorder	<String>	Read-write	use this argument to specify sort order. Default value: SORT_DESCENDING Possible values = ascending, descending
streamobjreq	<Double>	Read-only	Total number of Stream Requests recieved.
streamobjbandw	<Double>	Read-only	Total Bandwidth consumed.
streamobjresptime	<Double>	Read-only	Average response time of the stream session.

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
streamobjconn	<Double>	Read-only	Current connections on the stream session.
streamobjbreachcnt	<Double>	Read-only	Total Breaching Transactions in configured interval.
streamobjpktcredits	<Double>	Read-only	Calculated Packet credits on current connection.
streamobjpktspersecond	<Double>	Read-only	Total packets recieved in configured interval.
streamobjdroppedconns	<Double>	Read-only	Total connections dropped.

Operations

(click to see Properties)

GET (ALL)| GET

Some options that you can use for each operations:

- **Getting warnings in response:**NITRO allows you to get warnings in an operation by specifying the "warning" query parameter as "yes". For example, to get warnings while connecting to the Citrix ADC appliance, the URL is as follows:

  http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/login?warning=yes

  If any, the warnings are displayed in the response payload with the HTTP code "209 X-NITRO-WARNING".
- **Authenticated access for individual NITRO operations:**NITRO allows you to logon to the Citrix ADC appliance to perform individual operations. You can use this option instead of creating a NITRO session (using the login object) and then using that session to perform all operations,

To do this, you must specify the username and password in the request header of the NITRO request as follows:

X-NITRO-USER:<username>

X-NITRO-PASS:<password>

Note: In such cases, make sure that the request header DOES not include the following:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue>

`*Note:`*

Mandatory parameters are marked in redand placeholder content is marked in <green>.

get (all)

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/stat/streamidentifier

Query-parameters:

args

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/stat/streamidentifier?
args=name:<String_value>,pattern:<String_value>,detail:<Boolean_value>,fullvalues:<Boolean_value>,ntimes:<Double_value>,logfile:<String_val

Use this query-parameter to get streamidentifier resources based on additional properties.

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "streamidentifier": [ {

"name":<String_value>,

"streamobjconn":<Double_value>,

"streamobjdroppedconns":<Double_value>,

"streamobjbreachcnt":<Double_value>,

"streamobjbandw":<Double_value>,

"streamobjresptime":<Double_value>,

"streamobjpktcredits":<Double_value>,

"streamobjpktspersecond":<Double_value>,

"streamobjreq":<Double_value>

}]}
```

get

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/stat/streamidentifier/name_value><String>

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "streamidentifier": [ {

"name":<String_value>,

"streamobjconn":<Double_value>,

"streamobjdroppedconns":<Double_value>,

"streamobjbreachcnt":<Double_value>,

"streamobjbandw":<Double_value>,

"streamobjresptime":<Double_value>,

"streamobjpktcredits":<Double_value>,

"streamobjpktspersecond":<Double_value>,

"streamobjreq":<Double_value>

}]}
```

Sure Connect

SureConnect statistics. SureConnect allows the system to deliver alternate content to clients when the system is under heavy load for the original URL. The original content will be served to the client, automatically, after an estimated delay period.

NAME	DESCRIPTION
sc	Statistics for sc.
scpolicy	Statistics for SureConnect policy resource.

sc

Statistics for sc.

Properties

(click to see Operations)

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
clearstats	<String>	Read-write	Clear the statsistics / counters. Possible values = basic, full
sctotcondtriggered	<Double>	Read-only	Number of times that SureConnect conditions were triggered.
sccondtriggeredrate	<Double>	Read-only	Rate (/s) counter for sctotcondtriggered
scpolicyurlhits	<Double>	Read-only	Total number of incoming requests that matched configured sureconnect policies.
scpolicyurlhitsrate	<Double>	Read-only	Rate (/s) counter for scpolicyurlhits
scpopups	<Double>	Read-only	Total number of in-memory java script served which throws the pop-up window.
scpopupsrate	<Double>	Read-only	Rate (/s) counter for scpopups
sctotreissuedrequests	<Double>	Read-only	Total number of reissued SureConnect requests.

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
screissuedrequestsrate	<Double>	Read-only	Rate (/s) counter for sctotreissuedrequests
scsessionreqs	<Double>	Read-only	Total number of requests that were handled in a single SureConnect session.
scsessionreqsrate	<Double>	Read-only	Rate (/s) counter for scsessionreqs
scaltconturls	<Double>	Read-only	Total number of alternate content served which throws the pop-up window.
scaltconturlsrate	<Double>	Read-only	Rate (/s) counter for scaltconturls
scpostreqs	<Double>	Read-only	Total number of HTTP POST requests that triggered SureConnect feature.
scpostreqsrate	<Double>	Read-only	Rate (/s) counter for scpostreqs
scresetstats	<Double>	Read-only	Toal number of times that SureConnect statistics were reset.
scresetstatsrate	<Double>	Read-only	Rate (/s) counter for scresetstats
scunsupbrow	<Double>	Read-only	Total number of requests that came

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
			from all unsupported browsers.
scunsupbrowrate	<Double>	Read-only	Rate (/s) counter for scunsupbrow
scfaultycookies	<Double>	Read-only	Total number of corrupted SureConnect cookies.
scfaultycookiesrate	<Double>	Read-only	Rate (/s) counter for scfaultycookies

Operations

(click to see Properties)

GET (ALL)

Some options that you can use for each operations:

- **Getting warnings in response:**NITRO allows you to get warnings in an operation by specifying the "warning" query parameter as "yes". For example, to get warnings while connecting to the Citrix ADC appliance, the URL is as follows:

  http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/login?warning=yes

  If any, the warnings are displayed in the response payload with the HTTP code "209 X-NITRO-WARNING".
- **Authenticated access for individual NITRO operations:**NITRO allows you to logon to the Citrix ADC appliance to perform individual operations. You can use this option instead of creating a NITRO session (using the login object) and then using that session to perform all operations,

  To do this, you must specify the username and password in the request header of the NITRO request as follows:

  X-NITRO-USER:<username>

  X-NITRO-PASS:<password>

Note: In such cases, make sure that the request header DOES not include the following:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue>

`*Note:` *

Mandatory parameters are marked in redand placeholder content is marked in <green>.

get (all)

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/stat/sc

Query-parameters:

args

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/stat/sc?
args=detail:<Boolean_value>,fullvalues:<Boolean_value>,ntimes:<Double_value>,logfile:<String_value

Use this query-parameter to get sc resources based on additional properties.

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "sc": [ {

"scpopups":<Double_value>,

"scfaultycookiesrate":<Double_value>,

"screissuedrequestsrate":<Double_value>,
```

```
"scpostreqs":<Double_value>,

"scaltconturls":<Double_value>,

"scresetstatsrate":<Double_value>,

"scpopupsrate":<Double_value>,

"sctotreissuedrequests":<Double_value>,

"scunsupbrow":<Double_value>,

"scpolicyurlhitsrate":<Double_value>,

"sccondtriggeredrate":<Double_value>,

"scpolicyurlhits":<Double_value>,

"scsessionreqsrate":<Double_value>,

"scresetstats":<Double_value>,

"sctotcondtriggered":<Double_value>,

"scfaultycookies":<Double_value>,

"scsessionreqs":<Double_value>,

"scaltconturlsrate":<Double_value>,

"scunsupbrowrate":<Double_value>,

"scpostreqsrate":<Double_value>

}]}
```

scpolicy

Statistics for SureConnect policy resource.

Properties

(click to see Operations)

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
name	<String>	Read-write	Name of the policy about which to display statistics. To display statistics about all SureConnect policies, do not set this parameter. Minimum length = 1 Maximum length = 31
clearstats	<String>	Read-write	Clear the statsistics / counters. Possible values = basic, full
avgservertransaction time	<Double>	Read-only	Average server transaction time in seconds for this SureConnect Policy.
avgservertransaction timerate	<Double>	Read-only	Rate (/s) counter for avgservertransaction time
scaverageclientttlb	<Double>	Read-only	Average value of the client Time-To-Last-Byte in seconds for this SureConnect policy.
scaverageclientttlbrate	<Double>	Read-only	Rate (/s) counter for scaverageclientttlb

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
scphysicalserviceip	<String>	Read-only	IP address of the service in dotted notation for which these statistics are maintained.
scphysicalservicepor t	<Integer>	Read-only	Port of the service for which these statistics are maintained.
sccurrentclientconne ctions	<Double>	Read-only	Number of clients currently allowed a server connection by this SureConnect policy.
sccurrentclientconne ctionsrate	<Double>	Read-only	Rate (/s) counter for sccurrentclientconne ctions
sccurrentwaitingclien ts	<Double>	Read-only	Current number of SureConnect priority clients that are waiting for a server connection.
sccurrentwaitingclien tsrate	<Double>	Read-only	Rate (/s) counter for sccurrentwaitingclien ts
totopenconn	<Double>	Read-only	Current number of open connections to the servers matching this policy.
openconnrate	<Double>	Read-only	Rate (/s) counter for totopenconn
sccurrentwaitingtime	<Double>	Read-only	Value of the currently estimated waiting

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
			time in seconds for the configured URL.
sccurrentwaitingtimerate	<Double>	Read-only	Rate (/s) counter for sccurrentwaitingtime
sctotalclientconnections	<Double>	Read-only	Total number of clients that were allowed a server connection by this SureConnect policy.
scclientconnectionsrate	<Double>	Read-only	Rate (/s) counter for sctotalclientconnections
sctotalserverconnections	<Double>	Read-only	Total number of server connections that were established through this SureConnect policy.
scserverconnectionsrate	<Double>	Read-only	Rate (/s) counter for sctotalserverconnections
totclienttransaction	<Double>	Read-only	Total number of client transactions processed by this SureConnect policy.
clienttransactionrate	<Double>	Read-only	Rate (/s) counter for totclienttransaction
sctotalservertransactions	<Double>	Read-only	Number of 200 OK responses received from the web server by this SureConnect policy.

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
scservertransactionsrate	<Double>	Read-only	Rate (/s) counter for sctotalservertransactions
sctotalrequestsreceived	<Double>	Read-only	Total number of requests received by this SureConnect policy.
screquestsreceivedrate	<Double>	Read-only	Rate (/s) counter for sctotalrequestsreceived
sctotalrequestbytes	<Double>	Read-only	Total number of request bytes received by this SureConnect policy.
screquestbytesrate	<Double>	Read-only	Rate (/s) counter for sctotalrequestbytes
sctotalresponsesreceived	<Double>	Read-only	Total number of server responses received by this SureConnect policy.
scresponsesreceivedrate	<Double>	Read-only	Rate (/s) counter for sctotalresponsesreceived
sctotalresponsebytes	<Double>	Read-only	Total number of response bytes received by this SureConnect policy.
scresponsebytesrate	<Double>	Read-only	Rate (/s) counter for sctotalresponsebytes

Operations

(click to see Properties)

GET (ALL)| GET

Some options that you can use for each operations:

- **Getting warnings in response:**NITRO allows you to get warnings in an operation by specifying the "warning" query parameter as "yes". For example, to get warnings while connecting to the Citrix ADC appliance, the URL is as follows:

  http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/login?warning=yes

  If any, the warnings are displayed in the response payload with the HTTP code "209 X-NITRO-WARNING".

- **Authenticated access for individual NITRO operations:**NITRO allows you to logon to the Citrix ADC appliance to perform individual operations. You can use this option instead of creating a NITRO session (using the login object) and then using that session to perform all operations,

  To do this, you must specify the username and password in the request header of the NITRO request as follows:

  X-NITRO-USER:<username>

  X-NITRO-PASS:<password>

  Note: In such cases, make sure that the request header DOES not include the following:

  Cookie:NITRO_AUTH_TOKEN=<tokenvalue>

`*Note:`*

Mandatory parameters are marked in redand placeholder content is marked in <green>.

get (all)

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/stat/scpolicy

Query-parameters:

args

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/stat/scpolicy?
args=name:<String_value>,detail:<Boolean_value>,fullvalues:<Boolean_value>,ntimes:<Double_value>

Use this query-parameter to get scpolicy resources based on additional properties.

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "scpolicy": [ {

"name":<String_value>,

"sctotalresponsesreceived":<Double_value>,

"sccurrentwaitingtime":<Double_value>,

"clienttransactionrate":<Double_value>,

"sccurrentwaitingtimerate":<Double_value>,

"sccurrentwaitingclients":<Double_value>,

"sctotalrequestsreceived":<Double_value>,

"screquestsreceivedrate":<Double_value>,

"scresponsebytesrate":<Double_value>,

"screquestbytesrate":<Double_value>,

"sctotalresponsebytes":<Double_value>,

"scserverconnectionsrate":<Double_value>,
```

```
"sctotalservertransactions":<Double_value>,

"sccurrentwaitingclientsrate":<Double_value>,

"scaverageclientttlb":<Double_value>,

"avgservertransactiontime":<Double_value>,

"scphysicalserviceip":<String_value>,

"sccurrentclientconnectionsrate":<Double_value>,

"sccurrentclientconnections":<Double_value>,

"openconnrate":<Double_value>,

"scclientconnectionsrate":<Double_value>,

"scaverageclientttlbrate":<Double_value>,

"totclienttransaction":<Double_value>,

"scservertransactionsrate":<Double_value>,

"sctotalclientconnections":<Double_value>,

"scphysicalserviceport":<Integer_value>,

"totopenconn":<Double_value>,

"avgservertransactiontimerate":<Double_value>,

"scresponsesreceivedrate":<Double_value>,

"sctotalrequestbytes":<Double_value>,

"sctotalserverconnections":<Double_value>

}]}
```

get

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/stat/scpolicy/name_value><String>

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "scpolicy": [ {

"name":<String_value>,

"sctotalresponsesreceived":<Double_value>,

"sccurrentwaitingtime":<Double_value>,

"clienttransactionrate":<Double_value>,

"sccurrentwaitingtimerate":<Double_value>,

"sccurrentwaitingclients":<Double_value>,

"sctotalrequestsreceived":<Double_value>,

"screquestsreceivedrate":<Double_value>,

"scresponsebytesrate":<Double_value>,

"screquestbytesrate":<Double_value>,

"sctotalresponsebytes":<Double_value>,
```

```
"scserverconnectionsrate":<Double_value>,

"sctotalservertransactions":<Double_value>,

"sccurrentwaitingclientsrate":<Double_value>,

"scaverageclientttlb":<Double_value>,

"avgservertransactiontime":<Double_value>,

"scphysicalserviceip":<String_value>,

"sccurrentclientconnectionsrate":<Double_value>,

"sccurrentclientconnections":<Double_value>,

"openconnrate":<Double_value>,

"scclientconnectionsrate":<Double_value>,

"scaverageclientttlbrate":<Double_value>,

"totclienttransaction":<Double_value>,

"scservertransactionsrate":<Double_value>,

"sctotalclientconnections":<Double_value>,

"scphysicalserviceport":<Integer_value>,

"totopenconn":<Double_value>,

"avgservertransactiontimerate":<Double_value>,

"scresponsesreceivedrate":<Double_value>,

"sctotalrequestbytes":<Double_value>,

"sctotalserverconnections":<Double_value>

}]}
```

system

Statistics for system.

Properties

(click to see Operations)

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
clearstats	<String>	Read-write	Clear the statsistics / counters. Possible values = basic, full
voltagev12n	<Double>	Read-only	Power supply -12V output. Acceptable range is -13.20 through -10.80 volts. 9800 and 9960 platforms display standard value of -12.0V.
voltagev5n	<Double>	Read-only	Power supply -5V output. Acceptable range is -5.50 through -4.50 volts. 9800 and 9960 platforms display standard value of -5.0V.
cpuusage	<Double>	Read-only	CPU utilization percentage.
rescpuusage	<Double>	Read-only	Shows average CPU utilization percentage if more than 1 CPU is present.

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
slavecpuusage	<Double>	Read-only	CPU 1 (currently the slave CPU) utilization, as percentage of capacity. Not applicable for a single-CPU system.
mastercpuusage	<Double>	Read-only	CPU 0 (currently the master CPU) utilization, as percentage of capacity.
auxvolt7	<Double>	Read-only	Voltage of a device connected to health monitoring chip through pin 7.
auxvolt6	<Double>	Read-only	Voltage of a device connected to health monitoring chip through pin 6.
auxvolt5	<Double>	Read-only	Voltage of a device connected to health monitoring chip through pin 5.
auxvolt4	<Double>	Read-only	Voltage of a device connected to health monitoring chip through pin 4.
auxvolt3	<Double>	Read-only	Voltage of a device connected to health monitoring chip through pin 3.
auxvolt2	<Double>	Read-only	Voltage of a device connected to health

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
			monitoring chip through pin 2.
auxvolt1	<Double>	Read-only	Voltage of a device connected to health monitoring chip through pin 1.
auxvolt0	<Double>	Read-only	Voltage of a device connected to health monitoring chip through pin 0.
voltagevsen2	<Double>	Read-only	Voltage Sensor 2 Input. Currently only 13k Platforms will have valid value for this counter and for older platforms this will be 0.
voltagev5sb	<Double>	Read-only	Power Supply 5V Standby Voltage. Currently only 13k Platforms will have valid value for this counter and for older platforms this will be 0.
voltagevtt	<Double>	Read-only	Intel CPU Vtt power. Currently only 13k Platforms will have valid value for this counter and for older platforms this will be 0.
voltagevbat	<Double>	Read-only	Onboard battery power supply output. 9800 and 9950 platforms display

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
			standard value of 5.0V.
voltagev12p	<Double>	Read-only	Power supply +12V output. Acceptable range is 10.80 through 13.20 volts.
voltagev5p	<Double>	Read-only	Power supply +5V output. Acceptable range is 4.50 through 5.50 volts.
voltagev33stby	<Double>	Read-only	Standby power supply +3.3V output. Acceptable range is 2.970 through 3.630 volts. 9800 and 9960 platforms display standard value of 3.3V. You can configure Standby 3.3V Supply Voltage by using the Set snmp alarm VOLTAGE-LOW command to set the lower limit and the Set snmp alarm VOLTAGE-HIGH command to set the upper limit.
voltagev33main	<Double>	Read-only	Main power supply +3.3V output. Acceptable range is 2.970 through 3.630 volts. This is a critical counter. You can configure Main 3.3V Supply Voltage, by

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
			using the Set snmp alarm VOLTAGE-LOW command to set the lower limit and the Set snmp alarm VOLTAGE-HIGH command to set the upper limit.
voltagevcc1	<Double>	Read-only	CPU core 1 voltage. Acceptable range is 1.080 through 1.650 volts. If CPU 1 is not connected to the health monitoring chip, display shows voltage of CPU 0.
voltagevcc0	<Double>	Read-only	CPU core 0 voltage. Acceptable range is 1.080 through 1.650 volts.
numcpus	<Double>	Read-only	The number of CPUs on the Citrix ADC.
memusagepcnt	<Double>	Read-only	Percentage of memory utilization on Citrix ADC.
memuseinmb	<Double>	Read-only	Main memory currently in use, in megabytes.
addimgmtcpuusagepcnt	<Double>	Read-only	Additonal Management CPU utilization percentage.

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
mgmtcpu0usagepcnt	<Double>	Read-only	Management CPU-0 utilization percentage.
mgmtcpuusagepcnt	<Double>	Read-only	Average Management CPU utilization percentage.
pktcpuusagepcnt	<Double>	Read-only	Average CPU utilization percentage for all packet engines excluding management PE.
cpuusagepcnt	<Double>	Read-only	CPU utilization percentage.
rescpuusagepcnt	<Double>	Read-only	Average CPU utilization percentage. Not applicable for a single-CPU system.
starttimelocal	<String>	Read-only	Time (in local timezone format) when the Citrix ADC was last started.
starttime	<String>	Read-only	Time when the Citrix ADC was last started.
disk0perusage	<Double>	Read-only	Used space in /flash partition of the disk, as a percentage. This is a critical counter. You can configure /flash Used (%) by using the Set snmp alarm DISK-

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
			USAGE-HIGH command.
disk1perusage	<Double>	Read-only	Used space in /var partition of the disk, as a percentage. This is a critical counter. You can configure /var Used (%) by using the Set snmp alarm DISK-USAGE-HIGH command.
cpufan0speed	<Double>	Read-only	CPU Fan 0 speed. Acceptable range is 3000 through 6000 RPM. This is a critical counter. You can configure CPU Fan 0 Speed by using the Set snmp alarm FAN-SPEED-LOW command to set the lower limit.
cpufan1speed	<Double>	Read-only	CPU Fan 1 speed. Acceptable range is 3000 through 6000 RPM. 7000 platform displays speed of CPU fan 0. This is a critical counter. You can configure CPU Fan 1 Speed by using the Set snmp alarm FAN-SPEED-LOW command to set the lower limit.

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
systemfanspeed	<Double>	Read-only	System fan speed. Acceptable range is 3000 through 6000 RPM. This is a critical counter. You can configure System Fan Speed by using the Set snmp alarm FAN-SPEED-LOW command to set the lower limit.
fan0speed	<Double>	Read-only	System fan 1 speed. For new platforms associated pin is connected to CPU supporting fans. For platforms in which it is not connected, it will point to System Fan.
fanspeed	<Double>	Read-only	System fan 2 speed. For new platforms associated pin is connected to CPU supporting fans. For platforms in which it is not connected, it will point to System Fan
cpu0temp	<Double>	Read-only	CPU 0 temperature. 9800 and 9960 platforms display internal chip temperature. This is a critical counter. You can configure CPU 0 Temperature by

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
			using the Set snmp alarm TEMPERATURE-HIGH command to set the upper limit.
cpu1temp	<Double>	Read-only	CPU 1 temperature. 9800 and 9960 platforms display internal chip temperature. 7000, 9010 and 10010 platforms display CPU 0 temperature. This is a critical counter. You can configure CPU 1 Temperature by using the Set snmp alarm TEMPERATURE-HIGH command to set the upper limit.
internaltemp	<Double>	Read-only	Internal temperature of health monitoring chip. This is a critical counter. You can configure Internal Temperature by using the Set snmp alarm TEMPERATURE-HIGH command to set the upper limit.
powersupply1status	<String>	Read-only	Power supply 1 failure status.
powersupply2status	<String>	Read-only	Power supply 2 failure status.

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
powersupply3status	<String>	Read-only	Power supply 3 failure status.
powersupply4status	<String>	Read-only	Power supply 4 failure status.
disk0size	<Double>	Read-only	Size of /flash partition of the hard disk.
disk0used	<Double>	Read-only	Used space in /flash partition of the hard disk.
disk0avail	<Double>	Read-only	Available space in /flash partition of the hard disk.
disk1size	<Double>	Read-only	Size of /var partition of the hard disk.
disk1used	<Double>	Read-only	Used space in /var partition of the hard disk.
disk1avail	<Double>	Read-only	Available space in /var partition of the hard disk.
fan2speed	<Double>	Read-only	Speed of Fan 0 if associated pin is connected to health monitoring chip.
fan3speed	<Double>	Read-only	Speed of Fan 1 if associated pin is connected to health monitoring chip.
fan4speed	<Double>	Read-only	Speed of Fan 2 if associated pin is

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
			connected to health monitoring chip.
fan5speed	<Double>	Read-only	Speed of Fan 3 if associated pin is connected to health monitoring chip.
auxtemp0	<Double>	Read-only	Temperature of a device connected to health monitoring chip through pin 0.
auxtemp1	<Double>	Read-only	Temperature of a device connected to health monitoring chip through pin 1.
auxtemp2	<Double>	Read-only	Temperature of a device connected to health monitoring chip through pin 2.
auxtemp3	<Double>	Read-only	Temperature of a device connected to health monitoring chip through pin 3.
timesincestart	<String>	Read-only	Seconds since the Citrix ADC started.
memsizemb	<Double>	Read-only	Total amount of system memory, in megabytes.

Operations

(click to see Properties)

GET (ALL)

Some options that you can use for each operations:

- **Getting warnings in response:**NITRO allows you to get warnings in an operation by specifying the "warning" query parameter as "yes". For example, to get warnings while connecting to the Citrix ADC appliance, the URL is as follows:

  http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/login?warning=yes

  If any, the warnings are displayed in the response payload with the HTTP code "209 X-NITRO-WARNING".

- **Authenticated access for individual NITRO operations:**NITRO allows you to logon to the Citrix ADC appliance to perform individual operations. You can use this option instead of creating a NITRO session (using the login object) and then using that session to perform all operations,

  To do this, you must specify the username and password in the request header of the NITRO request as follows:

  X-NITRO-USER:<username>

  X-NITRO-PASS:<password>

  Note: In such cases, make sure that the request header DOES not include the following:

  Cookie:NITRO_AUTH_TOKEN=<tokenvalue>

`*Note:`*

Mandatory parameters are marked in redand placeholder content is marked in <green>.

get (all)

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/stat/system

Query-parameters:

args

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/stat/system?
args=detail:<Boolean_value>,fullvalues:<Boolean_value>,ntimes:<Double_value>,logfile:<String_value

Use this query-parameter to get system resources based on additional properties.

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "system": [ {

"voltagevbat":<Double_value>,

"auxvolt2":<Double_value>,

"addimgmtcpuusagepcnt":<Double_value>,

"voltagev5n":<Double_value>,

"auxvolt7":<Double_value>,

"cpu0temp":<Double_value>,

"fan0speed":<Double_value>,

"disk1perusage":<Double_value>,

"rescpuusagepcnt":<Double_value>,

"disk1used":<Double_value>,

"disk1avail":<Double_value>,

"fan4speed":<Double_value>,

"rescpuusage":<Double_value>,

"powersupply3status":<String_value>,
```

```
"auxvolt5":<Double_value>,

"auxvolt3":<Double_value>,

"disk0perusage":<Double_value>,

"fan2speed":<Double_value>,

"powersupply4status":<String_value>,

"auxvolt1":<Double_value>,

"fanspeed":<Double_value>,

"fan5speed":<Double_value>,

"disk0size":<Double_value>,

"mgmtcpuusagepcnt":<Double_value>,

"cpuusage":<Double_value>,

"voltagev5sb":<Double_value>,

"disk0used":<Double_value>,

"powersupply1status":<String_value>,

"cpufan0speed":<Double_value>,

"disk1size":<Double_value>,

"auxtemp1":<Double_value>,

"numcpus":<Double_value>,

"pktcpuusagepcnt":<Double_value>,

"voltagev5p":<Double_value>,

"voltagevsen2":<Double_value>,

"auxvolt0":<Double_value>,
```

```
"auxtemp2":<Double_value>,

"memsizemb":<Double_value>,

"voltagev33main":<Double_value>,

"cpu1temp":<Double_value>,

"voltagev12n":<Double_value>,

"memuseinmb":<Double_value>,

"auxtemp3":<Double_value>,

"internaltemp":<Double_value>,

"voltagev12p":<Double_value>,

"disk0avail":<Double_value>,

"voltagev33stby":<Double_value>,

"voltagevcc1":<Double_value>,

"fan3speed":<Double_value>,

"voltagevtt":<Double_value>,

"auxtemp0":<Double_value>,

"cpufan1speed":<Double_value>,

"voltagevcc0":<Double_value>,

"auxvolt4":<Double_value>,

"starttime":<String_value>,

"systemfanspeed":<Double_value>,

"starttimelocal":<String_value>,

"cpuusagepcnt":<Double_value>,
```

```
"mastercpuusage":<Double_value>,

"timesincestart":<String_value>,

"mgmtcpu0usagepcnt":<Double_value>,

"auxvolt6":<Double_value>,

"slavecpuusage":<Double_value>,

"memusagepcnt":<Double_value>,

"powersupply2status":<String_value>

}]}
```

systembw

Statistics for bw resource.

Properties

(click to see Operations)

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
clearstats	<String>	Read-write	Clear the statsistics / counters. Possible values = basic, full
httpcltpoolinactive	<Double>	Read-only	No of requests sent from BW client.
httpcltpooloutactive	<Double>	Read-only	No of responses Received.
httpsvr200okresp	<Double>	Read-only	Number of 200 Ok response sent from the BW appliance.

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
httpsvr200okresprate	<Double>	Read-only	Rate (/s) counter for httpsvr200okresp
httpsvr404notfound	<Double>	Read-only	Number of 404 Not Found responses sent
httpsvr404notfoundrate	<Double>	Read-only	Rate (/s) counter for httpsvr404notfound
httpclterrstray	<Double>	Read-only	Number of stray packets received from server without HTTP request
httpclterrstrayrate	<Double>	Read-only	Rate (/s) counter for httpclterrstray
httpcltttfplwm	<Double>	Read-only	Number of Responses Falling on LWM for TTFP.
httpcltttfplwmrate	<Double>	Read-only	Rate (/s) counter for httpcltttfplwm
httpcltttfp_0	<Double>	Read-only	Number of Responses Falling on Band-0 for TTFP.
httpcltttfp_0rate	<Double>	Read-only	Rate (/s) counter for httpcltttfp_0
httpcltttfp_1	<Double>	Read-only	Number of Responses Falling on Band-1 for TTFP.
httpcltttfp_1rate	<Double>	Read-only	Rate (/s) counter for httpcltttfp_1

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
httpcltttfp_2	<Double>	Read-only	Number of Responses Falling on Band-2 for TTFP.
httpcltttfp_2rate	<Double>	Read-only	Rate (/s) counter for httpcltttfp_2
httpcltttfp_3	<Double>	Read-only	Number of Responses Falling on Band-3 for TTFP.
httpcltttfp_3rate	<Double>	Read-only	Rate (/s) counter for httpcltttfp_3
httpcltttfp_4	<Double>	Read-only	Number of Responses Falling on Band-4 for TTFP.
httpcltttfp_4rate	<Double>	Read-only	Rate (/s) counter for httpcltttfp_4
httpcltttfp_5	<Double>	Read-only	Number of Responses Falling on Band-5 for TTFP.
httpcltttfp_5rate	<Double>	Read-only	Rate (/s) counter for httpcltttfp_5
httpcltttfp_6	<Double>	Read-only	Number of Responses Falling on Band-6 for TTFP.
httpcltttfp_6rate	<Double>	Read-only	Rate (/s) counter for httpcltttfp_6
httpcltttfp_7	<Double>	Read-only	Number of Responses Falling on Band-7 for TTFP.
httpcltttfp_7rate	<Double>	Read-only	Rate (/s) counter for httpcltttfp_7

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
httpclttfphwm	<Double>	Read-only	Number of Responses Falling on HWM for TTFP.
httpclttfphwmrate	<Double>	Read-only	Rate (/s) counter for httpclttfphwm
httpclttfpmax	<Double>	Read-only	Peak RTT observed for Time to First response packet.
httpclttlplwm	<Double>	Read-only	Number of Responses Falling on LWM for TTLP.
httpclttlplwmrate	<Double>	Read-only	Rate (/s) counter for httpclttlplwm
httpclttlp_0	<Double>	Read-only	Number of Responses Falling on Band-0 for TTLP.
httpclttlp_0rate	<Double>	Read-only	Rate (/s) counter for httpclttlp_0
httpclttlp_1	<Double>	Read-only	Number of Responses Falling on Band-1 for TTLP.
httpclttlp_1rate	<Double>	Read-only	Rate (/s) counter for httpclttlp_1
httpclttlp_2	<Double>	Read-only	Number of Responses Falling on Band-2 for TTLP.
httpclttlp_2rate	<Double>	Read-only	Rate (/s) counter for httpclttlp_2

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
httpclttlp_3	<Double>	Read-only	Number of Responses Falling on Band-3 for TTLP.
httpclttlp_3rate	<Double>	Read-only	Rate (/s) counter for httpclttlp_3
httpclttlp_4	<Double>	Read-only	Number of Responses Falling on Band-4 for TTLP.
httpclttlp_4rate	<Double>	Read-only	Rate (/s) counter for httpclttlp_4
httpclttlp_5	<Double>	Read-only	Number of Responses Falling on Band-5 for TTLP.
httpclttlp_5rate	<Double>	Read-only	Rate (/s) counter for httpclttlp_5
httpclttlp_6	<Double>	Read-only	Number of Responses Falling on Band-6 for TTLP.
httpclttlp_6rate	<Double>	Read-only	Rate (/s) counter for httpclttlp_6
httpclttlp_7	<Double>	Read-only	Number of Responses Falling on Band-7 for TTLP.
httpclttlp_7rate	<Double>	Read-only	Rate (/s) counter for httpclttlp_7
httpclttlphwm	<Double>	Read-only	Number of Responses Falling on HWM for TTLP.
httpclttlphwmrate	<Double>	Read-only	Rate (/s) counter for httpclttlphwm

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
httpclttlpmax	<Double>	Read-only	Peak RTT observed for Time to Last response packet.

Operations

(click to see Properties)

GET (ALL)

Some options that you can use for each operations:

- **Getting warnings in response:**NITRO allows you to get warnings in an operation by specifying the "warning" query parameter as "yes". For example, to get warnings while connecting to the Citrix ADC appliance, the URL is as follows:

  http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/login?warning=yes

  If any, the warnings are displayed in the response payload with the HTTP code "209 X-NITRO-WARNING".

- **Authenticated access for individual NITRO operations:**NITRO allows you to logon to the Citrix ADC appliance to perform individual operations. You can use this option instead of creating a NITRO session (using the login object) and then using that session to perform all operations,

  To do this, you must specify the username and password in the request header of the NITRO request as follows:

  X-NITRO-USER:<username>

  X-NITRO-PASS:<password>

  Note: In such cases, make sure that the request header DOES not include the following:

  Cookie:NITRO_AUTH_TOKEN=<tokenvalue>

`*Note:`*

Mandatory parameters are marked in redand placeholder content is marked in <green>.

get (all)

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/stat/systembw

Query-parameters:

args

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/stat/systembw?
args=detail:<Boolean_value>,fullvalues:<Boolean_value>,ntimes:<Double_value>,logfile:<String_value

Use this query-parameter to get systembw resources based on additional properties.

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "systembw": [ {

"httpclttfp_4rate":<Double_value>,

"httpclttlp_5rate":<Double_value>,

"httpclttfp_2rate":<Double_value>,

"httpclttfp_7rate":<Double_value>,

"httpclttlphwmrate":<Double_value>,

"httpclttfp_0rate":<Double_value>,
```

```
"httpclttlplwmrate":<Double_value>,

"httpclttfp_4":<Double_value>,

"httpcltpooloutactive":<Double_value>,

"httpclttlplwm":<Double_value>,

"httpclttlp_1":<Double_value>,

"httpsvr404notfoundrate":<Double_value>,

"httpclttlp_6":<Double_value>,

"httpclttlphwm":<Double_value>,

"httpclttlp_1rate":<Double_value>,

"httpcltpoolinactive":<Double_value>,

"httpclttlp_3":<Double_value>,

"httpclttlp_5":<Double_value>,

"httpclttlp_6rate":<Double_value>,

"httpsvr200okresprate":<Double_value>,

"httpclttfp_1rate":<Double_value>,

"httpclttfp_1":<Double_value>,

"httpclttfphwmrate":<Double_value>,

"httpclttfp_6":<Double_value>,

"httpclttfpmax":<Double_value>,

"httpclttfp_5":<Double_value>,

"httpclttlpmax":<Double_value>,

"httpclttlp_4rate":<Double_value>,
```

```
"httpclttlp_7rate":<Double_value>,

"httpclttfp_0":<Double_value>,

"httpclttfp_3":<Double_value>,

"httpclttfphwm":<Double_value>,

"httpclterrstray":<Double_value>,

"httpclttlp_0rate":<Double_value>,

"httpclttfplwm":<Double_value>,

"httpclttlp_0":<Double_value>,

"httpclterrstrayrate":<Double_value>,

"httpclttfplwmrate":<Double_value>,

"httpclttlp_2":<Double_value>,

"httpclttfp_2":<Double_value>,

"httpclttlp_2rate":<Double_value>,

"httpclttlp_4":<Double_value>,

"httpclttfp_7":<Double_value>,

"httpsvr200okresp":<Double_value>,

"httpsvr404notfound":<Double_value>,

"httpclttlp_3rate":<Double_value>,

"httpclttfp_6rate":<Double_value>,

"httpclttfp_5rate":<Double_value>,

"httpclttfp_3rate":<Double_value>,

"httpclttlp_7":<Double_value>
```

```
}]}
```

systemcpu

Statistics for cpu resource.

Properties

(click to see Operations)

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
id	<Double>	Read-write	ID of the CPU for which to display statistics. Default value: 65535 Minimum value = 0 Maximum value = 65534
clearstats	<String>	Read-write	Clear the statsistics / counters. Possible values = basic, full
percpuuse	<Double>	Read-only	CPU utilization percentage.

Operations

(click to see Properties)

GET (ALL)| GET

Some options that you can use for each operations:

- **Getting warnings in response:**NITRO allows you to get warnings in an operation by specifying the "warning" query parameter as "yes". For example, to get warnings while connecting to the Citrix ADC appliance, the URL is as follows:

  http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/login?warning=yes

  If any, the warnings are displayed in the response payload with the HTTP code "209 X-NITRO-WARNING".

- **Authenticated access for individual NITRO operations:**NITRO allows you to logon to the Citrix ADC appliance to perform individual operations. You can use this option instead of creating a NITRO session (using the login object) and then using that session to perform all operations,

  To do this, you must specify the username and password in the request header of the NITRO request as follows:

  X-NITRO-USER:<username>

  X-NITRO-PASS:<password>

  Note: In such cases, make sure that the request header DOES not include the following:

  Cookie:NITRO_AUTH_TOKEN=<tokenvalue>

`*Note:`*

Mandatory parameters are marked in redand placeholder content is marked in <green>.

get (all)

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/stat/systemcpu

Query-parameters:

args

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/stat/systemcpu?
args=id:<Double_value>,detail:<Boolean_value>,fullvalues:<Boolean_value>,ntimes:<Double_value>,lo

Use this query-parameter to get systemcpu resources based on additional properties.

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "systemcpu": [ {

"id":<Double_value>,

"percpuuse":<Double_value>

}]}
```

get

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/stat/systemcpu/id_value><Double>

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "systemcpu": [ {

"id":<Double_value>,

"percpuuse":<Double_value>

}]}
```

systemmemory

Statistics for Global system memory stats resource.

Properties

(click to see Operations)

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
clearstats	<String>	Read-write	Clear the statsistics / counters. Possible values = basic, full
cacmemmaxmemlimitpcnt	<Double>	Read-only	Integrated Cache memory insue percent.
cacmemmaxmemlimit	<Double>	Read-only	Integrated Cache memory insue, in megabytes.
shmemerrallocfailed	<Double>	Read-only	Total shared memory allocation failed.
shmemallocpcnt	<Double>	Read-only	Shared memory insue percent.
shmemallocinmb	<Double>	Read-only	Shared memory insue, in megabytes.

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
shmemtotinmb	<Double>	Read-only	Total shared memory allowed to allocate, in megabytes.
memtotallocfailed	<Double>	Read-only	Total system memory allocation failed.
memtotfree	<Double>	Read-only	Total Free PE Memory in the System.
memusagepcnt	<Double>	Read-only	Percentage of memory utilization on Citrix ADC.
memtotuseinmb	<Double>	Read-only	Total Citrix ADC Memory in use, in megabytes.
memtotallocpcnt	<Double>	Read-only	Currently allocated memory in percent.
memtotallocmb	<Double>	Read-only	Currently allocated memory, in megabytes.
memtotinmb	<Double>	Read-only	Total memory available (grabbed) for use by packet engine (PE), in megabytes.
memtotavail	<Double>	Read-only	Total system memory available for PE to grab from the system.
cacmemmaxsyslimitmb	<Double>	Read-only	Integrated Cache memory, in megabytes.

Operations

(click to see Properties)

GET (ALL)

Some options that you can use for each operations:

- **Getting warnings in response:**NITRO allows you to get warnings in an operation by specifying the "warning" query parameter as "yes". For example, to get warnings while connecting to the Citrix ADC appliance, the URL is as follows:

  http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/login?warning=yes

  If any, the warnings are displayed in the response payload with the HTTP code "209 X-NITRO-WARNING".

- **Authenticated access for individual NITRO operations:**NITRO allows you to logon to the Citrix ADC appliance to perform individual operations. You can use this option instead of creating a NITRO session (using the login object) and then using that session to perform all operations,

  To do this, you must specify the username and password in the request header of the NITRO request as follows:

  X-NITRO-USER:<username>

  X-NITRO-PASS:<password>

  Note: In such cases, make sure that the request header DOES not include the following:

  Cookie:NITRO_AUTH_TOKEN=<tokenvalue>

`*Note:`*

Mandatory parameters are marked in redand placeholder content is marked in <green>.

get (all)

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/stat/systemmemory

Query-parameters:

args

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/stat/systemmemory?
args=detail:<Boolean_value>,fullvalues:<Boolean_value>,ntimes:<Double_value>,logfile:<String_value

Use this query-parameter to get systemmemory resources based on additional properties.

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "systemmemory": [ {

"memtotallocpcnt":<Double_value>,

"memtotfree":<Double_value>,

"shmemallocpcnt":<Double_value>,

"cacmemmaxmemlimitpcnt":<Double_value>,

"shmemallocinmb":<Double_value>,

"memtotuseinmb":<Double_value>,

"cacmemmaxmemlimit":<Double_value>,

"memusagepcnt":<Double_value>,

"memtotinmb":<Double_value>,

"cacmemmaxsyslimitmb":<Double_value>,
```

```
"memtotallocfailed":<Double_value>,

"memtotallocmb":<Double_value>,

"shmemerrallocfailed":<Double_value>,

"memtotavail":<Double_value>,

"shmemtotinmb":<Double_value>

}]}
```

Traffic Management

TM session/policy statistics.

NAME	DESCRIPTION
tmsessionpolicy	Statistics for TM session policy resource.
tmtrafficpolicy	Statistics for TM traffic policy resource.

tmsessionpolicy

Statistics for TM session policy resource.

Properties

(click to see Operations)

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
name	<String>	Read-write	Name of the advanced tmsession policy for which to display statistics. If no name is specified, statistics for all advanced tmsession polices are shown.

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
			Minimum length = 1
clearstats	<String>	Read-write	Clear the statsistics / counters. Possible values = basic, full
pipolicyhits	<Double>	Read-only	Number of hits on the policy
pipolicyhitsrate	<Double>	Read-only	Rate (/s) counter for pipolicyhits
pipolicyundefhits	<Double>	Read-only	Number of undef hits on the policy
pipolicyundefhitsrate	<Double>	Read-only	Rate (/s) counter for pipolicyundefhits

Operations

(click to see Properties)

GET (ALL)| GET

Some options that you can use for each operations:

- **Getting warnings in response:**NITRO allows you to get warnings in an operation by specifying the "warning" query parameter as "yes". For example, to get warnings while connecting to the Citrix ADC appliance, the URL is as follows:

  http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/login?warning=yes

  If any, the warnings are displayed in the response payload with the HTTP code "209 X-NITRO-WARNING".
- **Authenticated access for individual NITRO operations:**NITRO allows you to logon to the Citrix ADC appliance to perform individual operations. You can use this option instead of creating a NITRO session (using the login object) and then using that session to perform all operations,

To do this, you must specify the username and password in the request header of the NITRO request as follows:

X-NITRO-USER:<username>

X-NITRO-PASS:<password>

Note: In such cases, make sure that the request header DOES not include the following:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue>

`*Note:`*

Mandatory parameters are marked in redand placeholder content is marked in <green>.

get (all)

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/stat/tmsessionpolicy

Query-parameters:

args

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/stat/tmsessionpolicy?
args=name:<String_value>,detail:<Boolean_value>,fullvalues:<Boolean_value>,ntimes:<Double_value>

Use this query-parameter to get tmsessionpolicy resources based on additional properties.

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "tmsessionpolicy": [ {

"name":<String_value>,

"pipolicyundefhitsrate":<Double_value>,

"pipolicyundefhits":<Double_value>,

"pipolicyhitsrate":<Double_value>,

"pipolicyhits":<Double_value>

}]}
```

get

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/stat/tmsessionpolicy/name_value><String>

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "tmsessionpolicy": [ {

"name":<String_value>,

"pipolicyundefhitsrate":<Double_value>,
```

```
"pipolicyundefhits":<Double_value>,

"pipolicyhitsrate":<Double_value>,

"pipolicyhits":<Double_value>

}]}
```

tmtrafficpolicy

Statistics for TM traffic policy resource.

Properties

(click to see Operations)

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
name	<String>	Read-write	The name of the TM traffic policy for which statistics will be displayed. If not given statistics are shown for all policies. Minimum length = 1
clearstats	<String>	Read-write	Clear the statsistics / counters. Possible values = basic, full
pipolicyhits	<Double>	Read-only	Number of hits on the policy
pipolicyhitsrate	<Double>	Read-only	Rate (/s) counter for pipolicyhits

Operations

(click to see Properties)

GET (ALL)| GET

Some options that you can use for each operations:

- **Getting warnings in response:**NITRO allows you to get warnings in an operation by specifying the "warning" query parameter as "yes". For example, to get warnings while connecting to the Citrix ADC appliance, the URL is as follows:

  http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/login?warning=yes

  If any, the warnings are displayed in the response payload with the HTTP code "209 X-NITRO-WARNING".

- **Authenticated access for individual NITRO operations:**NITRO allows you to logon to the Citrix ADC appliance to perform individual operations. You can use this option instead of creating a NITRO session (using the login object) and then using that session to perform all operations,

  To do this, you must specify the username and password in the request header of the NITRO request as follows:

  X-NITRO-USER:<username>

  X-NITRO-PASS:<password>

  Note: In such cases, make sure that the request header DOES not include the following:

  Cookie:NITRO_AUTH_TOKEN=<tokenvalue>

`*Note:`*

Mandatory parameters are marked in redand placeholder content is marked in <green>.

get (all)

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/stat/tmtrafficpolicy

Query-parameters:

args

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/stat/tmtrafficpolicy?**args=name:<String_value>,detail:<Boolean_value>,fullvalues:<Boolean_value>,ntimes:<Double_value>**

Use this query-parameter to get tmtrafficpolicy resources based on additional properties.

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "tmtrafficpolicy": [ {

"name":<String_value>,

"pipolicyhitsrate":<Double_value>,

"pipolicyhits":<Double_value>

}]}
```

get

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/stat/tmtrafficpolicy/name_value><String>

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "tmtrafficpolicy": [ {

"name":<String_value>,

"pipolicyhitsrate":<Double_value>,

"pipolicyhits":<Double_value>

}]}
```

Transform

URL Transformation statistics.

NAME	DESCRIPTION
transformpolicy	Statistics for URL Transformation policy resource.
transformpolicylabel	Statistics for transform policy label resource.

transformpolicy

Statistics for URL Transformation policy resource.

Properties

(click to see Operations)

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
name	<String>	Read-write	Name of the policy. Minimum length = 1

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
clearstats	<String>	Read-write	Clear the statsistics / counters. Possible values = basic, full
pipolicyhits	<Double>	Read-only	Number of hits on the policy
pipolicyhitsrate	<Double>	Read-only	Rate (/s) counter for pipolicyhits
pipolicyundefhits	<Double>	Read-only	Number of undef hits on the policy
pipolicyundefhitsrate	<Double>	Read-only	Rate (/s) counter for pipolicyundefhits

Operations

(click to see Properties)

GET (ALL)| GET

Some options that you can use for each operations:

- **Getting warnings in response:**NITRO allows you to get warnings in an operation by specifying the "warning" query parameter as "yes". For example, to get warnings while connecting to the Citrix ADC appliance, the URL is as follows:

  http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/login?warning=yes

  If any, the warnings are displayed in the response payload with the HTTP code "209 X-NITRO-WARNING".

- **Authenticated access for individual NITRO operations:**NITRO allows you to logon to the Citrix ADC appliance to perform individual operations. You can use this option instead of creating a NITRO session (using the login object) and then using that session to perform all operations,

  To do this, you must specify the username and password in the request header of the NITRO request as follows:

X-NITRO-USER:<username>

X-NITRO-PASS:<password>

Note: In such cases, make sure that the request header DOES not include the following:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue>

`*Note:`*

Mandatory parameters are marked in redand placeholder content is marked in <green>.

get (all)

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/stat/transformpolicy

Query-parameters:

args

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/stat/transformpolicy?
args=name:<String_value>,detail:<Boolean_value>,fullvalues:<Boolean_value>,ntimes:<Double_value>

Use this query-parameter to get transformpolicy resources based on additional properties.

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "transformpolicy": [ {

"name":<String_value>,
```

```
"pipolicyundefhitsrate":<Double_value>,

"pipolicyundefhits":<Double_value>,

"pipolicyhitsrate":<Double_value>,

"pipolicyhits":<Double_value>

}]}
```

get

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/stat/transformpolicy/name_value><String>

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "transformpolicy": [ {

"name":<String_value>,

"pipolicyundefhitsrate":<Double_value>,

"pipolicyundefhits":<Double_value>,

"pipolicyhitsrate":<Double_value>,
```

```
"pipolicyhits":<Double_value>

}]}
```

transformpolicylabel

Statistics for transform policy label resource.

Properties

(click to see Operations)

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
labelname	<String>	Read-write	The name of the URL Transformation policy label. Minimum length = 1
clearstats	<String>	Read-write	Clear the statsistics / counters. Possible values = basic, full
pipolicylabelhits	<Double>	Read-only	Number of times policy label was invoked.
pipolicylabelhitsrate	<Double>	Read-only	Rate (/s) counter for pipolicylabelhits

Operations

(click to see Properties)

GET (ALL)| GET

Some options that you can use for each operations:

- **Getting warnings in response:**NITRO allows you to get warnings in an operation by specifying the "warning" query parameter as "yes". For example, to get warnings while connecting to the Citrix ADC appliance, the URL is as follows:

  http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/login?warning=yes

  If any, the warnings are displayed in the response payload with the HTTP code "209 X-NITRO-WARNING".

- **Authenticated access for individual NITRO operations:**NITRO allows you to logon to the Citrix ADC appliance to perform individual operations. You can use this option instead of creating a NITRO session (using the login object) and then using that session to perform all operations,

  To do this, you must specify the username and password in the request header of the NITRO request as follows:

  X-NITRO-USER:<username>

  X-NITRO-PASS:<password>

  Note: In such cases, make sure that the request header DOES not include the following:

  Cookie:NITRO_AUTH_TOKEN=<tokenvalue>

`*Note:`*

Mandatory parameters are marked in redand placeholder content is marked in <green>.

get (all)

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/stat/transformpolicylabel

Query-parameters:

args

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/stat/transformpolicylabel?
args=labelname:<String_value>,detail:<Boolean_value>,fullvalues:<Boolean_value>,ntimes:<Double_va

Use this query-parameter to get transformpolicylabel resources based on additional properties.

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "transformpolicylabel": [ {

"labelname":<String_value>,

"pipolicylabelhitsrate":<Double_value>,

"pipolicylabelhits":<Double_value>

}]}
```

get

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/stat/transformpolicylabel/labelname_value><String>

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "transformpolicylabel": [ {
"labelname":<String_value>,
"pipolicylabelhitsrate":<Double_value>,
"pipolicylabelhits":<Double_value>
}]}
```

Tunnel

SSL VPN Tunnel statistics.

NAME	DESCRIPTION
tunneltrafficpolicy	Statistics for tunnel policy resource.

tunneltrafficpolicy

Statistics for tunnel policy resource.

Properties

(click to see Operations)

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
name	<String>	Read-write	Name of the advanced tunnel traffic policy. Minimum length = 1
clearstats	<String>	Read-write	Clear the statsistics / counters.

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
			Possible values = basic, full
pipolicyhits	<Double>	Read-only	Number of hits on the policy
pipolicyhitsrate	<Double>	Read-only	Rate (/s) counter for pipolicyhits
pipolicyundefhits	<Double>	Read-only	Number of undef hits on the policy
pipolicyundefhitsrate	<Double>	Read-only	Rate (/s) counter for pipolicyundefhits

Operations

(click to see Properties)

GET (ALL)| GET

Some options that you can use for each operations:

- **Getting warnings in response:**NITRO allows you to get warnings in an operation by specifying the "warning" query parameter as "yes". For example, to get warnings while connecting to the Citrix ADC appliance, the URL is as follows:

  http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/login?warning=yes

  If any, the warnings are displayed in the response payload with the HTTP code "209 X-NITRO-WARNING".

- **Authenticated access for individual NITRO operations:**NITRO allows you to logon to the Citrix ADC appliance to perform individual operations. You can use this option instead of creating a NITRO session (using the login object) and then using that session to perform all operations,

  To do this, you must specify the username and password in the request header of the NITRO request as follows:

  X-NITRO-USER:<username>

X-NITRO-PASS:<password>

Note: In such cases, make sure that the request header DOES not include the following:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue>

`*Note:`*

Mandatory parameters are marked in redand placeholder content is marked in <green>.

get (all)

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/stat/tunneltrafficpolicy

Query-parameters:

args

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/stat/tunneltrafficpolicy?
args=name:<String_value>,detail:<Boolean_value>,fullvalues:<Boolean_value>,ntimes:<Double_value>

Use this query-parameter to get tunneltrafficpolicy resources based on additional properties.

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "tunneltrafficpolicy": [ {

"name":<String_value>,

"pipolicyhitsrate":<Double_value>,
```

```
"pipolicyundefhitsrate":<Double_value>,

"pipolicyhits":<Double_value>,

"pipolicyundefhits":<Double_value>

}]}
```

get

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/stat/tunneltrafficpolicy/name_value><String>

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "tunneltrafficpolicy": [ {

"name":<String_value>,

"pipolicyhitsrate":<Double_value>,

"pipolicyundefhitsrate":<Double_value>,

"pipolicyhits":<Double_value>,

"pipolicyundefhits":<Double_value>
```

```
}]}
```

User

User protocol statistics.

NAME	DESCRIPTION
uservserver	Statistics for virtual server resource.

uservserver

Statistics for virtual server resource.

Properties

(click to see Operations)

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
name	<String>	Read-write	Name of the user defined virtual server. If no name is provided, statistical data of all configured user defined virtual servers is displayed. Minimum length = 1
clearstats	<String>	Read-write	Clear the statsistics / counters. Possible values = basic, full
sortby	<String>	Read-write	use this argument to sort by specific key. Possible values =
sortorder	<String>	Read-write	use this argument to specify sort order.

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
			Default value: SORT_DESCENDING Possible values = ascending, descending
establishedconn	<Double>	Read-only	Number of client connections in ESTABLISHED state.
inactsvcs	<Double>	Read-only	number of INACTIVE services bound to a vserver
vslbhealth	<Double>	Read-only	Health of the vserver. This gives percentage of UP services bound to this vserver.
primaryipaddress	<String>	Read-only	IP address of the vserver
primaryport	<Integer>	Read-only	The port on which the service is running.
protocolname	<String>	Read-only	Protocol associated with the vserver
state	<String>	Read-only	Current state of the server. There are seven possible values: UP(7), DOWN(1), UNKNOWN(2), BUSY(3), OFS(Out of Service)(4),

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
			TROFS(Transition Out of Service)(5), TROFS_DOWN(Down When going Out of Service)(8)
actsvcs	<Double>	Read-only	number of ACTIVE services bound to a vserver
tothits	<Double>	Read-only	Total vserver hits
hitsrate	<Double>	Read-only	Rate (/s) counter for tothits
totalrequests	<Double>	Read-only	Total number of requests received on this service or virtual server. (This applies to HTTP/SSL services and servers.)
requestsrate	<Double>	Read-only	Rate (/s) counter for totalrequests
totalresponses	<Double>	Read-only	Number of responses received on this service or virtual server. (This applies to HTTP/SSL services and servers.)
responsesrate	<Double>	Read-only	Rate (/s) counter for totalresponses
totalrequestbytes	<Double>	Read-only	Total number of request bytes received on this

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
			service or virtual server.
requestbytesrate	<Double>	Read-only	Rate (/s) counter for totalrequestbytes
totalresponsebytes	<Double>	Read-only	Number of response bytes received by this service or virtual server.
responsebytesrate	<Double>	Read-only	Rate (/s) counter for totalresponsebytes
totalpktsrecvd	<Double>	Read-only	Total number of packets received by this service or virtual server.
pktsrecvdrate	<Double>	Read-only	Rate (/s) counter for totalpktsrecvd
totalpktssent	<Double>	Read-only	Total number of packets sent.
pktssentrate	<Double>	Read-only	Rate (/s) counter for totalpktssent
curclntconnections	<Double>	Read-only	Number of current client connections.
cursrvrconnections	<Double>	Read-only	Number of current connections to the actual servers behind the virtual server.
invalidrequestrespon se	<Double>	Read-only	Number invalid requests/responses on this vserver

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
invalidrequestrespon sedropped	<Double>	Read-only	Number invalid requests/responses dropped on this vserver

Operations

(click to see Properties)

GET (ALL)| GET

Some options that you can use for each operations:

- **Getting warnings in response:**NITRO allows you to get warnings in an operation by specifying the "warning" query parameter as "yes". For example, to get warnings while connecting to the Citrix ADC appliance, the URL is as follows:

  http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/login?warning=yes

  If any, the warnings are displayed in the response payload with the HTTP code "209 X-NITRO-WARNING".
- **Authenticated access for individual NITRO operations:**NITRO allows you to logon to the Citrix ADC appliance to perform individual operations. You can use this option instead of creating a NITRO session (using the login object) and then using that session to perform all operations,

  To do this, you must specify the username and password in the request header of the NITRO request as follows:

  X-NITRO-USER:<username>

  X-NITRO-PASS:<password>

  Note: In such cases, make sure that the request header DOES not include the following:

  Cookie:NITRO_AUTH_TOKEN=<tokenvalue>

`*Note:`*

Mandatory parameters are marked in redand placeholder content is marked in <green>.

get (all)

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/stat/uservserver

Query-parameters:

args

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/stat/uservserver?
args=name:<String_value>,detail:<Boolean_value>,fullvalues:<Boolean_value>,ntimes:<Double_value>

Use this query-parameter to get uservserver resources based on additional properties.

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "uservserver": [ {

"name":<String_value>,

"curclntconnections":<Double_value>,

"establishedconn":<Double_value>,

"totalpktssent":<Double_value>,

"tothits":<Double_value>,

"totalrequests":<Double_value>,
```

```
"responsebytesrate":<Double_value>,

"invalidrequestresponsedropped":<Double_value>,

"totalresponses":<Double_value>,

"requestbytesrate":<Double_value>,

"hitsrate":<Double_value>,

"cursrvrconnections":<Double_value>,

"pktsrecvdrate":<Double_value>,

"primaryipaddress":<String_value>,

"responsesrate":<Double_value>,

"totalrequestbytes":<Double_value>,

"invalidrequestresponse":<Double_value>,

"protocolname":<String_value>,

"state":<String_value>,

"vslbhealth":<Double_value>,

"actsvcs":<Double_value>,

"totalpktsrecvd":<Double_value>,

"pktssentrate":<Double_value>,

"totalresponsebytes":<Double_value>,

"primaryport":<Integer_value>,

"requestsrate":<Double_value>,

"inactsvcs":<Double_value>

}]}
```

get

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/stat/uservserver/name_value><String>

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "uservserver": [ {

"name":<String_value>,

"curclntconnections":<Double_value>,

"establishedconn":<Double_value>,

"totalpktssent":<Double_value>,

"tothits":<Double_value>,

"totalrequests":<Double_value>,

"responsebytesrate":<Double_value>,

"invalidrequestresponsedropped":<Double_value>,

"totalresponses":<Double_value>,
```

```
"requestbytesrate":<Double_value>,

"hitsrate":<Double_value>,

"cursrvrconnections":<Double_value>,

"pktsrecvdrate":<Double_value>,

"primaryipaddress":<String_value>,

"responsesrate":<Double_value>,

"totalrequestbytes":<Double_value>,

"invalidrequestresponse":<Double_value>,

"protocolname":<String_value>,

"state":<String_value>,

"vslbhealth":<Double_value>,

"actsvcs":<Double_value>,

"totalpktsrecvd":<Double_value>,

"pktssentrate":<Double_value>,

"totalresponsebytes":<Double_value>,

"primaryport":<Integer_value>,

"requestsrate":<Double_value>,

"inactsvcs":<Double_value>

}]}
```

videooptimization

Statistics for videooptimization.

Properties

(click to see Operations)

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
clearstats	<String>	Read-write	Clear the statsistics / counters. Possible values = basic, full
voctpdvideo	<Double>	Read-only	Total number of progressive download HTTP videos
voctpdvideorate	<Double>	Read-only	Rate (/s) counter for voctpdvideo
voctabvideo	<Double>	Read-only	Total number of adaptive bitrate HTTP videos
voctabvideorate	<Double>	Read-only	Rate (/s) counter for voctabvideo
voeabvideo	<Double>	Read-only	Total number of adaptive bitrate HTTPS videos
voeabvideorate	<Double>	Read-only	Rate (/s) counter for voeabvideo
voquicvideo	<Double>	Read-only	Total number of QUIC videos
voquicvideorate	<Double>	Read-only	Rate (/s) counter for voquicvideo
vovideooptother	<Double>	Read-only	Non video traffic
vovideoopherrate	<Double>	Read-only	Rate (/s) counter for vovideooptother

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
voctabvideoses	<Double>	Read-only	Total number of adaptive bitrate HTTP video sessions
voctabvideosesrate	<Double>	Read-only	Rate (/s) counter for voctabvideoses
voeabvideoses	<Double>	Read-only	Total number of adaptive bitrate HTTPS videos
voeabvideosesrate	<Double>	Read-only	Rate (/s) counter for voeabvideoses
voquicvideoses	<Double>	Read-only	Total number of sessions with QUIC videos
voquicvideosesrate	<Double>	Read-only	Rate (/s) counter for voquicvideoses
voctpdvideobytes	<Double>	Read-only	Total number of bytes for progressive download HTTP videos
voctpdvideobytesrate	<Double>	Read-only	Rate (/s) counter for voctpdvideobytes
voctabvideobytes	<Double>	Read-only	Total number of bytes for adaptive bitrate HTTP videos
voctabvideobytesrate	<Double>	Read-only	Rate (/s) counter for voctabvideobytes
voeabvideobytes	<Double>	Read-only	Total number of bytes for adaptive bitrate HTTPS videos

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
voeabvideobytesrate	<Double>	Read-only	Rate (/s) counter for voeabvideobytes
voquicvideobytes	<Double>	Read-only	Total number of bytes for QUIC videos
voquicvideobytesrate	<Double>	Read-only	Rate (/s) counter for voquicvideobytes
vovideooptotherbytes	<Double>	Read-only	Total number of bytes for non video traffic
vovideoopherbytesrate	<Double>	Read-only	Rate (/s) counter for vovideooptotherbytes

Operations

(click to see Properties)

GET (ALL)

Some options that you can use for each operations:

- **Getting warnings in response:**NITRO allows you to get warnings in an operation by specifying the "warning" query parameter as "yes". For example, to get warnings while connecting to the Citrix ADC appliance, the URL is as follows:

  http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/login?warning=yes

  If any, the warnings are displayed in the response payload with the HTTP code "209 X-NITRO-WARNING".
- **Authenticated access for individual NITRO operations:**NITRO allows you to logon to the Citrix ADC appliance to perform individual operations. You can use this option instead of creating a NITRO session (using the login object) and then using that session to perform all operations,

  To do this, you must specify the username and password in the request header of the NITRO request as follows:

X-NITRO-USER:<username>

X-NITRO-PASS:<password>

Note: In such cases, make sure that the request header DOES not include the following:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue>

`*Note:` *

Mandatory parameters are marked in redand placeholder content is marked in <green>.

get (all)

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/stat/videooptimization

Query-parameters:

args

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/stat/videooptimization?
args=detail:<Boolean_value>,fullvalues:<Boolean_value>,ntimes:<Double_value>,logfile:<String_value

Use this query-parameter to get videooptimization resources based on additional properties.

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "videooptimization": [ {

"voctpdvideobytesrate":<Double_value>,
```

```
"voquicvideoses":<Double_value>,

"voctabvideobytesrate":<Double_value>,

"vovideooptother":<Double_value>,

"voctpdvideobytes":<Double_value>,

"voctabvideorate":<Double_value>,

"voquicvideobytes":<Double_value>,

"voctabvideobytes":<Double_value>,

"voeabvideosesrate":<Double_value>,

"voctabvideoses":<Double_value>,

"voeabvideoses":<Double_value>,

"vovideoopherbytesrate":<Double_value>,

"voquicvideo":<Double_value>,

"voeabvideorate":<Double_value>,

"voctpdvideo":<Double_value>,

"voeabvideo":<Double_value>,

"voquicvideosesrate":<Double_value>,

"voeabvideobytesrate":<Double_value>,

"vovideoopherrate":<Double_value>,

"voctabvideo":<Double_value>,

"voquicvideobytesrate":<Double_value>,

"voctpdvideorate":<Double_value>,

"voctabvideosesrate":<Double_value>,
```

```
"voquicvideorate":<Double_value>,

"vovideooptotherbytes":<Double_value>,

"voeabvideobytes":<Double_value>

}]}
```

videooptimizationdetectionpolicy

Statistics for videooptimization detectionpolicy resource.

Properties

(click to see Operations)

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
name	<String>	Read-write	Name of the Video Optimization detection policy for which to show detailed statistics. Minimum length = 1
clearstats	<String>	Read-write	Clear the statsistics / counters. Possible values = basic, full
pipolicyhits	<Double>	Read-only	Number of hits on the policy
pipolicyhitsrate	<Double>	Read-only	Rate (/s) counter for pipolicyhits
pipolicyundefhits	<Double>	Read-only	Number of undef hits on the policy
pipolicyundefhitsrate	<Double>	Read-only	Rate (/s) counter for pipolicyundefhits

Operations

(click to see Properties)

GET (ALL)| GET

Some options that you can use for each operations:

- **Getting warnings in response:**NITRO allows you to get warnings in an operation by specifying the "warning" query parameter as "yes". For example, to get warnings while connecting to the Citrix ADC appliance, the URL is as follows:

  http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/login?warning=yes

  If any, the warnings are displayed in the response payload with the HTTP code "209 X-NITRO-WARNING".

- **Authenticated access for individual NITRO operations:**NITRO allows you to logon to the Citrix ADC appliance to perform individual operations. You can use this option instead of creating a NITRO session (using the login object) and then using that session to perform all operations,

  To do this, you must specify the username and password in the request header of the NITRO request as follows:

  X-NITRO-USER:<username>

  X-NITRO-PASS:<password>

  Note: In such cases, make sure that the request header DOES not include the following:

  Cookie:NITRO_AUTH_TOKEN=<tokenvalue>

`*Note:`*

Mandatory parameters are marked in redand placeholder content is marked in <green>.

get (all)

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/stat/videooptimizationdetectionpolicy

Query-parameters:

args

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/stat/videooptimizationdetectionpolicy?**args=name:<String_value>,detail:<Boolean_value>,fullvalues:<Boolean_value>,ntimes:<Double_value>**

Use this query-parameter to get videooptimizationdetectionpolicy resources based on additional properties.

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "videooptimizationdetectionpolicy": [ {

"name":<String_value>,

"pipolicyundefhitsrate":<Double_value>,

"pipolicyundefhits":<Double_value>,

"pipolicyhitsrate":<Double_value>,

"pipolicyhits":<Double_value>

}]}
```

get

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/stat/videooptimizationdetectionpolicy/name_value><String>

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "videooptimizationdetectionpolicy": [ {

"name":<String_value>,

"pipolicyundefhitsrate":<Double_value>,

"pipolicyundefhits":<Double_value>,

"pipolicyhitsrate":<Double_value>,

"pipolicyhits":<Double_value>

}]}
```

videooptimizationdetectionpolicylabel

Statistics for videooptimization detection policy label resource.

Properties

(click to see Operations)

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
labelname	<String>	Read-write	Name of the videooptimization detection policy label. Minimum length = 1
clearstats	<String>	Read-write	Clear the statsistics / counters. Possible values = basic, full
pipolicylabelhits	<Double>	Read-only	Number of times policy label was invoked.
pipolicylabelhitsrate	<Double>	Read-only	Rate (/s) counter for pipolicylabelhits

Operations

(click to see Properties)

GET (ALL)| GET

Some options that you can use for each operations:

- **Getting warnings in response:**NITRO allows you to get warnings in an operation by specifying the "warning" query parameter as "yes". For example, to get warnings while connecting to the Citrix ADC appliance, the URL is as follows:

  http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/login?warning=yes

  If any, the warnings are displayed in the response payload with the HTTP code "209 X-NITRO-WARNING".
- **Authenticated access for individual NITRO operations:**NITRO allows you to logon to the Citrix ADC appliance to perform individual operations. You can use this option instead of creating a NITRO session (using the login object) and then using that session to perform all operations,

To do this, you must specify the username and password in the request header of the NITRO request as follows:

X-NITRO-USER:<username>

X-NITRO-PASS:<password>

Note: In such cases, make sure that the request header DOES not include the following:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue>

`*Note:`*

Mandatory parameters are marked in redand placeholder content is marked in <green>.

get (all)

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/stat/videooptimizationdetectionpolicylabel

Query-parameters:

args

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/stat/videooptimizationdetectionpolicylabel?
args=labelname:<String_value>,detail:<Boolean_value>,fullvalues:<Boolean_value>,ntimes:<Double_va

Use this query-parameter to get videooptimizationdetectionpolicylabel resources based on additional properties.

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "videooptimizationdetectionpolicylabel": [ {

"labelname":<String_value>,

"pipolicylabelhitsrate":<Double_value>,

"pipolicylabelhits":<Double_value>

}]}
```

get

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/stat/videooptimizationdetectionpolicylabel/labelname_value><String>

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "videooptimizationdetectionpolicylabel": [ {

"labelname":<String_value>,

"pipolicylabelhitsrate":<Double_value>,

"pipolicylabelhits":<Double_value>
```

```
}]}
```

videooptimizationpacingpolicy

Statistics for videooptimization pacingpolicy resource.

Properties

(click to see Operations)

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
name	<String>	Read-write	Name of the Video Optimization Pacing policy for which to show detailed statistics. Minimum length = 1
clearstats	<String>	Read-write	Clear the statsistics / counters. Possible values = basic, full
pipolicyhits	<Double>	Read-only	Number of hits on the policy
pipolicyhitsrate	<Double>	Read-only	Rate (/s) counter for pipolicyhits
pipolicyundefhits	<Double>	Read-only	Number of undef hits on the policy
pipolicyundefhitsrate	<Double>	Read-only	Rate (/s) counter for pipolicyundefhits

Operations

(click to see Properties)

GET (ALL)| GET

Some options that you can use for each operations:

- **Getting warnings in response:**NITRO allows you to get warnings in an operation by specifying the "warning" query parameter as "yes". For example, to get warnings while connecting to the Citrix ADC appliance, the URL is as follows:

  http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/login?warning=yes

  If any, the warnings are displayed in the response payload with the HTTP code "209 X-NITRO-WARNING".

- **Authenticated access for individual NITRO operations:**NITRO allows you to logon to the Citrix ADC appliance to perform individual operations. You can use this option instead of creating a NITRO session (using the login object) and then using that session to perform all operations,

  To do this, you must specify the username and password in the request header of the NITRO request as follows:

  X-NITRO-USER:<username>

  X-NITRO-PASS:<password>

  Note: In such cases, make sure that the request header DOES not include the following:

  Cookie:NITRO_AUTH_TOKEN=<tokenvalue>

`*Note:`*

Mandatory parameters are marked in redand placeholder content is marked in <green>.

get (all)

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/stat/videooptimizationpacingpolicy

Query-parameters:

args

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/stat/videooptimizationpacingpolicy?**args=name:<String_value>,detail:<Boolean_value>,fullvalues:<Boolean_value>,ntimes:<Double_value>**

Use this query-parameter to get videooptimizationpacingpolicy resources based on additional properties.

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "videooptimizationpacingpolicy": [ {

"name":<String_value>,

"pipolicyundefhitsrate":<Double_value>,

"pipolicyundefhits":<Double_value>,

"pipolicyhitsrate":<Double_value>,

"pipolicyhits":<Double_value>

}]}
```

get

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/stat/videooptimizationpacingpolicy/name_value><String>

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "videooptimizationpacingpolicy": [ {

"name":<String_value>,

"pipolicyundefhitsrate":<Double_value>,

"pipolicyundefhits":<Double_value>,

"pipolicyhitsrate":<Double_value>,

"pipolicyhits":<Double_value>

}]}
```

videooptimizationpacingpolicylabel

Statistics for videooptimization pacing policy label resource.

Properties

(click to see Operations)

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
labelname	<String>	Read-write	Name of the videooptimization pacing policy label. Minimum length = 1
clearstats	<String>	Read-write	Clear the statsistics / counters.

NAME	DATA TYPE	PERMISSIONS	DESCRIPTION
			Possible values = basic, full
pipolicylabelhits	<Double>	Read-only	Number of times policy label was invoked.
pipolicylabelhitsrate	<Double>	Read-only	Rate (/s) counter for pipolicylabelhits

Operations

(click to see Properties)

GET (ALL)| GET

Some options that you can use for each operations:

- **Getting warnings in response:**NITRO allows you to get warnings in an operation by specifying the "warning" query parameter as "yes". For example, to get warnings while connecting to the Citrix ADC appliance, the URL is as follows:

  http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/config/login?warning=yes

  If any, the warnings are displayed in the response payload with the HTTP code "209 X-NITRO-WARNING".

- **Authenticated access for individual NITRO operations:**NITRO allows you to logon to the Citrix ADC appliance to perform individual operations. You can use this option instead of creating a NITRO session (using the login object) and then using that session to perform all operations,

  To do this, you must specify the username and password in the request header of the NITRO request as follows:

  X-NITRO-USER:<username>

  X-NITRO-PASS:<password>

  Note: In such cases, make sure that the request header DOES not include the following:

  Cookie:NITRO_AUTH_TOKEN=<tokenvalue>

`*Note:` *

Mandatory parameters are marked in redand placeholder content is marked in <green>.

get (all)

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/stat/videooptimizationpacingpolicylabel

Query-parameters:

args

http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/stat/videooptimizationpacingpolicylabel?
args=labelname:<String_value>,detail:<Boolean_value>,fullvalues:<Boolean_value>,ntimes:<Double_va

Use this query-parameter to get videooptimizationpacingpolicylabel resources based on additional properties.

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "videooptimizationpacingpolicylabel": [ {

"labelname":<String_value>,

"pipolicylabelhitsrate":<Double_value>,

"pipolicylabelhits":<Double_value>

}]}
```

get

URL:http://<Citrix-ADC-IP-address(NSIP)>/nitro/v1/stat/videooptimizationpacingpolicylabel/labelname_value><String>

HTTP Method:GET

Request Headers:

Cookie:NITRO_AUTH_TOKEN=<tokenvalue> Accept:application/json

Response:

HTTP Status Code on Success: 200 OK HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for Citrix ADC specific errors). The response payload provides details of the error**Response Headers:**

Content-Type:application/json

Response Payload:

```
{ "videooptimizationpacingpolicylabel": [ {

"labelname":<String_value>,

"pipolicylabelhitsrate":<Double_value>,

"pipolicylabelhits":<Double_value>

}]}
```

Error Messages

- 0x300 ERRORS
- 0x850 ERRORS
- 0x900 ERRORS
- ACL ERRORS
- ACL6 ERRORS
- ADTCP ERRORS
- APPFLOW ERRORS
- AS ERRORS
- BASE ERRORS
- CACONF ERRORS
- CENTRALMGMTSERVER ERRORS
- CFE ERRORS
- CLI ERRORS
- CLOUDPARAM ERRORS
- CLUSTER ERRORS
- CQA ERRORS
- CR ERRORS
- CVPN ERRORS
- DSK CACHE ERRORS
- EV SUBSCRIBER ERRORS
- FWDSESSION ERRORS
- GENTOOL ERRORS
- GSLB ERRORS
- GSLB2 ERRORS
- HTML INJECTION ERRORS

- ICA CLIENT VERSION ERRORS
- ICA LATENCY PROFILE ERRORS
- INAT ERRORS
- INTERNAL NAMESERVER ERRORS
- INTF ERRORS
- LB ERRORS
- LB2 ERRORS
- LSN ERRORS
- MAP DOMAIN ERRORS
- NAT64 ERRORS
- NET ERRORS
- NETBRIDGE ERRORS
- NETPROFILE ERRORS
- NITRO ERRORS
- NNM ERRORS
- PARTITION ERRORS
- PBR ERRORS
- PBR6 ERRORS
- PCP ERRORS
- PI ERRORS
- POL ERRORS
- POOLED LIC ERRORS
- PQ ERRORS
- QOS ERRORS
- REWRITE ERRORS
- RISE ERRORS
- RNAT ERRORS
- SNMP ERRORS
- SSL ERRORS
- SSL2 ERRORS

- SSLVPN AAA ERRORS
- SST ERRORS
- SST ERRORS2
- SVPN ERRORS
- TD ERRORS
- URLFILT ERRORS
- URLSET ERRORS
- URLTRANS ERRORS
- VIDEOOPT ERRORS
- VLAN ERRORS
- VMAC ERRORS
- VXLAN ERRORS
- WARNING ERRORS
- XS ERRORS

0x300 ERRORS

0x300 ERRORS

ERROR CODE	ERROR CODE(DECIMAL)	ERROR CODE(HEX)	ERROR MESSAGE
NSERR_RNAT_INV	769	0x301	Reverse NAT not applicable for default route.
NSERR_INVALID_IF	770	0x302	Invalid interface name/number.
NSERR_MGRLIMIT_REACHED	771	0x303	Maximum manager limit reached.
NSERR_SP_INVALD_TABLE	772	0x304	SP table entries should be in increasing order.

ERROR CODE	ERROR CODE(DECIMAL)	ERROR CODE(HEX)	ERROR MESSAGE
NSERR_RNAT_NATIP_EXISTS	773	0x305	RNAT to the target network with specified NAT IP already exists.
NSERR_RNAT_EXISTS	774	0x306	ACL or NETWORK based RNAT rule already exists.
NSERR_RNAT_NOT_EXISTS	775	0x307	ACL or NETWORK based RNAT rule does not exist.
NSERR_RNAT_NATIP_NOT_EXISTS	776	0x308	RNAT to the target network with specified NAT IP doesnt exist.
NSERR_RNAT_INVALID_NATIP	777	0x309	NAT IP is not valid.
NSERR_RNAT_XACLWITHTTL	778	0x30a	Acl with ttl can not be used in NAT64/ RNAT/RNAT6/ ForwardingSession/ LSN rule.
NSERR_ARP_DISABLED	784	0x310	IP has arp disabled.
NSERR_ARP_SEC_NOT_OWNEDIP	785	0x311	Secondary can not arp for this IP.
NSERR_CPE_RULE_INVAL	786	0x312	Invalid rule.
NSERR_INVAL_FLOWTYPE	787	0x313	Only authorization, audit, VPN session and traffic policies

ERROR CODE	ERROR CODE(DECIMAL)	ERROR CODE(HEX)	ERROR MESSAGE
			can be bound to aaa user or group.
NSERR_INVAL_POLICY_TYPE	788	0x314	Response rule is invalid in an authorization policy.
NSERR_CPE_RULE_ACTION_INVAL	789	0x315	Request action is valid only for request rule.
NSERR_CPE_DEF_SET_INVAL	790	0x316	Default policy cannot be set.
NSERR_INVAL_FORCECLEANUP	791	0x317	Invalid forcecleanup value.
NSERR_INVAL_AAA_GROUP	792	0x318	Invalid authorizationgroup value.
NSERR_INVAL_PROXY	793	0x319	Invalid allprotocolproxy value.
NSERR_INVAL_HTTTPPROXY	800	0x320	Invalid HTTP proxy value.
NSERR_INVAL_FTPPROXY	801	0x321	Invalid FTP proxy value.
NSERR_INVAL_SOCKPROXY	802	0x322	Invalid SOCKS proxy value.
NSERR_INVAL_GOPHERPROXY	803	0x323	Invalid GOPHER proxy value.
NSERR_INVAL_SSLPROXY	804	0x324	Invalid SSL proxy value.

ERROR CODE	ERROR CODE(DECIMAL)	ERROR CODE(HEX)	ERROR MESSAGE
NSERR_INVAL_AAAGRP_MAX	805	0x325	Max 5 groups can be specified in authorizationgroup.
NSERR_INVAL_MAX_PORT_NUM	806	0x326	Maximum 16 ports can be specified in httpport.
NSERR_INVAL_HTTPPORT	807	0x327	Invalid port.
NSERR_INVAL_VPNVSERER_POLTYPE	808	0x328	This type of policy cannot be bound to VPN Vserver.
NSERR_INVAL_VPNGLOBAL_POLTYPE	809	0x329	This type of policy cannot be bound to VPN Global.
NSERR_CPE_REM_INUSE	810	0x32a	Bound policy cannot be removed.
NSERR_PROXY_CONFLICT	811	0x32b	Proxy server for all protocols already configured.
NSERR_PROXY_INVAL	812	0x32c	Domain names allowed only if proxy type is browser
NSERR_PXYEXCPT_INVAL	813	0x32d	Proxy exception allowed only if proxy type is browser
NSERR_CPE_POLTYPE_NO_CSE	814	0x32e	Policy type does not support client security expressions in rule

ERROR CODE	ERROR CODE(DECIMAL)	ERROR CODE(HEX)	ERROR MESSAGE
NSERR_SESSACT_CSE_INCOMPATIBLE	815	0x32f	Session action and rule are incompatible
NSERR_NOMEM_CSE	816	0x330	Not enough memory while adding client security expressions
NSERR_INCOMPAT_FS_RULE	817	0x331	File system expressions supported in authorization policy only
NSERR_INCOMPAT_FS_MIX	818	0x332	Incompatible expressions mixed with file system expressions in rule
NSERR_DR_ENABLE	819	0x333	Dynamic routing can be enabled on only one IP per subnet
NSERR_MAX_DISTANCE	820	0x334	Only null interface routes can have distance equal to 255
NSERR_NULL_ROUTE_DISTANCE	821	0x335	It is not possible to set the administrative distance/cost metric for a null interface route
NSERR_BAD_ACTION_TCP_PROFILE_TYPE	822	0x336	TCP profile cannot be set to this service type

ERROR CODE	ERROR CODE(DECIMAL)	ERROR CODE(HEX)	ERROR MESSAGE
NSERR_BAD_ACTION_HTTP_PROFILE_TYPE	823	0x337	HTTP profile cannot be set to this service type
NSERR_SP_INVALID_THRESHOLD	824	0x338	Invalid base threshold value
NSERR_VIP_ROUTE_EXISTS	825	0x339	VIP exists for this host route
NSERR_ACTION_BOUND	832	0x340	Action bound to policy can not be deleted.
NSERR_MAX_IPLIMIT	833	0x341	Maximum limit for bound IP to this resource record reached, remaining IPs destined for this resource record will be discarded.
NSERR_BAD_PREFIX_LEN	826	0x33a	Invalid rnat/lsn network prefix len it must be 0-128
NSERR_INV_NETADDR	827	0x33b	Invalid IPv6 network address
NSERR_MIN_DISTANCE	828	0x33c	Static route cannot have distance less than 1
NSERR_IPSEC_PROFILE_INUSE	829	0x33d	Profile is in use by an ip tunnel.
NSERR_IPSEC_INVALID_PROFILE_NAME	830	0x33e	Invalid profile name.

ERROR CODE	ERROR CODE(DECIMAL)	ERROR CODE(HEX)	ERROR MESSAGE
NSERR_SPOTTED	831	0x33f	Spotted IP cannot be used here
NSERR_IPSEC_PBR_ONLY_IP_SUPPORTED	834	0x342	Only IP range is supported
NSERR_IPSEC_IP_NOT_CONTIGUOUS	835	0x343	Invalid IP range. The IP range does not cover all the IPs for a given subnet mask
NSERR_INVAL_USER_ACCOUNTING	836	0x344	Invalid userAccounting value. userAccounting value must be a radiusPolicy.
NSERR_RDP_PROFILE_INUSE	837	0x345	Profile is in use.
NSERR_RDP_INVALID_PROFILE_NAME	838	0x346	Invalid profile name.
NSERR_RDP_INVALID_RDP_FILE_NAME	839	0x347	RDP file name should have .rdp extension and some special characters for RDP file name are not supported.
NSERR_RDP_INVALID_SERVERPROFILE_NAME	840	0x348	Invalid server profile name.
NSERR_RDP_SERVERPROFILE_ALREADY_IN_USE	841	0x349	Server profile is already in use.

ERROR CODE	ERROR CODE(DECIMAL)	ERROR CODE(HEX)	ERROR MESSAGE
NSERR_AUTO_PROXY_URL_INVALID	842	0x34a	Invalid Autoproxy URL
NSERR_AUTO_PROXY_NO_BROWSER	843	0x34b	Autoproxy URL is allowed only if proxy type is Browser.
NSERR_AUTO_PROXY_CONFLICT	844	0x34c	Both Autoproxy URL and manual proxy parameters (allprotocolproxy, http proxy, ftp proxy, proxy exception list, ...) cannot be configured at same time.
NSERR_RDP_INVALID_HOST_LEN	848	0x350	Invalid Host length. Atleast 3 characters expected
NSERR_RDP_NEED_IP	849	0x351	RDP IP is needed
NSERR_IPSEC_INV_PROF	850	0x352	For IPSEC tunnels, profile name cannot be none
NSERR_RDP_NEED_PSK	854	0x356	PSK is needed
NSERR_IPSEC_ONLY_V1	858	0x35a	Responder Only config is supported with IKEv1
NSERR_RDP_LISTENER_SPECIFY_PORT	859	0x35b	Invalid Listener. Specify Port

ERROR CODE	ERROR CODE(DECIMAL)	ERROR CODE(HEX)	ERROR MESSAGE
NSERR_RDP_INVALID_LISTENER_HOST	860	0x35c	Invalid Listener. Specify valid host, atleast 3 characters expected
NSERR_RDP_LISTENER_INVALID_PORT	861	0x35d	Invalid Listener. Specify valid Port
NSERR_PROXY_NOT_ALLOWED	862	0x35e	Proxy feature not licensed
NSERR_HOST_RT_GW_NOT_SET	3806	0xede	Host Route Gateway not set

0x850 ERRORS

0x850 ERRORS

ERROR CODE	ERROR CODE(DECIMAL)	ERROR CODE(HEX)	ERROR MESSAGE
NSERR_MONITOR_INTERVAL	2128	0x850	Monitor interval must be greater than response timeout
NSERR_MONITOR_DESTIP	2129	0x851	Destination IP must be specified for transparent, tunneled or tos monitors
NSERR_MONITOR_CODES	2130	0x852	Too many response codes, only 16 allowed
NSERR_MONITOR_REF	2131	0x853	Monitor must be unbound before it can be deleted

ERROR CODE	ERROR CODE(DECIMAL)	ERROR CODE(HEX)	ERROR MESSAGE
NSERR_MONITOR_BUILTIN	2132	0x854	Built-in monitors cannot be deleted
NSERR_MONITOR_BOUND	2133	0x855	The monitor is already bound to the service
NSERR_MONITOR_TYPE	2134	0x856	Invalid monitor type
NSERR_MONITOR_LOCAL	2135	0x857	Monitor object cannot be bound to local service
NSERR_TIMEOUT_RANGE	2136	0x858	Timeout value out of range; enter a value between 2 minutes and 1440 minutes
NSERR_INVALIDHASHLEN	2137	0x859	Hashlength should be between 1 and 4096 (inclusive)
NSERR_NOTAUTHORIZED	2138	0x85A	Not authorized to execute this command
NSERR_MONITOR_DEFAULT	2139	0x85B	No operations allowed with the default monitor
NSERR_MONITOR_LDNS_ADD_PERM	2140	0x85C	Cannot add monitor of ldns type
NSERR_MONITOR_LDNS_BIND_PERM	2141	0x85D	Cannot bind ldns monitor
NSERR_BACKUP_LOOP	2144	0x860	The VIP is already a backup in the chain

ERROR CODE	ERROR CODE(DECIMAL)	ERROR CODE(HEX)	ERROR MESSAGE
NSERR_SECURE_UDP	2154	0x86A	Secure mode not supported for this protocol
NSERR_MONITOR_WRONG_TYPE	2155	0x86B	Existing monitor is of different type than given
NSERR_LRTM_PERM	2156	0x86C	Enabling LRTM on this monitor type is not permitted
NSERR_MONITOR_SCRIPTNAME	2157	0x86D	Invalid script name for user monitor
NSERR_MONITOR_DISPATCHERIP	2159	0x86F	Invalid dispatcher IP for user monitor
NSERR_MONITOR_USERPERM	2160	0x870	Operation not permitted for a user monitor
NSERR_MONITOR_NOCODES	2161	0x871	A response code is required for this monitor type
NSERR_INVAL_MON	2162	0x872	Attempt to bind invalid monitor type
NSERR_TYPE_EXISTS	2163	0x873	Only one monitor of this type can be bound to a service
NSERR_MONITOR_UNBIND_DEFAULT	2164	0x874	Default monitor cannot be unbound from a service
NSERR_MONITOR_BIND_DEFAULT	2165	0x875	Default monitor cannot be bound explicitly to a service

ERROR CODE	ERROR CODE(DECIMAL)	ERROR CODE(HEX)	ERROR MESSAGE
NSERR_MONITOR_DISABLE_DEFAULT	2166	0x876	Default monitor cannot be disabled or enabled
NSERR_DRTM_PERM	2167	0x877	Dynamic response timeout is not permitted on this monitor type
NSERR_MON_DTRM_DEVIATION	2168	0x878	Response timeout of the monitor has been changed to interval -1
NSERR_MONITOR_INVALID_VALUE	2169	0x879	Time parameter must be a multiple of 10
NSERR_MONITOR_NOT_BOUND	2170	0x87A	Monitor not bound to service
NSERR_INVALID_MONITOR	2171	0x87B	Monitor does not exist
NSERR_MONITOR_NO_SUCHIPADDR	2172	0x87C	Some IP addresses were not present
NSERR_MONITOR_SCRIPT_ARG_SIZE	2173	0x87D	Combined argument size too long
NSERR_METRICTABLE_NOENT	2174	0x87E	Metric table does not exist
NSERR_METRICTABLE_EXIST	2175	0x87F	Metric table exists
NSERR_DEL_METRICTABLE_PERMANENT	2176	0x880	Permanent metric table cannot be deleted

ERROR CODE	ERROR CODE(DECIMAL)	ERROR CODE(HEX)	ERROR MESSAGE
NSERR_MAX_METRIC_BINDING	2177	0x881	Maximum of 10 SNMP type metric can be bound to the monitor
NSERR_METRIC_NOENT	2178	0x882	Metric does not exist
NSERR_METRIC_EXISTS	2179	0x883	Metric exists
NSERR_OID_EXIST	2180	0x884	SNMP OID exists
NSERR_SNMP_OID_INVAL	2181	0x885	SNMP OID is invalid
NSERR_METRICTABLE_RDONLY	2182	0x886	No operations allowed on read only metric table
NSERR_THRESHOLD_ZERO	2183	0x887	This metric will not be used for CUSTOM LB as its threshold value is zero
NSERR_LDAP_MONITOR_INCOMPLETE	2184	0x888	bindDN or baseDN must be specified before LDAP monitor can be used
NSERR_MYSQL_MONITOR_INCOMPLETE	2185	0x889	userName or database must be specified before MYSQL monitor can be used
NSERR_POP3_MONITOR_INCOMPLETE	2186	0x88A	userName must be specified before

ERROR CODE	ERROR CODE(DECIMAL)	ERROR CODE(HEX)	ERROR MESSAGE
			POP3 monitor can be used
NSERR_NNTP_MONITOR_INCOMPLETE	2187	0x88B	group must be specified before NNTP monitor can be used
NSERR_FTPEXTENDED_MONITOR_INCOMPLETE	2188	0x88C	fileName must be specified before FTP-EXTENDED monitor can be used
NSERR_SNMP_MONITOR_INCOMPLETE	2189	0x88D	snmpOID or snmpCommunity must be specified before SNMP monitor can be used
NSERR_CITRIX_XML_SERVICE	2190	0x88E	application must be specified before CITRIX-XML-SERVICE monitor can be used
NSERR_CITRIX_WEB_INTERFACE	2191	0x88F	sitePath must be specified before CITRIX-WEB-INTERFACE monitor can be used
NSERR_LDNS_MON_CANT_DISABLE	2192	0x890	Atleast one ldns monitor needs to be enabled
NSERR_RES_RETRY_ON_IP_SVR	2193	0x891	Resolve retry can be set only on domain based servers

ERROR CODE	ERROR CODE(DECIMAL)	ERROR CODE(HEX)	ERROR MESSAGE
NSERR_IP_ON_DBS_SVR	2194	0x892	IP Address cannot be set on a domain based server
NSERR_MONITOR_FAILURERETRIES	2196	0x894	Monitor retries must be greater than monitor failureRetries
NSERR_MONITOR_IP	2198	0x896	The server/monitor destination ip address is not appropriate for this monitor
NSERR_MONITOR_SUBNET	2199	0x897	The server/monitor destination ip address should be in a directly connected subnet for this monitor
NSERR_MONITOR_STATIC	2200	0x898	A static/permanent MAC entry is configured for the server/monitor destination ip address
NSERR_TOSID_NOT_SET	2201	0x899	Tos id must be specified for tos enabled vserver/ monitor
NSERR_FAILURERETRIES_NOT_SUPPORTED	2202	0x89A	The failure retries is not supported for this monitor type. Resetting the value of failure retries to 0(Disabled).

ERROR CODE	ERROR CODE(DECIMAL)	ERROR CODE(HEX)	ERROR MESSAGE
NSERR_SUCCESS RETRIES_NOT_SU PPORTED	2203	0x89B	The success retries is not supported for this monitor type. Resetting the value of success entries to 1(Disabled).
NSERR_SETVS_IN VAL_MYSQLPARAM S	2204	0x89C	This parameter is only applicable to a MYSQL vserver.
NSERR_SETVS_IN VAL_PROTOCOL_P ARAMS	2205	0x89D	The parameters are only applicable to HTTP or SSL vservers.
NSERR_SETVS_IN VAL_PROTOCOLS_ PARAMS	2206	0x89E	The parameters are only applicable to HTTP , SSL or RTSP vservers.
NSERR_DDC_VALI DATE_CRED_REQU IRED	2207	0x89F	Username, password and ddc domain are required for Xen Desktop monitor\\n
NSERR_CITRIX_WI _EXTENDED_MONI TOR_INCOMPLETE	2208	0x8A0	userName, password, sitepath and domain must be specified before CITRIX-WI-EXTENDED monitor can be used
NSERR_SETVS_IN VAL_MSSQLPARAM S	2209	0x8A1	This parameter is only applicable to a MSSQL vserver.
NSERR_DRTM_MO N_BOUNDED_SVC	2210	0x8A2	In a cluster system, deviation cant be set/

ERROR CODE	ERROR CODE(DECIMAL)	ERROR CODE(HEX)	ERROR MESSAGE
			unset on any already bound monitor.
NSERR_MONITOR_RESP_TO_THRESH	2211	0X8A3	Response timeout threshold is rounded off to nearest multiple of 10 less than the given value
NSERR_PASSIVE_ON_LOAD_ONLY	2212	0x8A4	PASSIVE option can be set only on LOAD monitors.
NSERR_SETVS_INVAL_ORACLEPARAMS	2213	0x8A5	This parameter is only applicable to a ORACLE vserver.
NSERR_RADIUS_ACCOUNTING_MONITOR_INCOMPLETE	2214	0x8A6	Secret key must be specified before RADIUS ACCOUNTING monitor can be used
NSERR_MONITOR_SCRIPT_ARG_CNT	2215	0x8A7	Too many script arguments
NSERR_INVALID_DBSLB_BIND	2216	0x8A8	More than one bind not possible on one service for DBSLB
NSERR_USER_NO NETPROF	2217	0x8A9	Netprofile cannot be set on scriptable monitors, so ignoring the configured netprofile for this monitor.
NSERR_MONITOR_NO_SSLPROFILE_BOUND	2218	0x8AA	No SSL profile is attached to monitor. Attach SSL profile first

ERROR CODE	ERROR CODE(DECIMAL)	ERROR CODE(HEX)	ERROR MESSAGE
NSERR_DIAMETER_MONITOR_INCOMPLETE	3550	0xDDE	originHost and originRealm configuration is missing

0x900 ERRORS

0x900 ERRORS

ERROR CODE	ERROR CODE(DECIMAL)	ERROR CODE(HEX)	ERROR MESSAGE
NSERR_SVCPORTTYPE	2305	0x901	Service exists with the same port and service type
NSERR_VIP_IN_GROUP	2306	0x902	vserver is bound to a group. Set persistence parameters on group to change them for all vservers in group or unbind vserver from group and set on individual vserver.
NSERR_UNSUPPORTED_BKP	2307	0x903	This backup/primary persistence combination is not supported
NSERR_NONHTTPSSL_VIPINGRP	2308	0x904	Cookie persistence cannot be applied - group has non HTTP/SSL type of vserver

ERROR CODE	ERROR CODE(DECIMAL)	ERROR CODE(HEX)	ERROR MESSAGE
NSERR_NONHTTP_VIPINGRP	2309	0x905	Cookie persistence cannot be applied - group has non-HTTP type of vserver
NSERR_USE_PROPER_RM_CMD	2310	0x906	Use remove IP option instead
NSERR_LBMETHOD_NOT_SUPPORTED	2321	0x911	LB method not supported for LLB/PBR
NSERR_PERSISTENCE_NOT_SUPPORTED	2322	0x912	Persistence not supported for LLB/PBR
NSERR_VSERVER_PARAMETERS	2323	0x913	Vserver arguments not valid for LLB/PBR VIP
NSERR_SERVICE_PARAMETERS	2324	0x914	Service parameters are invalid for LLB/PBR VIP
NSERR_LBVIP_DELETE	2325	0x915	Vserver cannot be removed with out removing LB/PBR route
NSERR_CACHEPOLICY_RESPACTION_INVAL	2326	0x916	Integrated caching action cannot be applied on a response
NSERR_CACHELICENSE_FAILED	2327	0x917	The license for Integrated Caching feature was not enabled due to an internal error

ERROR CODE	ERROR CODE(DECIMAL)	ERROR CODE(HEX)	ERROR MESSAGE
NSERR_LBVIP_MULTIROUTES	2328	0x918	VIP cant be associated with multiple LB/PBR routes
NSERR_LBMAC_INVAL	2329	0x919	MAC/IPTUNNEL mode can be set only for a VIP with wildcard IP or with service type ANY or for a sessionless VIP
NSERR_CACHESTATS_OBJ_NOTPRESENT	2336	0x920	No object in cache matching the specified attributes
NSERR_CACHEPOLICY_NOTACTIVE	2337	0x921	Integrated caching policy is not active
NSERR_CACHE_BUILTINS_NOT_SOURCED	2338	0x922	Failed sourcing cache builtins, Disabling IC
NSERR_CACHEPOLICY_PRIORITY_INVAL	2339	0x923	Invalid priority
NSERR_ROUTE6_DEFAULT_ONLY	2340	0x924	Only default route configuration supported
NSERR_ROUTE6_DEFAULT_EXISTS	2341	0x925	Configured route already exists
NSERR_ROUTE6_INVALID_GATEWAY	2342	0x926	Invalid gateway
NSERR_ROUTE6_MAX	2343	0x927	Already maximum number of routes configured

ERROR CODE	ERROR CODE(DECIMAL)	ERROR CODE(HEX)	ERROR MESSAGE
NSERR_ROUTE6_NOT_EXIST	2344	0x928	Route does not exist
NSERR_IPV6_INVALID_ADDR	2345	0x929	Incorrect IPv6 address type
NSERR_LBVIP_IPV6TOV4	2352	0x930	Changing IP from IPv6 to IPv4 not permitted
NSERR_IPV6_NSIPTOVIP	2353	0x931	Configuring NSIP as VIP not permitted
NSERR_IPV6_VIPTONSIP	2354	0x932	Configuring VIP as NSIP not permitted
NSERR_IPV6_SCOPE	2355	0x933	Incorrect IPv6 address scope. (Default: global)
NSERR_LBVIP_IPV4TOV6	2356	0x934	Changing IP from IPv4 to IPv6 not permitted
NSERR_IPV6_LINKLOCALTOVIP	2357	0x935	Configuring Link-Local address as VIP not permitted
NSERR_IPV6_MAPIPONNSIP	2358	0x936	Mapped IP should not be configured for NSIP (Ignoring mapped ip)
NSERR_SECUREIPPORTADDRINUSE	2359	0x937	Internal secure service exists with the same port and service type. Please use this for secure access

ERROR CODE	ERROR CODE(DECIMAL)	ERROR CODE(HEX)	ERROR MESSAGE
NSERR_IPV6_INVALID_PREFIX	2360	0x938	Link-local prefix length is not equal to 64
NSERR_ND6_LINKLOCAL_VLAN	2361	0x939	Vlan is necessary with IPv6 Link-local address
NSERR_INVALID_IPV6_PREFIXLFT	2362	0x93A	Invalid IPv6 prefix life time. prefix valid life time must be greater than or equal to preferred life time
NSERR_IPV6_ND_TIMEOUT	2363	0x93B	Can not SET/UNSET ndtimeouts, when ra_learning is enabled.
NSERR_ND6_VLAN_INTF	2368	0x940	Interface/channel not a member of given vlan
NSERR_PERSIST_TIMEOUT_TO_DEFAULT	2369	0x941	Unsetting Persistency, changing timeout to default
NSERR_LB_GROUP_NOT_EXIST	2370	0x942	LB group does not exist
NSERR_LB_VSERVER_ALREADY_BOUND	2371	0x943	Vserver is already bound to a LB group
NSERR_ROUTE6_INVALID_LINKLOCAL_GATEWAY_VLAN	2372	0x944	Vlan ID must be specified for linklocal gateway

ERROR CODE	ERROR CODE(DECIMAL)	ERROR CODE(HEX)	ERROR MESSAGE
NSERR_ROUTE6_NO_GWORIF	2373	0x945	Either Gateway or Interface (vlan) should be specified for route
NSERR_ROUTE6_VLAN_MONITOR	2374	0x946	Monitor should not be given for vlan route
NSERR_LBMETHOD_LEASTREQ_NOT_SUPPORTED	2375	0x947	LeastRequest LB method not supported for this service type
NSERR_IPV6_MULTIPLE_LLIP	2376	0x948	Multiple link-local IPs to a Vlan is not allowed
NSERR_IPV6_TOOMANY_PREFIXES	2377	0x949	Too many prefixes to a Vlan, maxmimum 5 prefixes are allowed
NSERR_RAPREFIX_BOUNDTOVLAN	2378	0x94A	Router Advertisement prefix is bound to vlan
NSERR_LB_VSERVER_NOT_BOUND	2379	0x94B	Given vserver is not bound to this LB group
NSERR_VSERVER_BOUND_TO_NG	2380	0x94C	Cannot delete vserver that is bound to a nodegroup
NSERR_LRMINTHROUGHPUT_ONLACPONLY	2382	0x94E	Link Redundancy minimum throughput can be set only on LACP Channels

ERROR CODE	ERROR CODE(DECIMAL)	ERROR CODE(HEX)	ERROR MESSAGE
NSERR_CACHE_POLICY_CONTENTGRPMISMATCH	2383	0x94F	Type of Content Group and Policy does not Match.
NSERR_NONHTTPNOSQL_VS	2384	0x950	Specified policy can be bound only to HTTP/SSL/MYSQL/MSSQL/PROXY vserver.
NSERR_SMPP_CUSTOM_SVRID_LEN_NOT_SUPPORTED	2385	0x951	CustomServerID length for SMPP protocol must be 2
NSERR_SMPP_CUSTOM_SVRID_NOT_EXIST	2386	0x952	CustomServerID is mandatory for SMPP protocol
NSERR_SMPP_CUSTOM_SVRID_EXIST	2387	0x953	CustomServerID already exists, it should be unique for SMPP protocol
NSERR_SIP_DUPLICATE_SRC_PORT	2388	0x954	RNAT source port and RNAT secure source port must be different
NSERR_SIP_DUPLICATE_DST_PORT	2389	0x955	RNAT destination port and RNAT secure destination port must be different
NSERR_IP_ADDR_ICMP_DIS_NOT_ALLOWED	2390	0x956	Disabling ICMP for an IP(v6) address is not allowed
NSERR_IP_ADDR_ND6_DIS_NOT_ALLOWED	2391	0x957	disabling neighbor discovery for an ipv6

ERROR CODE	ERROR CODE(DECIMAL)	ERROR CODE(HEX)	ERROR MESSAGE
			address is not allowed
NSERR_IP_ADDR_STATE_DIS_NOT_ALLOWED	2392	0x958	state disable is not allowed for an IP(v6) address
NSERR_ND6_LOOP_LL_NOT_ALLOW	2393	0x959	Loopback/self or link-local IP addresses not allowed
NSERR_PERSISTENCE_NOT_SUPPORTED_ON_GROUP	2417	0x971	Vserver level persistency is enabled on the group. Set Persistence parameters on bound vservers.
NSERR_ARGS_AAA_OTP	2420	0x974	authentication can not be ENABLED when otpsecret is specified
NSERR_ARGS_AAA_SSHPUBKEY	2421	0x975	authentication can not be ENABLED when sshPubKey is specified
NSERR_CLOUD_CRED	2422	0x976	Unable to connect to the cloud. Check credentials.
NSERR_CLOUD_NSCRED	2423	0x977	Unable to issue NITRO API calls.
NSERR_SECTYPE_PLAINTEXT	2424	0x978	validateServerCert configuration not allowed when secType is PLAINTEXT.

ERROR CODE	ERROR CODE(DECIMAL)	ERROR CODE(HEX)	ERROR MESSAGE
NSERR_ARGS_AAA_HASH_ATTRIB	2425	0x979	Value to
NSERR_CLOUD_DNS	2432	0x980	Unable to connect to the cloud. DNS resolution failed.
NSERR_CLOUD_FSERR	2433	0x981	Unable to validate cloud credentials. Access error.

ACL ERRORS

ACL ERRORS

ERROR CODE	ERROR CODE(DECIMAL)	ERROR CODE(HEX)	ERROR MESSAGE
NSERR_ACL_NOT_EXISTS	864	0x360	ACL rule does not exist
NSERR_ACL_EXISTS	865	0x361	IP address has existing ACL rule
NSERR_ACLPIP_WOSRCDST	866	0x362	Peer IP cant be given without src/dst flag
NSERR_ACL_SAMEIP_PIP	867	0x363	IP address and peer IP cant be same
NSERR_ACL_INVAL_PEERIP	868	0x364	Invalid peer IP
NSERR_ACL_IPPIP_EXISTS	869	0x365	ACL with identical parameter specification already exists

ERROR CODE	ERROR CODE(DECIMAL)	ERROR CODE(HEX)	ERROR MESSAGE
NSERR_XACLDELERROR	870	0x366	ACL has already been removed
NSERR_XACLADDERROR	871	0x367	Src/DstPort or RedirectPort can be specified only if protocol is TCP (6) or UDP (17)
NSERR_XACL_PRIORITY_EXISTS	872	0x368	ACL with this priority already exists
NSERR_XACL_ICMP_REQD	873	0x369	ICMP type / code can be specified only if protocol is ICMP(1)
NSERR_NOLOOPBACK	874	0x36a	ACL cannot be configured on the loopback interface
NSERR_INVICMPTYPE	875	0x36b	Invalid ICMP type
NSERR_INVICMPCODE	876	0x36c	Invalid ICMP code
NSERR_XACLRNATDEL	877	0x36d	ACL is bounded to RNAT/LSN, cannot be removed
NSERR_XACLMODCFGINFO	878	0x36e	ACL modified, apply ACLs to activate change
NSERR_XACLFWDSESSIONDEL	879	0x36f	ACL is bounded to FORWARDSESSION, cannot be removed

ACL6 ERRORS

ACL6 ERRORS

ERROR CODE	ERROR CODE(DECIMAL)	ERROR CODE(HEX)	ERROR MESSAGE
NSERR_ACL6_IPPIP_EXISTS	896	0x380	ACL6 with identical parameter specification already exists
NSERR_ACL6_DELERROR	897	0x381	ACL6 has already been removed
NSERR_ACL6_ADDERROR	898	0x382	Port can be specified only if protocol is TCP (6) or UDP (17)
NSERR_ACL6_PRIORITY_EXISTS	899	0x383	ACL6 with this priority already exists
NSERR_ACL6_ICMP_REQD	900	0x384	ICMPv6 type/code can be specified only if protocol is ICMPv6 (58)
NSERR_ACL6_UNSPECADDR	901	0x385	unspecified address (::) can not be configured in ACL6
NSERR_ACL6_MODCFGINFO	902	0x386	ACL6 modified, apply ACLs6 to activate change
NSERR_ACL6_PREFIXLEN	903	0x387	Prefix length should not be configured in ACL6. (Use range instead)

ERROR CODE	ERROR CODE(DECIMAL)	ERROR CODE(HEX)	ERROR MESSAGE
NSERR_ACL6_RNAT_NOT_ALLOWED	904	0x388	Action must be ALLOW to bind ACL6 to RNAT
NSERR_ACL6RNAT_DELERROR	905	0x389	This ACL6 is associated with rnat6/nat64/ forwardingsession/ lsn config, Remove rnat6/nat64/ forwardingsession/ lsn first
NSERR_RNAT6_INVALID_REDPORT	906	0x38a	Invalid option for SET/UNSET, redirect port can be specified only for acl6 based RNAT
NSERR_ACL6_RNAT_EXISTS	907	0x38b	ACL6 based rnat is already exist for this acl6(Duplicate config)
NSERR_NET_RNAT6_EXISTS	908	0x38c	Network based rnat/ lsn already exists for this network address(Duplicate config)
NSERR_RNAT_VIP_PORT_ALLOC_FAIL	909	0x38d	ERROR : Port allocation for VIP v6 address Failed
NSERR_ACL6_RNAT_NAT64_NOT_ALLOWED	910	0x38e	ERROR : Acl6 already bound to nat64 config

ERROR CODE	ERROR CODE(DECIMAL)	ERROR CODE(HEX)	ERROR MESSAGE
NSERR_ACL_SRCMAC_MASK_LENGTH	911	0x38f	SrcMacMask length should be 12
NSERR_ACL_INVALID_NAME	1444	0x5a4	ERROR : ACL name length should be >=1 and <=128
NSERR_ACL_SRCMAC_MASK_PAIR	1455	0x5af	SrcMacMask should Contain consecutive 1s or 0s per octet and pair of 0s should not be present after pair of 1s

ADTCP ERRORS

ADTCP ERRORS

ERROR CODE	ERROR CODE(DECIMAL)	ERROR CODE(HEX)	ERROR MESSAGE
NSERR_ADTCP_PROFILE_NOT_FOUND	3781	0xec5	AdaptiveTCP profile not found
NSERR_ADTCP_PROFILE_INVALID	3792	0xed0	Invalid AdaptiveTCP profile
NSERR_ADTCP_CQARULE_NAME_INUSE	3793	0xed1	Rule name already in use

APPFLOW ERRORS

APPFLOW ERRORS

ERROR CODE	ERROR CODE(DECIMAL)	ERROR CODE(HEX)	ERROR MESSAGE
NSERR_APPFLOW_INVALIDPORT	3936	0xf60	Invalid collector port
NSERR_NO_SUCH_COLLECTOR	3938	0xf62	No matching collector
NSERR_APPFLOW_METRICS_ACTION	3939	0xf63	Set operation is not allowed on Appflow action with metricsLog option enabled.
NSERR_APPFLOW_NONZERO_REFCOUNT	3940	0xf64	The AppFlow entity is in use.
NSERR_APPFLOW_COLLECTOR_NAME_INUSE	3941	0xf65	AppFlow collector name already in use.
NSERR_APPFLOW_ACT_INVAL	3942	0xf66	No such AppFlow action exists.
NSERR_APPFLOW_NETPROF_REQD	3472	0xd90	NetProfile required for AppFlow collector within partition.
NSERR_COLLECTOR_TRANSPORT_MISMATCH	1220	0x4c4	All collectors attached to an action should have the same transport type.
NSERR_SAME_TYPE_PROFILE_BOUND	1221	0x4c5	Two profiles of same type cannot be bound.
NSERR_COLLECTOR_LOGSTREAM	1222	0x4c6	ANOMALOUS transaction option is

ERROR CODE	ERROR CODE(DECIMAL)	ERROR CODE(HEX)	ERROR MESSAGE
			valid only with Logstream collector.
NSERR_HTTP_STREAM	1225	0x4c9	Only REST type collectors are allowed with this profile.
NSERR_PROFILE_TYPE_MISMATCH	1226	0x4ca	Invalid profile type.
NSERR_ENABLE_IPFIX_LOGGING	1227	0x4cb	IPFIX logging not enabled.

AS ERRORS

AS ERRORS

ERROR CODE	ERROR CODE(DECIMAL)	ERROR CODE(HEX)	ERROR MESSAGE
NSERR_AS_NOSTARTURL	3120	0xc30	No such StartURL check
NSERR_AS_EXIST_STARTURL	3121	0xc31	The StartURL check is already in use
NSERR_AS_NODENYURL	3122	0xc32	No such DenyURL check
NSERR_AS_EXIST_DENYURL	3123	0xc33	The DenyURL check is already in use
NSERR_AS_NOCOOKIECONSISTENCY	3124	0xc34	No such CookieConsistency check

ERROR CODE	ERROR CODE(DECIMAL)	ERROR CODE(HEX)	ERROR MESSAGE
NSERR_AS_EXIST_COOKIECONSISTENCY	3125	0xc35	The CookieConsistency check is already in use
NSERR_AS_NOFIELDCONSISTENCY	3126	0xc36	No such FieldConsistency check
NSERR_AS_EXIST_FIELDCONSISTENCY	3127	0xc37	The FieldConsistency check is already in use
NSERR_AS_NOXSS	3128	0xc38	No such CrossSiteScripting check
NSERR_AS_EXIST_XSS	3129	0xc39	The CrossSiteScripting check is already in use
NSERR_AS_NOSQL	3130	0xc3a	No such SQLInjection check
NSERR_AS_EXIST_SQL	3131	0xc3b	The SQLInjection check is already in use
NSERR_AS_NOFIELDFORMAT	3132	0xc3c	No such FieldFormat check
NSERR_AS_EXIST_FIELDFORMAT	3133	0xc3d	The FieldFormat check is already in use
NSERR_AS_NOOBJECTEXPRESSION	3134	0xc3e	No such SafeObject check

ERROR CODE	ERROR CODE(DECIMAL)	ERROR CODE(HEX)	ERROR MESSAGE
NSERR_AS_EXIST_OBJECTEXPRESSION	3135	0xc3f	The SafeObject check is already in use
NSERR_AS_NOFIELDTYPE	3136	0xc40	No such FieldType. See /var/log/ns.log for more details.
NSERR_AS_NOT_SUPPORTED_VS	3137	0xc41	Specified policy can be bound only to HTTP/SSL/PROXY vserver
NSERR_AS_BAD_ACTION_STARTURL	3138	0xc42	Invalid StartURL Action
NSERR_AS_BAD_ACTION_DENYURL	3139	0xc43	Invalid DenyURL Action
NSERR_AS_BAD_ACTION_COOKIECONSISTENCY	3140	0xc44	Invalid CookieConsistency Action
NSERR_AS_BAD_ACTION_FIELDCONSISTENCY	3141	0xc45	Invalid FieldConsistency Action
NSERR_AS_BAD_ACTION_XSS	3142	0xc46	Invalid CrossSiteScripting Action
NSERR_AS_BAD_ACTION_SQL	3143	0xc47	Invalid SQLInjection Action
NSERR_AS_BAD_ACTION_FIELDFORMAT	3144	0xc48	Invalid FieldFormat Action

ERROR CODE	ERROR CODE(DECIMAL)	ERROR CODE(HEX)	ERROR MESSAGE
NSERR_AS_BAD_ACTION_OBJECTEXPRESSION	3145	0xc49	Invalid SafeObject Action
NSERR_AS_BAD_ACTION_BUFFEROVERFLOW	3146	0xc4a	Invalid BufferOverflow Action
NSERR_AS_BAD_ACTION_CCARD	3147	0xc4b	Invalid CreditCard Action
NSERR_AS_FIELDFORMAT_MIN_GT_MAX	3148	0xc4c	Minimum FieldFormat length cannot exceed maximum FieldFormat length
NSERR_AS_FIELDTYPE_BAD_NAME_LEN	3149	0xc4d	Invalid FieldType name length
NSERR_AS_BAD_COMMENT_LEN	3150	0xc4e	Invalid comment length
NSERR_AS_FIELDTYPE_BAD_REGEX_LEN	3151	0xc4f	Invalid regex length
NSERR_AS_SECURITY_CHECK_REQUIRED	3152	0xc50	Security check required
NSERR_AS_FIELDTYPE_BUILTIN	3153	0xc51	Built-in FieldTypes cannot be modified or deleted
NSERR_AS_BAD_ACTION_CONTENT_TYPE	3154	0xc52	Invalid ContentType Action

ERROR CODE	ERROR CODE(DECIMAL)	ERROR CODE(HEX)	ERROR MESSAGE
NSERR_AS_BAD_DEFAULT_CHARSET	3155	0xc53	Invalid default character set
NSERR_AS_BAD_COOKIECONSISTENCY_NAME	3156	0xc54	Invalid Cookie name
NSERR_AS_BAD_ERRORURL	3157	0xc55	Invalid ErrorURL
NSERR_AS_BAD_ENCODING_URL	3158	0xc56	Invalid encoding for URL
NSERR_AS_BAD_ENCODING_FIELDNAME	3159	0xc57	Invalid encoding for field name
NSERR_AS_BAD_ENCODING_COOKIENAME	3160	0xc58	Invalid encoding for cookie name
NSERR_AS_BAD_ENCODING_OBJECTEXPRESSION_NAME	3161	0xc59	Invalid encoding for SafeObject name
NSERR_AS_BAD_ENCODING_EXPRESSION	3162	0xc5a	Invalid encoding for expression
NSERR_AS_BAD_ENCODING_REGEX	3163	0xc5b	Invalid encoding for regex
NSERR_AS_MODIFIED_URL	3164	0xc5c	Replaced character that is not printable ASCII with escaped equivalent in URL

ERROR CODE	ERROR CODE(DECIMAL)	ERROR CODE(HEX)	ERROR MESSAGE
NSERR_AS_MODIFIED_OBJECTEXPRESSION	3165	0xc5d	Replaced character that is not printable ASCII with escaped equivalent in expression
NSERR_AS_CFFIELD_BAD_FIELDNAME_LEN	3166	0xc5e	Invalid confidential form field name length
NSERR_AS_CFFIELD_BAD_URL_LEN	3167	0xc5f	Invalid confidential form field URL length
NSERR_AS_NOCFFIELD	3168	0xc60	No such confidential form field
NSERR_AS_BAD_CFF_NAME	3169	0xc61	Name may not contain leading/ trailing spaces.
NSERR_AS_BAD_CFF_URL	3170	0xc62	URL may not contain leading/trailing spaces.
NSERR_AS_CFF_DUP	3171	0xc63	The confidential field is already in use.
NSERR_AS_BAD_ACTION_XDOS	3172	0xc64	Invalid XML Dos Action
NSERR_AS_EXIST_XML_DOS_URL	3173	0xc65	The XML DoS URL check is already in use.
NSERR_XML_URL_NOT_SUPPORTED	3174	0xc66	Only .* is supported for XML checks.
NSERR_AS_NO_XDOS_URL	3175	0xc67	No such URL exist for XDOS check.

ERROR CODE	ERROR CODE(DECIMAL)	ERROR CODE(HEX)	ERROR MESSAGE
NSERR_AS_INVALID_XML_DOS_CONF	3176	0xc68	Invalid configuration: xmlMaxFileSize can not be less than xmlMinFileSize when both checks are enabled.
NSERR_AS_BAD_ACTION_XML_SQLINJECTION	3177	0xc69	Invalid XML - SQLInjection Action.
NSERR_AS_BAD_ACTION_XML_XSS	3178	0xc6a	Invalid XML - CrossSiteScripting Action.
NSERR_AS_BAD_ACTION_XML_WELLFORMEDNESS	3179	0xc6b	Invalid XML - Format Action.
NSERR_AS_BAD_ACTION_PROFILE_TYPE	3180	0xc6c	Invalid Appsecure Profile Type.
NSERR_AS_MODIFIED_FIELDNAME	3181	0xc6d	Replaced character that is not printable ASCII with escaped equivalent in form field name
NSERR_AS_BAD_ACTION_WSI	3182	0xc6e	Invalid XML WS-I Action.
NSERR_AS_EXIST_XML_WSI_URL	3183	0xc6f	The XML WS-I URL check is already in use.
NSERR_AS_NO_WSI_URL	3184	0xc70	No such URL exist for WS-I check.

ERROR CODE	ERROR CODE(DECIMAL)	ERROR CODE(HEX)	ERROR MESSAGE
NSERR_AS_INVALID_XML_WSI_CONF	3185	0xc71	Invalid WS-I rule id in the list.
NSERR_AS_OBJECTNAME_TOO_BIG	3186	0xc72	Object too big.
NSERR_AS_OBJECT_NO_EXIST	3187	0xc73	Imported file does not exist [Please import the file before use]. See /var/log/ns.log for more details.
NSERR_AS_SERVER_NAME_TOO_BIG	3188	0xc74	Server name too big.
NSERR_AS_OBJECT_NOT_READABLE	3189	0xc75	Object not readable [Please make sure it exists].
NSERR_AS_NOPROFILE	3190	0xc76	No such profile.
NSERR_AS_BAD_ACTION_XML_ATTACHMENT	3191	0xc77	Invalid XML - Attachment Action
NSERR_AS_BAD_ACTION_MSGVAL	3192	0xc78	Invalid XML Message Validation Action.
NSERR_AS_EXIST_XML_MSGVAL_URL	3193	0xc79	The XML MSGVAL URL check is already in use.
NSERR_AS_NO_MSGVAL_URL	3194	0xc7a	No such URL exist for MSGVAL check.

ERROR CODE	ERROR CODE(DECIMAL)	ERROR CODE(HEX)	ERROR MESSAGE
NSERR_AS_INVALID_XML_MSGVAL_CONF	3195	0xc7b	Invalid MsgVal configuration.
NSERR_AS_BIND_XML_MSGVAL_CONF	3196	0xc7c	Error in message validation binding. For More details see log messages.
NSERR_AS_IMPORT_FAILED	3197	0xc7d	Importing the resource failed
NSERR_AS_OBJECT_EXIST	3198	0xc7e	Object already exists
NSERR_AS_INVALID_OPTION	3199	0xc7f	Invalid option
NSERR_AS_REMOVE_FAILED	3200	0xc80	Problem in removing resource
NSERR_AS_NO_RESOURCE	3201	0xc81	No such resource. Object you are trying to update or remove does not exist.
NSERR_AS_SHOW_FAILED	3202	0xc82	Problem in showing object
NSERR_AS_DEPENDENCY_FAILED	3203	0xc83	Problem in downloading dependencies
NSERR_AS_COMPILATION_FAILED	3204	0xc84	Problem compiling object. For more details see /var/log/ns.log file
NSERR_INVALID_OBJECT_NAME	3205	0xc85	Invalid name; names must begin with an alphanumeric

ERROR CODE	ERROR CODE(DECIMAL)	ERROR CODE(HEX)	ERROR MESSAGE
			character or underscore and must contain only alphanumerics, _, #, ., :, @, =or -
NSERR_AS_IMPORT_DOWNLOAD_FAILED	3206	0xc86	Problem in importing the object. Please check the DNS NameServer/Route settings and try again. For more details see /var/log/ns.log file
NSERR_AS_RESOURCE_INUSE	3207	0xc87	Can not remove resource. Resource is in use
NSERR_AS_BAD_MAPPING_DATA	3208	0xc88	Mapping Data Corrupted
NSERR_AS_PROFILE_CHANGE_HTML	3209	0xc89	HTML checks will not be applicable when profile type is XML
NSERR_AS_PROFILE_CHANGE_XML	3210	0xc8a	XML checks will not be applicable when profile type is HTML
NSERR_AS_IMPORT_ALREADY_INPROCESS	3211	0xc8b	Import failed. Another resource with the same name being processed
NSERR_AS_INVALID_XML_ERROR_OBJECT	3212	0xc8c	Invalid XML error object

ERROR CODE	ERROR CODE(DECIMAL)	ERROR CODE(HEX)	ERROR MESSAGE
NSERR_AS_XML_WELLFORMEDNESS_DISABLED	3213	0xc8d	XML Security checks can not be performed, once an XML message is found not wellformed.
NSERR_AS_INDIVIDUAL_IMPORT_LIMIT_EXCEEDED	3214	0xc8e	Import failed - importing file size greater than configured size limit
NSERR_AS_TOTAL_IMPORT_LIMIT_EXCEEDED	3215	0xc8f	Import failed - exceeding the configured total size limit on the imported objects
NSERR_AS_LEARNING_BUSY	3216	0xc90	Please wait for the learning database to finish updating
NSERR_AS_DEPRECATED_XML_WSI_RULE_R4003	3217	0xc91	WS-I Rule R4003 has been deprecated, it will be removed from the WS-I binding list.
NSERR_AS_EXIST_XML_ATTACHMENT_URL	3218	0xc92	The XML Attachment URL check is already in use.
NSERR_AS_NO_XML_ATTACHMENT_URL	3219	0xc93	No such URL exist for XML Attachment check.
NSERR_AS_BAD_ACTION_SOAP_FAULT	3220	0xc94	Invalid SOAP Fault Filtering Action.

ERROR CODE	ERROR CODE(DECIMAL)	ERROR CODE(HEX)	ERROR MESSAGE
NSERR_AS_XML_DOS_VALUE_OUT_OF_RANGE	3221	0xc95	XML DoS check value is not within the allowed range.
NSERR_AS_XML_MSGVAL_CLI_ERROR	3222	0xc96	-XMLValidateResponse requires one of [-XMLValidateSOAPEnvelope, -XMLWSDL, -XMLRequestSchema, -XMLResponseSchema]
NSERR_AS_INVALID_CUSTOM_SETTINGS_OBJECT	3223	0xc97	Import failed. Please check syntax of signature object
NSERR_AS_SESSION_TIMEOUT_LIFETIME_CONFLICT	3224	0xc98	The session lifetime cannot be less than the session timeout
NSERR_AS_CLASSIC_POLICY_ALREADY_BOUND	3225	0xc99	Failed to bind advanced policy because a classic one is already bound globally or to a vserver. Binding classic and advanced policies at the same time is not allowed
NSERR_AS_ADVANCED_POLICY_ALREADY_BOUND	3226	0xc9a	Failed to bind classic policy because an advanced one is already bound globally or to a vserver or to a user defined policylabel.

ERROR CODE	ERROR CODE(DECIMAL)	ERROR CODE(HEX)	ERROR MESSAGE
			Binding classic and advanced policies at the same time is not allowed
NSERR_AS_CUSTOM_SETTINGS_NO_INJECTION_TYPE	3227	0xc9b	Failed to set signature object. One of the
NSERR_AS_NOCSRF_TAG	3228	0xc9c	No such CrossSiteRequestForgery check
NSERR_AS_EXIST_CSRF_TAG	3229	0xc9d	The CrossSiteRequestForgery check is already in use
NSERR_AS_BAD_ACTION_CSRF_TAG	3230	0xc9e	Invalid CSRF Tag Action
NSERR_AS_TURNING_TAGGING_OFF_CSRF_TAG_ON	3231	0xc9f	Must set CSRF tagging check to nonebefore disabling form tagging.
NSERR_AS_TURNING_CSRF_TAG_ON_TAGGING_OFF	3232	0xca0	Form tagging must be enabled before turning on CSRF tag checks.
NSERR_AS_IMPORT_LIMIT_LESS_THAN_IMPORT_DIR_SIZE	3233	0xca1	Import limit should be greater than the current total imported objects size.
NSERR_AS_UNSUPPORTED_IMPORT_PROTOCOL	3234	0xca2	Unsupported import source protocol. Supported protocols are http, https and ftp

ERROR CODE	ERROR CODE(DECIMAL)	ERROR CODE(HEX)	ERROR MESSAGE
NSERR_AS_UNSUPPORTED_EXPORT_PROTOCOL	3235	0xca3	Unsupported export target protocol. Supported protocols are http and https
NSERR_AS_EXPORT_FAILED	3236	0xca4	Export failed. Check /var/log/ns.log for details.
NSERR_AS_BAD_LOCAL_FILE	3237	0xca5	Local file cannot use .. to backtrack up a directory.
NSERR_AS_INVALID_REGEX_CUSTOM_OBJECT	3238	0xca6	Invalid regular expression in custom object. Reverting to default settings.
NSERR_AS_BUILTIN_INVALID_OP	3239	0xca7	Invalid operation for built-in profile
NSERR_AS_INVALID_ADV_POLICY_STATE	3240	0xca8	Cannot bind advanced policy with state set to DISABLED
NSERR_AS_BUILTIN_NAMES_IN_CONF	3241	0xca9	Entities with names reserved for built-in profiles have been discarded
NSERR_AS_DEFAULT_PROFILE_IS_BYPASS	3242	0xcaa	The default profile is set to bypass traffic. It can be set using the set appfw settingscommand.
NSERR_AS_NO_LEARN_DENYURL	3243	0xcab	Learn is not supported as a denyURLAction.

ERROR CODE	ERROR CODE(DECIMAL)	ERROR CODE(HEX)	ERROR MESSAGE
NSERR_AS_INVALID_XML_FILE_SIZE	3244	0xcac	Invalid configuration: xmlMinFileSize / xmlMaxFileSize can not be less than 4 or greater than 1000000000.
NSERR_AS_NO_LEARN_BUFFEROVERFLOW	3246	0xcae	Learn is not supported as a bufferOvewflowAction.
NSERR_AS_NO_LEARN_XML_FORMAT	3247	0xcaf	Learn is not supported as an XMLFormatAction.
NSERR_AS_NO_LEARN_XML_SQLINJECTION	3251	0xcb3	Learn is not supported as an XMLSQLInjectionAction.
NSERR_AS_BAD_ACTION_MULTIPLE_HEADER	3252	0xcb4	Invalid Multiple Header Action. Only one of Block and KeepLast is allowed / None cannot be used with another Multiple Header Action.
NSERR_AS_NO_LEARN_XML_MSGVAL	3253	0xcb5	Learn is not supported as an XMLValidationAction.
NSERR_AS_NO_LEARN_XML_SOAP_FAULT	3254	0xcb6	Learn is not supported as an XMLSOAPFaultAction.

ERROR CODE	ERROR CODE(DECIMAL)	ERROR CODE(HEX)	ERROR MESSAGE
NSERR_AS_IMPORT_INTERNAL_ERROR	3255	0xcb7	Internal error while importing resource.
NSERR_AS_NETSVC_CONN_FAILED	3256	0xcb8	Critical internal error. please retry after some time.
NSERR_AS_VALIDATION_FAILED	3257	0xcb9	Problem validating Imported object against specification. For more details see /var/log/ns.log file.
NSERR_AS_CKI_TRANSFORM_DISABLED	3258	0xcba	This setting will not take effect until cookieTransforms is turned ON.
NSERR_AS_NOXMLXSS	3259	0xcbb	No such XMLXSS check
NSERR_AS_EXIST_XMLXSS	3260	0xcbc	The XMLXSS check is already in use
NSERR_AS_NOXMLSQL	3261	0xcbd	No such XMLSQLInjection check
NSERR_AS_EXIST_XMLSQL	3262	0xcbe	The XMLSQLInjection check is already in use
NSERR_AS_NO_ENDTAG	3263	0xcbf	Missing end tag in the imported signature file.

ERROR CODE	ERROR CODE(DECIMAL)	ERROR CODE(HEX)	ERROR MESSAGE
NSERR_AS_SIG_INVALID_RULEID	3264	0xcc0	Invalid rule id in imported signature.
NSERR_AS_SIG_INVALID_RULE_VERSION	3265	0xcc1	Invalid version in imported signature.
NSERR_AS_SIG_PARSE_ERROR	3266	0xcc2	Error parsing imported signatures.
NSERR_AS_SIG_INVALID_ATTRIBUTE	3267	0xcc3	Imported signatures have invalid attributes
NSERR_AS_SIG_MULTIPLE_LOCATIONS	3268	0xcc4	Imported signature patterns has multiple locations
NSERR_AS_SIG_INVALID_STATE	3269	0xcc5	Error parsing imported signatures. Invalid parse state.
NSERR_AS_CKI_ENCRYPT_METHOD_IS_NONE	3270	0xcc6	For security purposes, please set the NS encryptions parameter using the \ \"set ns encryptionParams\\" command.
NSERR_AS_UPDATE_FAILED	3271	0xcc7	Updating the resource failed
NSERR_AS_OBJECT_NOT_USED	3272	0xcc8	The specified object is not used
NSERR_AS_SIG_MULTIPLE_URLS	3273	0xcc9	Imported signature pattern location has multiple URLs

ERROR CODE	ERROR CODE(DECIMAL)	ERROR CODE(HEX)	ERROR MESSAGE
NSERR_AS_SIG_MULTIPLE_FIELD_NAMES	3274	0xcca	Imported signature pattern location has multiple field names
NSERR_AS_SIG_NO_FASTMATCH	3275	0xccb	A signature rule has no fastmatch pattern. See log for details.
NSERR_AS_PE_POLICY_NOTSUPPORTED	3276	0xccc	Not supported for classic policy.
NSERR_AS_AUDITLOG_POLICY_NOTSUPPORTED	3277	0xccd	Not supported for auditlog policy.
NSERR_AS_SIG_MULTIPLE_CREDIT_CARD_RULES	3278	0xcce	Imported signatures has multiple credit card patterns in Rule.
NSERR_AS_TURNING_TAGGING_OFF_SESSIONLESS_FFC_ON	3279	0xccf	Must disable sessionless field consistency checks before disabling form tagging.
NSERR_AS_TURNING_SESSIONLESS_FFC_ON_TAGGING_OFF	3280	0xcd0	Form tagging must be enabled before enabling sessionless field consistency checks.
NSERR_AS_SIG_INVALID_CREDIT_CARD_PATTERN	3281	0xcd1	Imported Signatures has invalid Credit Card pattern.
NSERR_AS_SIG_INVALID_SAFE_OBJECT_PATTERN	3282	0xcd2	Imported Signatures has invalid Safe Object pattern.

ERROR CODE	ERROR CODE(DECIMAL)	ERROR CODE(HEX)	ERROR MESSAGE
NSERR_AS_IMPORT_INVALID_TAR_ARCHIVE	3283	0xcd3	Imported archive is invalid. Please provide valid tar.gz archive.
NSERR_AS_SQLINJECTION_KEYWORD_INVALID_ATTRIBUTE	3284	0xcd4	Imported signatures has invalid SQL injection attributes.
NSERR_AS_SPLSTRING_INVALID_ATTRIBUTE	3285	0xcd5	Imported signatures has invalid special string attributes.
NSERR_AS_SQLINJECTION_KEYWORD_LITERAL_EXCEED_MAXLEN	3286	0xcd6	Maximum allowed length for type \ \"LITERAL\\" for keyword is 255.
NSERR_AS_SPLSTRING_LITERAL_EXCEED_MAXLEN	3287	0xcd7	Maximum allowed length for type \ \"LITERAL\\" for special string is 255.
NSERR_AS_XSLT_TRANFORM_LATEST	3288	0xcd8	Problem converting imported signature to the latest version. For more details see /var/log/ns.log file.
NSERR_AS_XSLT_TRANFORM_USER	3289	0xcd9	Problem converting imported signature with the given xslt file. For more details see /var/log/ns.log file.

ERROR CODE	ERROR CODE(DECIMAL)	ERROR CODE(HEX)	ERROR MESSAGE
NSERR_AS_SIG_RESP_BODY_EXPR_ERROR	3290	0xcda	Match type EXPRESSION is not supported in response rules.
NSERR_AS_NOTRUSTEDLEARNINGCLIENT	3291	0xcdb	No Such Trusted Learning Client.
NSERR_AS_EXIST_TRUSTEDLEARNINGCLIENT	3292	0xcdc	The Trusted Learning Client is already in use.
NSERR_AS_POLICY_ALLOWED_BINDPOINT_NONE	3293	0xcdd	Only bindpoint none is allowed for this policy.
NSERR_AS_XSS_DENIED_PATTERN_INVALID_ATTRIBUTE	3294	0xcde	Imported signatures has invalid denied pattern attribute.
NSERR_AS_XSS_DENIED_PATTERN_EXCEED_MAXLEN	3295	0xcdf	Maximum allowed length for type \ \"LITERAL\\" for denied pattern is 255.
NSERR_AS_XSS_ALLOWED_PATTERN_INVALID_ATTRIBUTE	3296	0xce0	Imported signatures has invalid allowed pattern attribute.
NSERR_AS_XSS_ALLOWED_PATTERN_EXCEED_MAXLEN	3297	0xce1	Maximum allowed length for type \ \"LITERAL\\" for allowed pattern is 255.

ERROR CODE	ERROR CODE(DECIMAL)	ERROR CODE(HEX)	ERROR MESSAGE
NSERR_AS_SIG_MULTIPLE_SUB_LOCATIONS	3298	0xce2	Imported signature pattern location has multiple sub locations
NSERR_AS_SIG_INTERNAL_ERROR	3299	0xce3	Internal error when configuring signatures.
NSERR_AS_SIG_URL_TAG_ERROR	3300	0xce4	Signature file has
NSERR_AS_SIG_HEADER_NAME_TAG_ERROR	3301	0xce5	Signature file has
NSERR_AS_SIG_COOKIE_NAME_TAG_ERROR	3302	0xce6	Signature file has
NSERR_AS_SIG_FIELD_NAME_TAG_ERROR	3303	0xce7	Signature file has
NSERR_AS_SIG_PATTERN_ADD_ERROR	3304	0xce8	Problem constructing signature pattern from file
NSERR_AS_SIG_NON_LITERAL_FASTMATCH	3305	0xce9	Signature file has \\"fastmatch\\" on a non-literal match. See /var/log/ns.log for more details
NSERR_AS_SIG_NEGATED_LITERAL_FASTMATCH	3306	0xcea	Signature file has \\"fastmatch\\" on a negated literal match.

ERROR CODE	ERROR CODE(DECIMAL)	ERROR CODE(HEX)	ERROR MESSAGE
NSERR_AS_SIG_MEM_ALLOC_FAILED	3307	0xceb	Memory allocation failed while loading signatures.
NSERR_AS_SIG_TOO_MANY_PATTERNS	3308	0xcec	Signature file has a rule with too many patterns.
NSERR_AS_SIG_LITERAL_HEX_PARSE_ERROR	3309	0xced	Signature file has a pattern with hex byte syntax error.
NSERR_AS_SIG_PCRE_COMPILE_ERROR	3310	0xcee	Signature file has an invalid PCRE.
NSERR_INVALID_SIGNATURE_NAME	3311	0xcef	Invalid signature name; it must begin with an alphanumeric character or underscore and must contain only alphanumerics, _, ., or -
NSERR_AS_SIGNATURE_MERGE	3312	0xcf0	Problem merging imported signature rules with the existing signature rules. For more details see /var/log/ns.log file.
NSERR_AS_RECORDER_MEM_ALLOC_ERROR	3313	0xcf1	Cannot allocate memory for requested recorder size.

ERROR CODE	ERROR CODE(DECIMAL)	ERROR CODE(HEX)	ERROR MESSAGE
NSERR_AS_SIG_INVALID_FILE_VERSION	3314	0xcf2	Signature file has an invalid SignaturesFile versionvalue.
NSERR_AS_AUTO_SIGNATURE_UPDATED_FAILED	3315	0xcf3	Updating of signatures failed. For more details see /var/log/ns.log file.
NSERR_AS_NO_CONTENT_TYPE	3316	0xcf4	No Such content-type.
NSERR_AS_EXIST_LOG_EXPRESSION	3317	0xcf5	LogExpression is already in use.
NSERR_AS_POLICY_SET_NOTALLOWED	3318	0xcf6	Set disallowed as advance profile can only be associated to policy that is either not bound or bound to spotted vservers.
NSERR_AS_SIGNATURE_URL_NOT_ACCESSIBLE	3319	0xcf7	Signature URL set in set appfw settingsis not accessible. Please check the DNS NameServer/ Route settings and try again.
NSERR_AS_NODEGROUP_UNBIND_VSERVER_NOTALLOWED	3320	0xcf8	Unbinding vserver from nodegroup disallowed as vserver has appfw policy bound to it.
NSERR_AS_NODEGROUP_BIND_NODE_NOTALLOWED	3321	0xcf9	Nodegroup could be associated with only one node if it has a

ERROR CODE	ERROR CODE(DECIMAL)	ERROR CODE(HEX)	ERROR MESSAGE
			vserver that has appfw policy bound to it.
NSERR_AS_BUILT_IN_OBJECT_ERROR	3322	0xcfa	Cannot add or remove built in objects.
NSERR_AS_SIGNATURE_VALIDATE_FAILED	3323	0xcfb	Imported signature is failed to validate against sha1 file. For more details see /var/log/ns.log file.
NSERR_AS_INVALID_PARAMETER	3325	0xcfd	Sha1 parameter is allowed to use only with Import Appfw Signatures command
NSERR_AS_LEARNING_DB_ERROR	3326	0xcfe	Learning database is either corrupt or does not exist.
NSERR_AS_VALIDATION_SIGNATURE_MAPPING_FAILED	3327	0xcff	SignatureMapping.xml is invalid. Failed to validate SignatureMapping.xml against its schema file.
NSERR_AS_EXIST_CONTENT_TYPE	3332	0xd04	Content-type is already in use.
NSERR_AS_SIG_MISSING_MAX_MATCH_LEN	3333	0xd05	Signature file has a rule with a response body pattern without the required maxMatchLength attribute

ERROR CODE	ERROR CODE(DECIMAL)	ERROR CODE(HEX)	ERROR MESSAGE
NSERR_AS_NOT_ALLOWED_CONTENT_TYPE	3334	0xd06	Content-types application/x-www-form-urlencodedor multipart/form-dataor text/x-gwt-rpcare allowed by default and do not need to be explicitly configured in the allowed list
NSERR_AS_TOTAL_IMPORT_SIZE_LIMIT_EXCEEDED	3335	0xd07	Update is successful but update operation of this import object exceeded total import size limit. Now total import size limit is changed to newer value.
NSERR_AS_MAX_TOTAL_IMPORT_LIMIT_EXCEEDED	3336	0xd08	Update operation of this import object exceeded maximum total import size limit. Please remove the object.
NSERR_INVALID_DEFAULT_SIGNATURE_NAME	3337	0xd09	Signature name must only be *Default Signaturesor *Xpath Injection Patterns
NSERR_AS_SIG_OFFSET_GTE_DEPTH	3338	0xd0a	Signature file has a pattern with offset attribute greater than or equal to the depth attribute. See /var/log/ns.log for more details

ERROR CODE	ERROR CODE(DECIMAL)	ERROR CODE(HEX)	ERROR MESSAGE
NSERR_AS_SQLWILDCHAR_INVALID_ATTRIBUTE	3339	0xd0b	Imported signatures has invalid wild char attributes.
NSERR_AS_WILDCHAR_LITERAL_EXCEED_MAXLEN	3340	0xd0c	Maximum allowed length for type \ \"LITERAL\\" for wild char exceeded.
NSERR_AS_SIG_EXCLAMATION_LITERAL	3341	0xd0d	Cannot have an exclamation point alone as a literal
NSERR_AS_REMOVE_FAILED_RENAME_FAILED	3342	0xd0e	Problem in removing resource - file rename failed
NSERR_AS_ARCHIVE_PROFILE_FAILED	3343	0xd0f	Some commands failed during archive profile. Please see /var/log/ns.log for more details.
NSERR_AS_RESTORE_PROFILE_FAILED	3345	0xd11	Some commands failed during restore profile. Please see /var/log/ns.log for more details.
NSERR_AS_EMPTY_CONTENT_TYPE	3352	0xd18	Empty content type
NSERR_AS_SIG_FASTMATCH_BAD_LOCATION	3359	0xd1f	A signature rule has fastmatch pattern in a bad location. See log for details.
NSERR_AS_SESSIONLESS_FFC_POST_ONLY	3360	0xd20	POST ONLYis same as option ONin streaming

ERROR CODE	ERROR CODE(DECIMAL)	ERROR CODE(HEX)	ERROR MESSAGE
NSERR_AS_TURNING_TAGGING_OFF_FFC_ON	3361	0xd21	Form-tagging cannot be disabled and remains enabled as long as field consistency check is enabled.
NSERR_AS_NO_CREDIT_CARD_NUMBER	3363	0xd23	No such credit card number.
NSERR_AS_EXIST_CREDIT_CARD_NUMBER	3364	0xd24	Credit card number is already in use.
NSERR_AS_NO_SUCH_IMPORTED_OBJECT	3365	0xd25	No such imported object.
NSERR_AS_LEARNING_SQL_ERROR	3366	0xd26	SQL Error when performing learning operation
NSERR_AS_IMPORT_FILE_IS_EMPTY	3367	0xd27	Imported file is empty
NSERR_AS_IMPORT_OVERWRITE_OPTION_NOT_SUPPORTED	3368	0xd28	Overwrite option is not supported.
NSERR_AS_BAD_INSPECT_CONTENT_TYPE	3369	0xd29	Invalid inspection content type
NSERR_AS_GEO_LOCATION_LOGGING_ENABLED	3376	0xd30	GeoLocationLogging only works when CEFLogging is enabled

ERROR CODE	ERROR CODE(DECIMAL)	ERROR CODE(HEX)	ERROR MESSAGE
NSERR_AS_NO_SUCH_WSDL_OBJECT	3377	0xd31	WSDL does not exist
NSERR_AS_NO_SUCH_XMLSCHEMA_OBJECT	3378	0xd32	XMLSchema does not exist
NSERR_AS_NO_SUCH_XMLERRORPAGE_OBJECT	3379	0xd33	XMLErrorPage does not exist
NSERR_AS_NO_SUCH_SIGNATURE_OBJECT	3380	0xd34	Signature does not exist
NSERR_AS_NO_SUCH_ARCHIVE_OBJECT	3381	0xd35	Archive does not exist
NSERR_AS_NO_SUCH_HTMLERRORPAGE_OBJECT	3382	0xd36	HTMLErrorPage does not exist
NSERR_AS_XSS_ALLOWED_TAG_INVALID_ATTRIBUTE	3393	0xd41	Xss allowed Tag has invalid attribute
NSERR_AS_XSS_ALLOWED_ATTR_INVALID_ATTRIBUTE	3394	0xd42	Xss allowed attribute has invalid attribute
NSERR_AS_SQL_TRANSFORM_INVALID_ATTRIBUTE	3395	0xd43	SQL transform has invalid attribute
NSERR_AS_SRC_NOT_DEFAULT	3403	0xd4b	?
NSERR_AS_SRC_INVALID	3404	0xd4c	Invalid source

ERROR CODE	ERROR CODE(DECIMAL)	ERROR CODE(HEX)	ERROR MESSAGE
NSERR_AS_IMPORT_NOT_DEFAULT_WITH_SHA1	3405	0xd4d	Name should be *Default Signatures when -sha1 is specified
NSERR_AS_WAR_UPDATE_FOLLOWED_BY_IMPORT	3407	0xd4f	Execute update command to push the object to packet engine so that the imported object is applied.
NSERR_AS_CUSTOM_SETTINGS_SQL_MULTIPLE_INJECTION_TYPE	3423	0xd5f	Failed to set signature object. Multiple injection tags of sql type are not allowed
NSERR_AS_BAD_CCARD_FLAGS	3439	0xd6f	Invalid Credit Card Value
NSERR_AS_NO_LOGEXPRESSION	3790	0xece	No Such Log Expression
NSERR_AS_RFCPROFILE_BAD_ACTION	3808	0xee0	Bad rfc profile action
NSERR_AS_RFCPROFILE	3809	0xee1	Invalid rfc profile
NSERR_AS_BAD_MALFORMED_REQ_ACTION	3814	0xee6	Invalid malformed request Action
NSERR_AS_SIG_INVALID_RULE_YEAR	3822	0xeee	Invalid year attribute value in imported signature.

BASE ERRORS

BASE ERRORS

ERROR CODE	ERROR CODE(DECIMAL)	ERROR CODE(HEX)	ERROR MESSAGE
NSERR_NOTRUNNING	256	0x100	Citrix ADC not running
NSERR_PERM	257	0x101	Operation not permitted
NSERR_NOENT	258	0x102	No such resource
NSERR_SRCH	259	0x103	No such process
NSERR_INTR	260	0x104	Interrupted system call
NSERR_IO	261	0x105	Input/output error
NSERR_NXIO	262	0x106	Device not configured
NSERR_2BIG	263	0x107	Argument list too long
NSERR_NOEXEC	264	0x108	Exec format error
NSERR_BADF	265	0x109	Bad file descriptor
NSERR_CHILD	266	0x10A	No child processes
NSERR_DEADLK	267	0x10B	Resource deadlock avoided
NSERR_NOMEM	268	0x10C	Cannot allocate memory
NSERR_ACCES	269	0x10D	Permission denied
NSERR_FAULT	270	0x10E	Bad address

ERROR CODE	ERROR CODE(DECIMAL)	ERROR CODE(HEX)	ERROR MESSAGE
NSERR_NOTBLK	271	0x10F	Block device required
NSERR_BUSY	272	0x110	Device busy
NSERR_EXIST	273	0x111	Resource already exists
NSERR_XDEV	274	0x112	Cross-device link
NSERR_NODEV	275	0x113	Operation not supported by device
NSERR_NOTDIR	276	0x114	Not a directory
NSERR_ISDIR	277	0x115	Is a directory
NSERR_INVAL	278	0x116	Invalid argument
NSERR_NFILE	279	0x117	Too many open files in system
NSERR_MFILE	280	0x118	Too many open files
NSERR_NOTTY	281	0x119	Inappropriate ioctl for device
NSERR_TXTBSY	282	0x11A	Text file busy
NSERR_FBIG	283	0x11B	File too large
NSERR_NOSPACE	284	0x11C	No space left on device
NSERR_SPIPE	285	0x11D	Illegal seek
NSERR_ROFS	286	0x11E	Read-only file system
NSERR_MLINK	287	0x11F	Too many links

ERROR CODE	ERROR CODE(DECIMAL)	ERROR CODE(HEX)	ERROR MESSAGE
NSERR_PIPE	288	0x120	Broken pipe
NSERR_DOM	289	0x121	Numerical argument out of domain
NSERR_RANGE	290	0x122	Result too large
NSERR_AGAIN	291	0x123	Resource temporarily unavailable
NSERR_INPROGRESS	292	0x124	Operation now in progress
NSERR_ALREADY	293	0x125	Operation already in progress
NSERR_NOTSOCK	294	0x126	Socket operation on non-socket
NSERR_DESTADDRREQ	295	0x127	Destination address required
NSERR_MSGSIZE	296	0x128	Message too long
NSERR_PROTOTYPE	297	0x129	Protocol wrong type for socket
NSERR_NOPROTOOPT	298	0x12A	Protocol not available
NSERR_PROTONOSUPPORT	299	0x12B	Protocol not supported
NSERR_SOCKTNOSUPPORT	300	0x12C	Socket type not supported
NSERR_OPNOTSUPP	301	0x12D	Operation not supported

ERROR CODE	ERROR CODE(DECIMAL)	ERROR CODE(HEX)	ERROR MESSAGE
NSERR_PFNOSUPPORT	302	0x12E	Protocol family not supported
NSERR_AFNOSUPPORT	303	0x12F	Address family not supported by protocol family
NSERR_ADDRINUSE	304	0x130	Address already in use
NSERR_ADDRNOTAVAIL	305	0x131	Cant assign requested address
NSERR_NETDOWN	306	0x132	Network is down
NSERR_NETUNREACH	307	0x133	Network is unreachable
NSERR_NETRESET	308	0x134	Network dropped connection on reset
NSERR_CONNABORTED	309	0x135	Software caused connection abort
NSERR_CONNRESET	310	0x136	Connection reset by peer
NSERR_NOBUFS	311	0x137	No buffer space available
NSERR_ISCONN	312	0x138	Socket is already connected
NSERR_NOTCONN	313	0x139	Socket is not connected
NSERR_SHUTDOWN	314	0x13A	Cant send after socket shutdown
NSERR_TOOMANYREFS	315	0x13B	Resource in use

ERROR CODE	ERROR CODE(DECIMAL)	ERROR CODE(HEX)	ERROR MESSAGE
NSERR_TIMEDOUT	316	0x13C	Operation timed out
NSERR_CONNREFUSED	317	0x13D	Connection refused
NSERR_LOOP	318	0x13E	Too many levels of symbolic links
NSERR_NAMETOOLONG	319	0x13F	File name too long
NSERR_HOSTDOWN	320	0x140	Host is down
NSERR_HOSTUNREACH	321	0x141	No route to host
NSERR_NOTEMPTY	322	0x142	Directory not empty
NSERR_PROCLIM	323	0x143	Too many processes
NSERR_USERS	324	0x144	Too many users
NSERR_DQUOT	325	0x145	Disc quota exceeded
NSERR_STALE	326	0x146	Stale NFS file handle
NSERR_REMOTE	327	0x147	Too many levels of remote in path
NSERR_BADRPC	328	0x148	RPC struct is bad
NSERR_RPCMISMATCH	329	0x149	RPC version wrong
NSERR_PROGUNAVAIL	330	0x14A	RPC prog. not avail
NSERR_PROGMISMATCH	331	0x14B	Program version wrong

ERROR CODE	ERROR CODE(DECIMAL)	ERROR CODE(HEX)	ERROR MESSAGE
NSERR_PROCUNAVAIL	332	0x14C	Bad procedure for program
NSERR_NOLCK	333	0x14D	No locks available
NSERR_NOSYS	334	0x14E	System Call not supported; Possible Reason: Citrix ADC is not running
NSERR_FTYPE	335	0x14F	Inappropriate file type or format
NSERR_AUTH	336	0x150	Authentication error
NSERR_NEEDAUTH	337	0x151	Need authenticator
NSERR_WOULDBLOCK	338	0x152	Operation would block
NSERR_NOCODE	339	0x153	Feature is not implemented
NSERR_NOTSUSER	340	0x154	Not super-user
NSERR_BIGDATA	341	0x155	Data size is too big
NSERR_SMALLDATA	342	0x156	Data size is too small
NSERR_NOMORENT	343	0x157	No more entry in table
NSERR_NOSERVICE	344	0x158	No Service
NSERR_OSERROR	345	0x159	Operating system error

ERROR CODE	ERROR CODE(DECIMAL)	ERROR CODE(HEX)	ERROR MESSAGE
NSERR_NONEXPCMD	346	0x15A	Unexpected command
NSERR_CMDPROPFAIL	347	0x15B	Command propagation failed
NSERR_TOOMANYNODES	348	0x15C	Too many nodes
NSERR_SECONDARYFAIL	349	0x15D	Command failed on secondary node, but succeeded on primary node. Configuration will be synchronized to ensure secondary and primary have same configuration.
NSERR_INVALBACKUP	350	0x15E	Invalid backup vserver
NSERR_NOSERVER	351	0x15F	No such server
NSERR_LOGINREQD	352	0x160	Login is required
NSERR_RPCINVAL	353	0x161	RPC invalid argument
NSERR_NOUSER	354	0x162	Invalid username or password
NSERR_INVALPASSWD	355	0x163	Invalid username or password
NSERR_LICENSE	356	0x164	Feature(s) not licensed

ERROR CODE	ERROR CODE(DECIMAL)	ERROR CODE(HEX)	ERROR MESSAGE
NSERR_DEFERRED	357	0x165	The command is stored for later execution
NSERR_PROPAUTHFAIL	358	0x166	Command propagation failed due to user authentication problems
NSERR_NODELSUSER	359	0x167	The superuser cannot be removed
NSERR_NOMODSUSER	360	0x168	Permissions for this superuser cannot be modified
NSERR_INVALNODEID	362	0x16A	Invalid node ID specified
NSERR_NOTOPHA	363	0x16B	Operation not permitted on stand-alone node
NSERR_NOOPPEERBAD	364	0x16c	Operation not possible due to invalid peer state. Rectify and retry.
NSERR_NOOPBAD	365	0x16d	Operation not possible as system state is invalid. Use show node for more information.
NSERR_NOPNOW	366	0x16e	Operation not possible now. Please wait for system to stabilize before retrying.

ERROR CODE	ERROR CODE(DECIMAL)	ERROR CODE(HEX)	ERROR MESSAGE
NSERR_NOOPPRI	367	0x16f	Operation permitted only on secondary node.
NSERR_NOOPSEC	368	0x170	Operation permitted only on primary node.
NSERR_REDIRECT	369	0x171	Redirect request (VIP down)
NSERR_BUFOVERFLOW	370	0x172	Buffer overflow occurred
NSERR_NOUSERPOLICY	371	0x173	No command policy defined, permission denied
NSERR_NOSYSGROUP	372	0x174	System group does not exist
NSERR_NOSYSCMDPOL	373	0x175	System command policy does not exist
NSERR_HAIPV6PT	374	0x176	IPV6 Feature cannot be disabled while HA is running over IPv6.
NSERR_HANSIPV6	375	0x177	IPv6 NSIP cannot be changed while HA is running over IPv6.
NSERR_NSIPV6ACTIVE	376	0x178	IPv6 NSIP is not in ACTIVE state, rectify and retry.
NSERR_RTMON_STANDALONE	377	0x179	Unbind all route monitors before making node standalone.

ERROR CODE	ERROR CODE(DECIMAL)	ERROR CODE(HEX)	ERROR MESSAGE
NSERR_IPV6FEAT_DISABLED	378	0x17a	IPv6 Feature Disabled
NSERR_DHCP_PBR_DEPENDENT	379	0x17b	PBR rule present with nexthop dependent on released DHCP lease
NSERR_IPSECPROF_PERM	380	0x17c	Tunnel protocol IPSEC must have ipsec profile set to none.
NSERR_IFPAIR_BOUND	381	0x17d	Interface(s) already member of other interface pair
NSERR_IFPAIR_NOTPAIR	382	0x17e	Number of member interfaces need to be exactly 2
NSERR_PBR_SETIPTUNNEL	383	0x17f	SET operations on PBR for changing iptunnel not allowed. Remove and add PBR with new iptunnel
NSERR_IPRANGENOTALLOWD	394	0x18A	Range not allowed for wildcard IP
NSERR_IVALIDIPRANGE	395	0x18B	Invalid range value
NSERR_IPRANGEMAXLIMIT	396	0x18C	Range value greater than maximum limit 254

ERROR CODE	ERROR CODE(DECIMAL)	ERROR CODE(HEX)	ERROR MESSAGE
NSERR_IP_NOT_EXIST	397	0x18D	IP does not exist for default or specified td
NSERR_TOOMANYRULES	398	0x18E	Maximum number of ACLs on the system has been exceeded
NSERR_PASSWD_LEN_MIN_8	399	0x18F	Password too short - minimum length is 8 characters
NSERR_SYNC_DISABLED	410	0x19A	Synchronization disabled
NSERR_NODE_DISABLED	411	0x19B	Node disabled
NSERR_SYNC_PROGRESS	412	0x19C	Synchronization is in progress
NSERR_ADNS_PERM	413	0x19D	Operation not permitted on ADNS service
NSERR_NOINC_IPSAMESUBNET	414	0x19E	Peer IP should be on the NSIP subnet if INC mode is disabled
NSERR_INVALID_PEERIP	415	0x19F	Not a valid peer IP
NSERR_REDIRECT307	416	0x1A0	Redirect request (send 307 temporary redirect
NSERR_INVALHOSTNAME	417	0x1A1	Invalid hostname

ERROR CODE	ERROR CODE(DECIMAL)	ERROR CODE(HEX)	ERROR MESSAGE
NSERR_REWRITE_NOT_SUPPORTED	418	0x1A2	Port rewrite not supported on this vserver type
NSERR_IPCHG_DENY	419	0x1A3	IP address change not permitted on this entity
NSERR_IPCHG_GSLB	420	0x1A4	GSLB local service IP cannot be changed. Please change the corresponding vservers IP
NSERR_GSLB_IPCHG	421	0x1A5	corresponding GSLB service/GSLB service item IP has also been changed.
NSERR_NOAUDITSRVC	422	0x1A6	No audit service running on the specified port
NSERR_LACPKEY_NOTSET	423	0x1A7	LACP key not set
NSERR_CHANNEL_INUSEBYLACP	424	0x1A8	Channel in use by LACP
NSERR_LACPENABLED	425	0x1A9	LACP enabled interface cannot be bound manually to a channel
NSERR_IFACEMANUALLYBOUND	426	0x1AA	LACP cannot be enabled, interface manually bound to a channel

ERROR CODE	ERROR CODE(DECIMAL)	ERROR CODE(HEX)	ERROR MESSAGE
NSERR_INTRECINUSE	427	0x1AB	Domain name for the reverse domain name already exists
NSERR_PUBIPPORT_VIP_CONFLICT	428	0x1AC	Public IP, port conflict with another entity bound to the same GSLB vserver
NSERR_IPPORT_VIP_BOUND	429	0x1AD	IP, port conflict with an entity already bound to the vserver
NSERR_NO_BACKUP_VIP	430	0x1AE	No such backup vserver
NSERR_REQ_SET_ARGS	431	0x1AF	No arguments to set
NSERR_SVCGRP_MEMBER_NAMECONFLICT	432	0x1B0	Name conflicts with an existing service or service group member name
NSERR_SERVER_NAME_EXIST	433	0x1B1	Server name already exists
NSERR_MAX_SERVICE_BINDING_ON_VSERVER	434	0x1B2	Maximum services bound to vserver exceeded
NSERR_MAX_SVC_ENTITY_BINDING_ON_SVCGROUP	435	0x1B3	Maximum services bound to service group exceeded
NSERR_FSMALL	436	0x1B4	File too small
NSERR_INTOFLOW	437	0x1B5	Integer overflow

ERROR CODE	ERROR CODE(DECIMAL)	ERROR CODE(HEX)	ERROR MESSAGE
NSERR_ASYNC_BLOCKED	438	0x1B6	Expression evaluation has blocked for body accumulation
NSERR_SACL_CLEARPENDING	439	0x1B7	Clear in progress
NSERR_SACL_NAME_EXISTS	440	0x1B8	An ACL with the same name exists
NSERR_SACL_SUPERSET_EXISTS	441	0x1B9	Operation not permitted. (An overlapping rule already exists)
NSERR_SACL_SUBSET_EXISTS	442	0x1BA	Operation not permitted. (A rule which is subset of this rule already exists)
NSERR_NOINC_ROUTEMONITOR	443	0x1BB	Route monitors should be added only if INC mode is enabled
NSERR_SESSION_EXPIRED	444	0x1BC	Session expired or killed. Please login again
NSERR_SESSION_EXCEEDED	445	0x1BD	Session limit exceeded
NSERR_CFE_CONN_EXCEEDED	446	0x1BE	Connection limit to CFE exceeded
NSERR_CFE_KILLSELF	449	0x1C1	Invalid session id. Current session cannot be killed

ERROR CODE	ERROR CODE(DECIMAL)	ERROR CODE(HEX)	ERROR MESSAGE
NSERR_CFE_INCOMPLETESESSION	450	0x1C2	Invalid session. Response/challenge is incomplete
NSERR_ACL_ALREADYINSTATE	451	0x1C3	Entity already in this state
NSERR_CFE_SESSION_NOEXIST	452	0x1C4	Session does not exist
NSERR_SYSGROUP_USER_EXISTS	453	0x1C5	User already bound to system group
NSERR_SYSGROUP_POLICY_EXISTS	454	0x1C6	Policy already bound to system group
NSERR_SLEEP	455	0x1C7	Sleep process sleep
NSERR_PPEDIE	456	0x1C8	Your PPE is no more
NSERR_NOCONN_CMDPROPFAIL	457	0x1C9	Unable to establish connection with the secondary. Command propagation failed
NSERR_TIMEOUT_SECONDARY	458	0x1CA	There is no response from the secondary. Propagation timed out
NSERR_RPC_TIMEOUT	459	0x1CB	RPC timeout
NSERR_NOT_SUPPORTED	460	0x1CC	Feature not supported in this release
NSERR_NOENT_VSERVER	461	0x1CD	No such Vserver

ERROR CODE	ERROR CODE(DECIMAL)	ERROR CODE(HEX)	ERROR MESSAGE
NSERR_NOENT_SVC	462	0x1CE	No such service
NSERR_NOENT_SVC_SVCGRP	463	0x1CF	No such service/ serviceGroup
NSERR_HA_NOV4_NETMASK	464	0x1D0	IPv4 netmask is not required for IPv6
NSERR_MAX_VSEVRVER_BINDINGS_TO_SERVICE	465	0x1D1	Maximum vservers bound to a service exceeded
NSERR_SACL6_CLEARPENDING	466	0x1D2	Simple ACL6 Clear in progress
NSERR_SACL6_NAME_EXISTS	467	0x1D3	Simple ACL6 with the same name exists
NSERR_SACL6_SUPERSET_EXISTS	468	0x1D4	Operation not permitted. (An overlapping rule already exists)
NSERR_SACL6_SUBSET_EXISTS	469	0x1D5	Operation not permitted. (A rule which is subset of this rule already exists)
NSERR_NO_BACKUP_VIP_BOUND	470	0x1D6	No backupVserver bound
NSERR_SESSION_EXPIRED_REDIRECT	471	0x1D7	Session expired, Please login again
NSERR_NOENT_VLAN	472	0x1D8	Vlan does not exist

ERROR CODE	ERROR CODE(DECIMAL)	ERROR CODE(HEX)	ERROR MESSAGE
NSERR_MAX_SVC_ENTITY	473	0x1D9	Maximum service limit reached
NSERR_CL_EXIST	474	0x1DA	Resource already exists in cluster
NSERR_OWNERNODE_NOTALLOWED	475	0x1DB	Can not specify owner node.
NSERR_OWNERNODE_NOT_SPECIFIED	476	0x1DC	Owner node is not specified.
NSERR_NOT_SUPPINCLUSTER	477	0x1DD	Operation not supported in Cluster.
NSERR_VSERVER_TYPE_MISMATCH	478	0x1DE	Vserver type mismatch.
NSERR_ITER_END	479	0x1DF	End of iteration reached.
NSERR_CONNECT_TO_CLUSTER_IP	1382	0x566	Please connect to cluster IP and run showtechsupport on that node.
NSERR_LB_AUDIT_IP_DENY	1383	0x567	Changing IP address of LB vserver not permitted when Audit server also referring to same server_info .
NSERR_IC_OFF_DISABLING_FEO	1384	0x568	IC is disabled, but FEO is enabled.
NSERR_SYNC_VLAN_EXIST	1385	0x569	Please unset sync vlan, before removing the HA node.

ERROR CODE	ERROR CODE(DECIMAL)	ERROR CODE(HEX)	ERROR MESSAGE
NSERR_CMDPROP_DISABLED	1450	0x5AA	Command propagation disabled.
NSERR_NOTHEME	1451	0x5AB	Portal theme doesnt exist
NSERR_THEMEEXISTS	1452	0x5AC	Portal theme already exists
NSERR_THEME_INUSE	1453	0x5AD	Portal theme in use
NSERR_NO_CONTENT	1454	0x5AE	204 No Content
NSERR_ARP_IFACE_NODE_MISMATCH	1440	0x5a0	Interface and node mismatch
NSERR_HA_BOTH_PRIMARY	1441	0x5a1	Cannot failover as Node State set to STAYPRIMARY
NSERR_PROP_DISABLED	1448	0x5a8	Propagation disabled
NSERR_HA_FAILOVER_PERM	1449	0x5a9	Failover not permitted as one node is up and another node is down
NSERR_NOEULA	938	0x3AA	Eula name doesnt exist
NSERR_EULAEXISTS	939	0x3AB	Eula with same name already exists

ERROR CODE	ERROR CODE(DECIMAL)	ERROR CODE(HEX)	ERROR MESSAGE
NSERR_EULA_INUSE	940	0x3AC	Eula name in use
NSERR_SYSLOG_TCP_CONFIG_ERR	1459	0x5B3	tcpProfileName/ maxLogDataSizeToHold cannot be set when transport is UDP
NSERR_SYSLOG_LOOPBACK_OR_LOCAL_IP_NOTALLOWED	1460	0x5B4	TCP transport is not allowed for local logging(when serverIP is configured as Loopback IP or NetScalers local IP)
NSERR_MAX_CONC_SAVECONFIG	1461	0x5B5	Maximum number of concurrent save configs are in progress. Try again later.
NSERR_SYSLOG_LBMETHOD_NOTALLOWED	1462	0x5B6	LB method AUDITLOGHASH can be set only for service type SYSLOGTCP/ SYSLOGUDP
NSERR_SYSLOG_PERSISTENCE_NOTSUPPORTED	1463	0x5B7	Persistency is not supported for SYSLOGTCP and SYSLOGUDP service types
NSERR_SYSLOG_HASHLBMETHODS_NOTALLOWED	1464	0x5B8	Hash based LB methods which depend on packet parameters cannot

ERROR CODE	ERROR CODE(DECIMAL)	ERROR CODE(HEX)	ERROR MESSAGE
			be set for SYSLOGTCP/ SYSLOGUDP service types
NSERR_SYSLOG_LEASTCONN_NOTSUPPORTED	1467	0x5BB	LEASTCONNECTION algorithm is not supported for SYSLOGTCP/ SYSLOGUDP service types
NSERR_SVC_EXISTS	1468	0x5BC	Service exists with this server information
NSERR_SYSLOG_NOLBVIP	1469	0x5BD	No such LB vserver of type SYSLOGUDP/ SYSLOGTCP exists
NSERR_SYSLOG_LBPROTOCOL_NOTALLOWED	1470	0x5BE	LB vserver of type other than SYSLOGUDP/ SYSLOGTCP cannot be set to syslogaction
NSERR_SYSLOG_TRANSPORT_UNSET_NOTALLOWED	1471	0x5BF	Transport cannot be unset when LB vserver name is configured
NSERR_SYSLOG_TRANSPORT_MISMATCH	1472	0x5C0	Mismatch in transport type of LB vserver and syslog action

ERROR CODE	ERROR CODE(DECIMAL)	ERROR CODE(HEX)	ERROR MESSAGE
NSERR_SYSLOG_LBVSRVR_INUSE_BY_SYSACT	1473	0x5C1	The LB vserver is being accessed by syslog action
NSERR_SYSLOG_LBVSRVR_RENAME_NOTSUPPORTED	1474	0x5C2	Renaming LB vserver is not supported for SYSLOGUDP/ SYSLOGTCP service types
NSERR_SYSLOG_INVALID_SERVIP	1475	0x5C3	ServerIP cannot be configured as 0.0.0.0 or 0 for syslog action
NSERR_AUDITLOG_PORT_SET_UNSET_NOTALLOWED	1476	0x5C4	Port cannot be set/ unset when LB vserver name/Server domain name is configured
NSERR_SYSLOG_TCPPROFILE_NOTALLOWED	1477	0x5C5	TCP profile name cannot be set when LB vserver name is configured
NSERR_SYSLOG_CANNOT_SET_LBVSNAME	1478	0x5C6	Unset the TCP profile name and then set the LB vserver name
NSERR_PORTALBINDING_NOT_SUPPORTED	1479	0x5C7	Operation not supported. Unset VPN uitheme parameter first and then try again.
NSERR_PORTALTHEME_NAME_NOT_ALLOWED	1480	0x5C8	\\EULA\\and \ \Custom\\keywords are reserved. Please

ERROR CODE	ERROR CODE(DECIMAL)	ERROR CODE(HEX)	ERROR MESSAGE
			try with any other name
NSERR_AUDIT_VERSION_MISMATCH	1481	0x5C9	Audit version mismatch
NSERR_RADIUS_LISTENER_PERM	1482	0x5CA	Operation not permitted on RADIUS LISTENER service
NSERR_SYSLOG_NETPROFILE_NOTALLOWED	1483	0x5CB	Network profile name cannot be set when LB vserver name is configured
NSERR_AUDITLOG_DMNRESOLVRETRY	1484	0x5CC	Domain name can be resolved only when server domain name is configured
NSERR_AUDITLOG_ARGUMENT_MISSING	1485	0x5CD	One of the arguments \\"Server IP/Server domain name/LB vserver name\\" is required for auditlog action
NSERR_AUDITLOG_DBS_SERVER_EXIST	1486	0x5CE	Auditlog internal DBS server with the same domain name exists
NSERR_ICAMODE_STATUS_NOT_OK	1487	0x5CF	vpn icamode status not okay
NSERR_SYSLOG_LBVSERVER_NOTALLOWED	1488	0x5D0	LB vserver name cannot be set when network profile is configured. Unset

ERROR CODE	ERROR CODE(DECIMAL)	ERROR CODE(HEX)	ERROR MESSAGE
			the netProfile and set LB vserver name
NSERR_INSTALL_FAILED	1490	0x5D2	Installation failed.
NSERR_OSPFAREA_UNSET	1491	0x5D3	Unset of OSPFArea is allowed only for VIP and SNIP
NSERR_PREQ_HOSTROUTE	1492	0x5D4	Pre-requisite Hostroute is disabled
NSERR_PREQ_NETWORKROUTE	1493	0x5D5	Pre-requisite Networkroute is disabled
NSERR_AUDITLOG_SVC_EXISTS	1496	0x5D8	Auditlog service exists with this server information
NSERR_AUDITSINFO_EXISTS_WITH_SAME_DOMAINNAME	1497	0x5D9	Auditlog action exists with the same domain name
NSERR_NO_PERM_ON_PLATFORM	1501	0x5DD	Operation not supported on this platform
NSERR_NOTDBS_SERVER_AUTOSCALE_SGRP	3779	0xEC3	Cannot configure autoScale as POLICY option for an IP based server.
NSERR_AUTO_SCALE_CLOUD_TYPE_ONLY	3810	0xEE2	Operation is supported for servicegroup with autoscale type CLOUD only.

ERROR CODE	ERROR CODE(DECIMAL)	ERROR CODE(HEX)	ERROR MESSAGE
NSERR_AUTO_DISABLE_GRACEFUL_NOT_SET	3811	0xEE3	Auto Disable Graceful not enabled. Set autoDisablegraceful parameter to YES.
NSERR_AUTO_DISABLE_GRACEFUL_NOT_SUPPORTED_ON_AZURE	3812	0xEE4	Auto Disable Graceful not supported on Azure Platform.
NSERR_MAX_MONITORS_BINDINGS_TO_SERVICE	3813	0xEE5	Maximum monitors bound to this entity exceeded
NSERR_INSUFFICIENT_IAM_PRIVILEGES	3815	0xEE7	Insufficient iam privileges configured for this feature
NSERR_LOGIN_NEXTFACTOR	3816	0xEE8	Nextfactor Login

CACONF ERRORS

CACONF ERRORS

ERROR CODE	ERROR CODE(DECIMAL)	ERROR CODE(HEX)	ERROR MESSAGE
NSERR_CACONF_CNFL_HEUREXP_RELEXP	480	0x1E0	Conflicting arguments, heurExpiryParam and relExpiry
NSERR_CACONF_CNFL_HEUREXP_RELEXPMILI	481	0x1E1	Conflicting arguments, heurExpiryParam and relExpiryMilliSec

ERROR CODE	ERROR CODE(DECIMAL)	ERROR CODE(HEX)	ERROR MESSAGE
NSERR_CACONF_CNFL_RELEXP_RELEXPMILI	482	0x1E2	Conflicting arguments, relExpiry and relExpiryMiliSec
NSERR_CACONF_CNFL_ABSEXP_HEUREXP	483	0x1E3	Conflicting arguments, absExpiry and heurExpiryParam
NSERR_CACONF_CNFL_ABSEXP_RELEXPMILI	484	0x1E4	Conflicting arguments, absExpiry and relExpiryMilliSec
NSERR_CACONF_CNFL_ABSEXPGMT_HERUEXP	485	0x1E5	Conflicting arguments, absExpiryGMT and heurExpiryParam
NSERR_CACONF_CNFL_ABSEXP_ABSEXPGMT	486	0x1E6	Conflicting arguments, absExpiry and absExpiryGMT
NSERR_CACONF_CNFL_ABSEXPGMT_RELEXPMILI	487	0x1E7	Conflicting arguments, absExpiryGMT and relExpiryMilliSec
NSERR_CACONF_CNFL_HITPARAMS_HITSLCTR	488	0x1E8	Conflicting arguments, hitParams and hitSelector
NSERR_CACONF_CNFL_INVALPARAMS_HITSLCTR	489	0x1E9	Conflicting arguments, invalParams and hitSelector

ERROR CODE	ERROR CODE(DECIMAL)	ERROR CODE(HEX)	ERROR MESSAGE
NSERR_CACONF_CNFL_HITPARAMS_INVLSLCTR	490	0x1EA	Conflicting arguments, hitParams and invalSelector
NSERR_CACONF_CNFL_INVALPARAMS_INVLSLCTR	491	0x1EB	Conflicting arguments, invalParams and invalSelector
NSERR_CACONF_CNFL_HITSLCTR_MATCHCOOKY	492	0x1EC	Conflicting arguments, hitSelector and matchCookies
NSERR_CACONF_CNFL_HITSLCTR_INVALREST2HOST	493	0x1ED	Conflicting arguments, hitSelector and invalRestrictedToHost
NSERR_CACONF_CNFL_HITSLCTR_IGNRPARAMVALCASE	494	0x1EE	Conflicting arguments, hitSelector and ignoreParamValueCase
NSERR_CACONF_CNFL_INVALSLCTR_MATCHCOOKY	495	0x1EF	Conflicting arguments, invalSelector and matchCookies
NSERR_CACONF_CNFL_INVALSLCTR_INVALREST2HOST	496	0x1F0	Conflicting arguments, invalSelector and invalRestrictedToHost
NSERR_CACONF_CNFL_INVALSLCTR	497	0x1F1	Conflicting arguments,

ERROR CODE	ERROR CODE(DECIMAL)	ERROR CODE(HEX)	ERROR MESSAGE
_IGNRPARAMVALCASE			invalSelector and ignoreParamValueCase
NSERR_CACONF_CNFL_POSRELEXP_RELEXP	498	0x1F2	Conflicting arguments, weekPosRelExpiry and relExpiry
NSERR_CACONF_CNFL_POSRELEXP_RELEXPMILI	499	0x1F3	Conflicting arguments, weekPosRelExpiry and relExpiryMiliSec
NSERR_CACONF_CNFL_ABSEXP_POSRELEXP	500	0x1F4	Conflicting arguments, absExpiry and weekPosRelExpiry
NSERR_CACONF_CNFL_ABSEXPGMT_POSRELEXP	501	0x1F5	Conflicting arguments, absExpiryGMT and weekPosRelExpiry
NSERR_CACONF_CNFL_IGNRPARAMVALCASE_HITPARAMS	502	0x1F6	Invalid arguments, ignoreParamValueCase is set while hitParams is 0
NSERR_CACONF_CNFL_HIT_INVALPARAMS_MATCHCUKY	503	0x1F7	Invalid arguments, matchCookies is set while both hitParams and invalParams are 0
NSERR_CACONF_CNFL_INVALREST2HOST_INVALPARAM	504	0x1F8	Invalid arguments, invalRestrictedToHost is set while invalParams is 0

ERROR CODE	ERROR CODE(DECIMAL)	ERROR CODE(HEX)	ERROR MESSAGE
NSERR_CACONF_CNFL_PREFETCH_PREFETCHSEC	505	0x1F9	Invalid arguments, prefetchPeriod is set while prefetch is disabled
NSERR_CACONF_CNFL_PREFETCH_PREFETCHMILI	506	0x1FA	Invalid arguments, prefetchPeriodMilliSec is set while prefetch is disabled
NSERR_CACONF_CNFL_PREFETCHMILI_PREFETCHSEC	507	0x1FB	Conflicting arguments, prefetchPeriodMilliSec and prefetchPeriod
NSERR_CACONF_ARG_LE_MIN_VAL	508	0x1FC	Invalid arguments, given value is less than the minimum value
NSERR_CACONF_ARG_GE_MAX_VAL	509	0x1FD	Invalid arguments, given value is more than the maximum value
NSERR_CACHEGROUP_NEGMINHIT	510	0x1FE	ContentGroup cannot have negative minHit.
NSERR_CACHEHA_OBJPERSIST_NOTSET	511	0x1FF	Cache Object Sync to Secondary is not set, use set cache parameter to enable.
NSERR_CACHEGROUP_INVALID_OPERATION	758	0x2f6	This operation is not permitted on selector based contentgroup

CENTRALMGMTSERVER ERRORS

CENTRALMGMTSERVER ERRORS

ERROR CODE	ERROR CODE(DECIMAL)	ERROR CODE(HEX)	ERROR MESSAGE
NSERR_CENTRALMGMTSERVER_REGFAILED	2047	0x7ff	Registration of device failed
NSERR_CENTRALMGMTSERVER_REQARGMISSING	2988	0xbac	Required argument missing

CFE ERRORS

CFE ERRORS

ERROR CODE	ERROR CODE(DECIMAL)	ERROR CODE(HEX)	ERROR MESSAGE
NSERR_NOSUCHIOCTL	1184	0x4a0	Command not implemented on server
NSERR_NOTARGETS	1185	0x4a1	No configured targets
NSERR_CANTRECOVER	1186	0x4a2	Configuration possibly inconsistent. Please check with the \\"show configstatus\\" command or reboot.
NSERR_IGNOREDIOCTL	1187	0x4a3	The command was ignored.
NSERR_REMOTECLOSE	1188	0x4a4	The remote side closed the connection.

ERROR CODE	ERROR CODE(DECIMAL)	ERROR CODE(HEX)	ERROR MESSAGE
NSERR_INVALID_TARGET	1189	0x4a5	The specified target does not exist
NSERR_FILE_ERROR	1190	0x4a6	File operation failed
NSERR_COMMENT_DROPPED	1191	0x4a7	Failed to retain all comments
NSERR_AGGREQ_TIMEOUT	1192	0x4a8	Request to Aggregator timed out
NSERR_AGGREAD	1193	0x4a9	Failed to read data from Aggregator
NSERR_RPC_CMD_DUP	1194	0x4aa	Found unexpected RPC duplicate command
NSERR_RPC_CMD_NONDUP	1195	0x4ab	Found unexpected RPC command
NSERR_CFE_PE_COMM	1196	0x4ac	Communication error with the packet engine
NSERR_AGG_CONFAIL	1197	0x4ad	Failed to connect to the aggregator
NSERR_CFE_PE_TIMOUT	1198	0x4ae	No response from the packet engine
NSERR_AGG_INVALIDRESPONSE	1199	0x4af	Invalid response from the aggregator
NSERR_NONTPSVR	1200	0x4b0	Cannot enable ntpd when there is no ntp server configured
NSERR_AGG_SENDFAIL	1201	0x4b1	Failed to send to aggregator

ERROR CODE	ERROR CODE(DECIMAL)	ERROR CODE(HEX)	ERROR MESSAGE
NSERR_CFE_ASLEARN_COMM	1202	0x4b2	Communication error with aslearn
NSERR_READONLY_SESSION	1203	0x4b3	Session is read-only; connect to the Cluster IP address to modify the configuration
NSERR_WRONG_PE_STATE	1204	0x4b4	The packet engine is in the wrong state to receive this command
NSERR_CFE_VALIDATION_FAILED	1205	0x4b5	Config validation failed
NSERR_XPATH_INTERNAL	1206	0x4b6	XPATH Internal Error
NSERR_CFE_CALLHOME_COMM	1207	0x4b7	Communication error with callhome
NSERR_BIND_NOENT	1208	0x4b8	No such bind resource
NSERR_PPNETMASK	1209	0x4b9	Netmask length 31 supports only point to point communications
NSERR_CFE_CLFSYNCD_COMM	1210	0X4ba	Communication error with cluster filesync server
NSERR_DB_INTERNAL	1211	0x4bb	Config DB Internal Error
NSERR_MIPDEPRECATED	1212	0x4bc	MIP is deprecated. Please use SNIP

ERROR CODE	ERROR CODE(DECIMAL)	ERROR CODE(HEX)	ERROR MESSAGE
NSERR_KEKEXISTS	1213	0x4bd	Cannot create KEK. KEK already exists.
NSERR_VAL_INCONSITENT	1214	0x4be	Validation inconsistent between PE and configd.
NSERR_CALLHOME_LOW_HB_CUSTOM_INTERVAL	1215	0x4bf	Setting a low heartbeat interval will result in frequent CallHome uploads
NSERR_INVALIDTIMEOUT	1219	0x4c3	Invalid Timeout, it cannot be beyond admin configured value and should be between in the range of 300-86400

CLI ERRORS

CLI ERRORS

ERROR CODE	ERROR CODE(DECIMAL)	ERROR CODE(HEX)	ERROR MESSAGE
NSERR_TCPCONNFAIL	1024	0x400	Connection failed
NSERR_LOGINFAIL	1025	0x401	Login failed
NSERR_NOLOGIN	1026	0x402	Not logged in
NSERR_AUTHTIMEOUT	1027	0x403	Not logged in or connection timed out
NSERR_NOT_PRIMARY	1028	0x404	You are connected to a secondary node; configuration

ERROR CODE	ERROR CODE(DECIMAL)	ERROR CODE(HEX)	ERROR MESSAGE
			changes made in this session will not be propagated to, or saved on, other nodes
NSERR_REMOTEOP	1029	0x405	Operation cannot be performed from remote login
NSERR_CONNLOST	1030	0x406	Connection with Citrix ADC lost
NSERR_RPCDATAMISMATCH	1031	0x407	Communication error (RPC data-size mismatch)
NSERR_RPCBADREPLY	1032	0x408	Communication error (bad RPC reply)
NSERR_UNABLETOPROMPT	1033	0x409	Unable to display secondary prompt
NSERR_LOGIN_CHALLENGED	1034	0X40A	Login challenged
NSERR_LOGOUT_SAVE	1035	0X40B	You are leaving without saving changes. Save changes now (Y/N)? [N]:
NSERR_CLEAR_CONFIG	1036	0X40C	Are you sure you want to clear the configuration(Y/N)? [N]:
NSERR_REBOOT	1037	0X40D	Are you sure you want to restart Citrix ADC (Y/N)? [N]:

ERROR CODE	ERROR CODE(DECIMAL)	ERROR CODE(HEX)	ERROR MESSAGE
NSERR_SHUTDOWNNS	1038	0X40E	Are you sure you want to completely stop Citrix ADC (Y/N)? [N]:
NSERR_REBOOT_SAVE	1039	0X40F	Configuration not Saved. Are you sure you want to restart Citrix ADC (Y/N)? [N]:
NSERR_USERABORT	1040	0x410	User requested abort
NSERR_EOF	1041	0x411	EOF
NSERR_INTERRUPT	1042	0x412	Interrupted
NSERR_INTERNAL	1043	0x413	Internal error
NSERR_NOT_CLIP	1044	0x414	You are connected to NSIP of a cluster node; configuration changes made in this session will not be propagated to, or saved on, other cluster nodes
NSERR_EULA_NOT ACCEPTED	1045	0x415	End User License Agreement not accepted. Login from CLI/GUI to accept it.
NSERR_FORCE_PWDCHANGE	1046	0x416	ForcePasswordChange is enabled. Please login from CLI/GUI to change

ERROR CODE	ERROR CODE(DECIMAL)	ERROR CODE(HEX)	ERROR MESSAGE
			the default nsroot password
NSERR_PWD_EXISTS	1047	0x417	Default nsroot password is not allowed. Please enter a new password.
NSERR_STRMAXLEN_255	1048	0x418	String length exceeds maximum. Allowed maximum length is 255 bytes.
NSERR_STRMAXLEN_32	1049	0x419	String length exceeds maximum. Allowed maximum length is 32.
NSERR_NOPREFIX_LENGTH	1050	0x41A	Prefix length must be given with network
NSERR_INVALIDPOLLVAL	1051	0x41B	Minpoll value should not be greater than maxpoll
NSERR_SHELL_ACCESS	1052	0x41C	User has shell access
NSERR_NORESPONSE	1056	0x420	No response from Citrix ADC
NSERR_IOERROR	1057	0x421	I/O error
NSERR_ENV	1058	0x422	Environment error
NSERR_CMDSFAILED	1059	0x423	Some commands failed
NSERR_ALLCMDSFAILED	1060	0x424	All commands failed

ERROR CODE	ERROR CODE(DECIMAL)	ERROR CODE(HEX)	ERROR MESSAGE
NSERR_INTMINVAL_CLT_IDLETIMEOUT	1061	0x425	Integer value below minimum for client idle timeout.
NSERR_INTMINVAL_SRV_IDLETIMEOUT	1062	0x426	Integer value below minimum for server idle timeout.
NSERR_INVALID_TCP_OPTION_TYPE	1063	0x427	Invalid tcp option type value. Allowed type values are 0 to 254
NSERR_OPNOTALLOW_ONCLIP	1065	0x429	Operation not permitted on CLIP
NSERR_LICEXPIRED	1066	0x42A	Features(s) license expired
NSERR_FEATDISABLED	1067	0x42B	Feature(s) not enabled
NSERR_LESS_PE	1068	0x42C	Operation not permitted. PE pool is not enough to enable additional management CPU
NSERR_MAXLIMIT	1072	0x430	Maximum resource limit reached
NSERR_SET_NOSUPPORT	1073	0x431	Set operation not supported
NSERR_INVALIDVALUE	1074	0x432	Invalid value
NSERR_INVALIDNAME	1075	0x433	Invalid name; names must begin with an alphanumeric character or

ERROR CODE	ERROR CODE(DECIMAL)	ERROR CODE(HEX)	ERROR MESSAGE
			underscore and must contain only alphanumerics, _, #, ., , :, @, =or -
NSERR_INVALIDPASSWORD	1076	0x434	Invalid password. Password must contain atleast one lower case character, one upper case character, one numeric character and one special character when strong password is enabled.
NSERR_INVALIDMINPASSWORDLENGTH	1077	0x435	Minimum password length should be greater than or equal to 4 when strong password is enabled
NSERR_ENABLINGPASSWORDCOMPLEXITY	1078	0x436	Strong Password now enabled. Please ensure all the existing user passwords adhere to this restriction. Minimum Password Length is set to 4 as default.
NSERR_INVALIDPASSWORDLENGTH	1079	0x437	Password length should adhere to minimum password length value in system parameter settings.

ERROR CODE	ERROR CODE(DECIMAL)	ERROR CODE(HEX)	ERROR MESSAGE
NSERR_CANNOTSETMINPASSWORDLENGTH	1080	0x438	Strong Password needs to be enabled to set the Minimum Password Length value.
NSERR_NOSUCHCMD	1088	0x440	No such command
NSERR_CMDAMBIGUOUS	1089	0x441	Ambiguous command name
NSERR_NOSUCHARG	1090	0x442	No such argument
NSERR_ARGVALMISSING	1091	0x443	Required argument value missing
NSERR_ARGSMUTEX	1092	0x444	Arguments cannot both be specified
NSERR_ARGPREREQ	1093	0x445	Argument pre-requisite missing
NSERR_ARGSTOOFEW	1094	0x446	Too few arguments
NSERR_ARGMISSING	1095	0x447	Required argument missing
NSERR_ARGORDER	1096	0x448	Argument(s) out of order
NSERR_ARGVALBAD	1097	0x449	Invalid argument value
NSERR_ARGVALSEQ	1098	0x44A	Arguments cannot have the same value
NSERR_ARGAMBIGUOUS	1099	0x44B	Ambiguous argument name

ERROR CODE	ERROR CODE(DECIMAL)	ERROR CODE(HEX)	ERROR MESSAGE
NSERR_SYNCGSLBCONFIG	1100	0x44C	Please confirm whether you want to sync-config (Y/N)? [N]:
NSERR_SYNCGSLBCONFIG_WARN	1101	0x44D	Configuration mismatch with the slave sites will result in the master sites configuration prevailing on the slave sites. This might result in disruptions in traffic.
NSERR_FORCESYNCGSLBCONFIG_WARN	1134	0x46e	GSLB configuration will be removed from the slave sites and the master sites configuration will be applied to the slave sites. Traffic disruptions might occur during forced synchronization.
NSERR_ARGVALSNEQ	1104	0x450	Arguments must have the same value
NSERR_ARGVALAMBIGUOUS	1105	0x451	Ambiguous argument value
NSERR_STRMAXLEN	1106	0x452	String length exceeds maximum
NSERR_STRMINLEN	1107	0x453	String too short
NSERR_INTMAXVAL	1108	0x454	Integer value exceeds maximum

ERROR CODE	ERROR CODE(DECIMAL)	ERROR CODE(HEX)	ERROR MESSAGE
NSERR_INTMINVAL	1109	0x455	Integer value below minimum
NSERR_INVALIDIP	1110	0x456	Invalid IP address
NSERR_INVALIDNETMASK	1111	0x457	Invalid netmask
NSERR_TOOMANYVALS	1112	0x458	Argument has too many values
NSERR_BADRANGE	1113	0x459	Range minimum exceeds range maximum
NSERR_EXPRQUOTES	1114	0x45A	Expression must be enclosed in quotes
NSERR_BADQUOTES	1115	0x45B	Unmatched quote
NSERR_INVALIDRANGE	1116	0x45C	Invalid range specification
NSERR_MISMATCHRANGES	1117	0x45D	Ranges are different sizes
NSERR_MULTIRANGES	1118	0x45E	Only one range allowed per argument
NSERR_NOMATCHCHAR	1119	0x45F	Unmatched character
NSERR_NEEDREBOOT	1120	0x460	The configuration must be saved and the system rebooted for these settings to take effect

ERROR CODE	ERROR CODE(DECIMAL)	ERROR CODE(HEX)	ERROR MESSAGE
NSQ_SAVECONFIG	1121	0x461	Do you want to save the new configuration?
NSERR_NOTSAVED	1122	0x462	The configuration changes were not saved - they will be lost on the next reboot
NSERR_CMDEXEC	1123	0x463	Error(s) occurred - some or all changes may not be applied
NSQ_REBOOT	1124	0x464	Do you want to reboot the system now?
NSERR_NOREBOOT	1125	0x465	The configuration changes will not take effect until the system is rebooted
NSERR_PGMFAILED	1126	0x466	External program failed
NSERR_CTXMODE	1127	0x467	Contextual CLI mode
NSERR_CMDINCOMPLETE	1128	0x468	Incomplete command
NSERR_CMDOUTOFCTX	1129	0x469	Command not valid here
NSERR_CONFIGSAVED	1130	0x46A	The running configuration has been saved

ERROR CODE	ERROR CODE(DECIMAL)	ERROR CODE(HEX)	ERROR MESSAGE
NSERR_CONFIGCLEARED	1131	0x46B	The running configuration has been cleared
NSERR_REBOOTING	1132	0x46C	Citrix ADC is rebooting now
NSERR_NOCONFIGSAVE	1133	0x46D	The running configuration has not changed
NSERR_REGEXNOANCHOR	1135	0x46F	First character of regular expression must be \^
NSERR_REGEXNOMATCH	1136	0x470	Regular expression does not match
NSERR_REGEXINVALID	1137	0x471	Invalid regular expression
NSERR_REGEXNOTALLOWED	1138	0x472	Regular expression not allowed
NSERR_REGEXNOCMD	1139	0x473	No matches
NSERR_INVALIDIPV6_FORMAT	1140	0x474	Invalid IPv6 address
NSERR_INVALIDIPV6_TWO_DOUBECOLON	1141	0x475	Double-colon can appear only once in an IPv6 address
NSERR_INVALIDIPV6_NOPREFIX_LENGTH	1142	0x476	Prefix length must be given with an IPv6 address
NSERR_INVALIDIPV6_PREFIX_VALUE	1143	0x477	Prefix length must be in the range of 0-128

ERROR CODE	ERROR CODE(DECIMAL)	ERROR CODE(HEX)	ERROR MESSAGE
NSERR_TERMNAMEINVALID	1144	0x478	Invalid term name
NSERR_TERMINVALID	1145	0x479	Invalid syntax in term
NSERR_FORCEFAILOVER	1146	0x47a	Please confirm whether you want force-failover (Y/N)? [N]:
NSERR_FORCEFAIL_HEALTH_WARN	1147	0x47b	Force Failover may cause configuration loss, peer health not optimum. Reason(s):
NSERR_HELLOTIME_MULTIPLE	1148	0x47c	Invalid value. Hellotime Interval must be a multiple of 200
NSERR_FORCESYNCSAVE	1149	0x47d	Do you want to save the config after sync (Y/N)? [Y]:
NSERR_ERROUTFILENAME	1150	0x47e	Character /not allowed in outfilename
NSERR_RNATIPDEL	1151	0x47f	address is bound to rnat config, can not be removed
NSERR_INVALIDALIAS	1152	0x480	Invalid alias
NSERR_NOSUCHALIAS	1153	0x481	No such alias

ERROR CODE	ERROR CODE(DECIMAL)	ERROR CODE(HEX)	ERROR MESSAGE
NSERR_ALIASEXISTS	1154	0x482	Alias already exists
NSERR_NOSUCHFILE	1155	0x483	No such file
NSERR_NOTREGFILE	1156	0x484	Not a file
NSERR_DEPRCMD	1157	0x485	Command deprecated
NSERR_DEPRARG	1158	0x486	Argument deprecated
NSERR_NOTLOGFILE	1159	0x487	Not a Citrix ADC log file
NSERR_NOPLENFORIPV6RANGE	1160	0x488	Prefix length cannot be specified for an IPv6 range
NSERR_INVALIDINT	1161	0x489	Integer not in range
NSERR_CMDAMBIGUOUS_USECOMPLETIONSOPTIONS	1162	0x48A	Ambiguous (use cmd completion for options)
NSERR_METADATA_INVAL_ENTITYTYPE	1163	0x48B	Invalid Entity Type
NSERR_SETNOTEXIST	1164	0x48C	set operation does not exist
NSERR_SETARGNOTEXIST	1165	0x48d	set operation for one of the modified argument does not exist

ERROR CODE	ERROR CODE(DECIMAL)	ERROR CODE(HEX)	ERROR MESSAGE
NSERR_INVALID_DS	1166	0x48e	Invalid datasource
NSERR_NOSUCHCOUNTER	1167	0x48F	No such counter
NSERR_INVALIDIPMASK	1168	0x490	Invalid IP address mask
NSERR_INVALIDIPPAT	1169	0x491	Invalid IP address pattern
NSERR_BADRC	1170	0x492	Unexpected return code
NSERR_INVALIDRANGETYPE	1171	0x493	Inappropriate range values for value type
NSERR_INVALIDRANGEVAL	1172	0x494	Invalid range value
NSERR_INVALIDIPV6_PREFIX_LENGTH	1173	0x495	Prefix length must not be given
NSERR_ENTITYDELETE_FAIL	1174	0x496	Can not delete active entity
NSERR_DONE	1175	0X497	This is the last result
NSERR_RNATIPDEL2	1176	0x498	An RNAT address requires an IP address to exist
NSERR_INCOMPATIBLEIP	1181	0x49d	Private IP and public IP must be either IPv4 or IPv6
NSERR_TRANSCR_IP	1182	0x49e	For transparent CR vserver, you cannot specify IP address

ERROR CODE	ERROR CODE(DECIMAL)	ERROR CODE(HEX)	ERROR MESSAGE
NSERR_PASSWORD_MISMATCH	1183	0x49f	Passwords entered do not match
NSERR_LEADING_TRAILING_SPACES	1053	0x41d	Leading or Trailing spaces in string
NSERR_SAVENOTDONE	1054	0x41e	The running configuration is not saved
NSERR_INVALIDIP_VSVRTYPE	1178	0x49a	Invalid IP address or Vserver Type
NSERR_HELLOTIME_DEADINTERVAL_COMBINATION	1177	0x499	The Hello Interval must be set to 200 msec when dead interval is less than 3 sec
NSERR_MAXSESSION	1179	0x49b	Maxsession cannot be set for this user
NSERR_NEEDREPLICATE	1180	0x49c	This command will not be propagated to the other cluster nodes as you are connected to the NSIP of this cluster node
NSERR_HDOSP_DEPRECATED	3917	0xF4D	HTTP DOS protection feature has been deprecated in favour of AppQoE
NSERR_PQ_DEPRECATED	3871	0xF1F	Priority Queuing feature has been deprecated in favour of AppQoE

ERROR CODE	ERROR CODE(DECIMAL)	ERROR CODE(HEX)	ERROR MESSAGE
NSERR_IOH_DEPRECATED	3870	0xF1E	Sure Connect feature has been deprecated in favour of AppQoE
NSERR_NOT_PRIVATE_EXPORT	1069	0x42D	Private entities can not be exported
NSERR_OBJECT_IN_USE	1070	0x42E	Object in use, cannot delete
NSERR_CERBUNDLE_NOT_FOUND	1071	0x42F	Certbundle name not found in mapping file
NSERR_CERTBUNDLE_CANNOT_OPEN_PID	1086	0x43E	Cannot open nscertforgerd pid file
NSERR_FAILED_TO_APPLY	1087	0x43F	Failed to apply new cert bundle
NSERR_AWS_UNSUP	1103	0x44F	Command not supported on AWS VPX
NSERR_POOLED_VPX	1102	0x44E	Ensure to provision the VPX with the recommended settings for this Bandwidth.
NSERR_GCP_UNSUPPORTED_MTU	2439	0x987	MTU greater than 1460 bytes is not supported on this platform.
NSERR_AWS_UNSUPPORTED_MTU	2440	0x988	MTU greater than 9000 bytes is not supported on this platform.

ERROR CODE	ERROR CODE(DECIMAL)	ERROR CODE(HEX)	ERROR MESSAGE
NSERR_INVALID_QUERYTYPE_FOR_IPV6	2441	0x989	Cannot use ipv6Address option with querytype set to A

CLOUDPARAM ERRORS

CLOUDPARAM ERRORS

ERROR CODE	ERROR CODE(DECIMAL)	ERROR CODE(HEX)	ERROR MESSAGE
NSERR_UUID_FORMAT	3563	0xdeb	Invalid UUID format
NSERR_CUSTID_FORMAT	3564	0xdec	Invalid Citrix Cloud Customer ID format
NSERR_FQDN_FORMAT	3565	0xded	Invalid FQDN format

CLUSTER ERRORS

CLUSTER ERRORS

ERROR CODE	ERROR CODE(DECIMAL)	ERROR CODE(HEX)	ERROR MESSAGE
NSERR_CLNOTEXIST	2448	0x990	Cluster does not exist
NSERR_TOOMANYCLS	2449	0x991	This node is already added to another cluster

ERROR CODE	ERROR CODE(DECIMAL)	ERROR CODE(HEX)	ERROR MESSAGE
NSERR_RMLOCALNODE	2450	0x992	Local Node cannot be deleted
NSERR_BKPLANEENABLED	2451	0x993	Cluster Backplane is enabled on this interface
NSERR_LOCALNODE_NOTCONFIGURED	2452	0x994	Clustering cannot be enabled as the local node is not a member of the cluster.
NSERR_HACL_CANNOT_COEXIST	2453	0x995	Clustering and HA cannot coexist.
NSERR_CLACPKEYNOTCRT	2454	0x996	LACP key for Node/ cluster is not correct
NSERR_CLNODE_EXIST	2455	0x997	already in cluster
NSERR_BKPLANEDISABLED	2456	0x998	Disabling Cluster backplane interface
NSERR_CLAGTHROUGHPUT	2457	0x999	Currently throughput settings are not allowed on CLAG
NSERR_LICMISMATCH	2458	0x99A	License mismatch between Configuration Coordinator and local node
NSERR_CLIP_INVALIDNETMASK	2459	0x99B	CLIP netmask should be 255.255.255.255.

ERROR CODE	ERROR CODE(DECIMAL)	ERROR CODE(HEX)	ERROR MESSAGE
NSERR_BKPLANELACPDELETE	2460	0x99C	This is the last interface in LACP channel configured as backplane
NSERR_LINKSETMEMBER	2461	0x99D	Interface/Channel is Linkset member.
NSERR_CONFIG_PARAM	2462	0x99E	Setting up of Citrix ADC parameter ignored using this command in HA/ Cluster mode. Use set nsparamcommand for this purpose.
NSERR_CLIP_INVALIDPREFIXLEN	2463	0x99F	CLIP prefix length should be 128.
NSERR_CLUSTER_L2NOTSUPP	2464	0x9A0	L2 Mode not supported in Cluster mode.
NSERR_LINKSET_ONEPERNODE	2465	0x9A1	Linkset can have only one Interface/ Channel per Node
NSERR_LINKSET_NOTBOUND	2466	0x9A2	Interface/Channel not bound to Linkset
NSERR_LINKSET_IDINVALID	2467	0x9A3	Linkset name should be of form LS/
NSERR_CLUSTER_NBNOTSUPP	2468	0x9A4	CloudBridge is not supported in Cluster mode.

ERROR CODE	ERROR CODE(DECIMAL)	ERROR CODE(HEX)	ERROR MESSAGE
NSERR_CLUSTER_VPNNOTSUPP	2469	0x9A5	SSLVPN is not supported in Cluster mode.
NSERR_CLUSTER_AAANOTSUPP	2470	0x9A6	AAA is not supported in Cluster mode.
NSERR_CLUSTER_IPSECNOTSUPP	2471	0x9A7	IPSEC is not supported in Cluster mode.
NSERR_LINKSET_LAMEMBER	2472	0x9A8	Interface part of a Channel
NSERR_CLIP_LOCALNODE_NOTCONFIGURED	2473	0x9A9	CLIP cannot be added as the local node is not a member of the cluster.
NSERR_CLVERSIONMISMATCH	2474	0x9AA	Cluster version mismatch between Configuration Coordinator and local node
NSERR_CLUSTER_CMNOTSUPP	2475	0x9AB	Connection Mirroring is not supported in Cluster mode
NSERR_ARP_CLAG_OWNERNODE_MANDATORY	2476	0x9AC	Owner Node should be specified for CLAG.
NSERR_ARP_OWNERNODE_MANDATORY	2477	0x9AD	Owner Node should be specified.
NSERR_NOOPCCO	2478	0x9AE	Operation not permitted on the

ERROR CODE	ERROR CODE(DECIMAL)	ERROR CODE(HEX)	ERROR MESSAGE
			Configuration Coordinator.
NSERR_SPOTTED_CONFIG_PRESENT	2479	0x9AF	Operation not permitted as the node is part of spotted entity configuration.
NSERR_NOT_CLUSTER_IP	2480	0x9B0	IP address should be a CLIP address.
NSERR_CLUSTER_NNRES	2481	0x9B1	Error in communication with other nodes in the cluster
NSERR_LAST_CLUSTER_IP_REMOVAL	2482	0x9B2	A Citrix ADC cluster must have a minimum of one cluster IP address. This is the last available cluster IP address and hence it cannot be deleted.
NSERR_CONFIG_PARAM_INVALID	2483	0x9B3	Configured Param(s) too small for Cluster. Parameter value(s) will be bumped to OVS node count.
NSERR_CLUSTER_INVALIDNODE	2484	0x9B4	Owner node value should be between 0 and 31, both inclusive.
NSERR_LRONCLACPONLY	2485	0x9B5	Link Redundancy can be set only in

ERROR CODE	ERROR CODE(DECIMAL)	ERROR CODE(HEX)	ERROR MESSAGE
			Cluster LACP Channels
NSERR_MULTIBIND_NOT_SUPPINCLUSTER	2486	0x9B6	Multiple Bindings in Single Bind Command not Supported in Cluster.
NSERR_CLUSTER_FISNAMES_DUPNOTALLOWED	2487	0x9B7	In a cluster setup, the FIS name on each node must be unique.
NSERR_CLUSTER_CLAG_NOLOCALINTF	2488	0x9B8	CLA cannot be bound to the FIS as the CLA does not contain a local interface.
NSERR_FIS_MAX_LIMIT	2489	0x9B9	An appliance can have a maximum of 64 failover interface sets (FIS). This appliance or ownernode (cluster) already has maximum number of allowable FIS.
NSERR_BOUND_TO_FIS	2490	0x9BA	Interface bound to FIS.
NSERR_CHANNEL_OPEN_FAILED	2491	0x9BB	Unable to open channel
NSERR_NOT_BOUND_TO_FIS	2492	0x9BC	Interface not bound to this FIS
NSERR_CLUSTER_INC_NOTSUPP	2493	0x9BD	Feature not supported when

ERROR CODE	ERROR CODE(DECIMAL)	ERROR CODE(HEX)	ERROR MESSAGE
			cluster is configured in INC mode
NSERR_CONFIG_NG_CHANGED	4095	0xfff	Config NodeGroup changed, force cluster sync should be fired on newly added node to be in sync
NSERR_STRIPED_IP_EXIST	2494	0x9BE	Striped SNIPs/MIPs must be removed when transitioning to cluster-inc
NSERR_WRONG_NG	2495	0x9BF	All cluster nodes should be bound to DEFAULT_NG before transitioning to cluster-inc
NSERR_NG_REQUIRED	2496	0x9C0	The nodegroup parameter is mandatory when the -inc parameter is enabled
NSERR_NG_CHANGE	2497	0x9C1	NodeGroup can be changed only for transition from cluster to cluster-inc
NSERR_NG_INVALID	2498	0x9C2	Invalid nodegroup
NSERR_NOT_LAST_NODE	2499	0x9C3	The -flushall parameter can be enabled only when there is one node in the nodegroup

ERROR CODE	ERROR CODE(DECIMAL)	ERROR CODE(HEX)	ERROR MESSAGE
NSERR_MULTI_NG_BINDING_EXIST	2500	0x9C4	More than one Nodegroup has nodes bound to it
NSERR_INVALID_NC_CONFIG	2501	0x9C5	The owner group parameter is applicable only when the INC parameter is enabled on the cluster instance
NSERR_FIPSCL_CANNOT_COEXIST	2502	0x9C6	Clustering and FIPs configuration cannot coexist.
NSERR_LS_CLAG_INTF_SAME_NG	2503	0x9C7	All interfaces of LinkSet/CLAG should belong to same nodegroup
NSERR_NODE_BINDING_INC	2504	0x9C8	Node cannot be bound to Default nodeGroup on cluster-inc setup
NSERR_IP_OWNERDOWN_PARAM	2522	0x9DA	OwnerDownResponse allowed only for spotted SNIP on cluster
NSERR_INVALID_LS	2411	0x96b	Invalid linkset
NSERR_IP_ALREADY_OWNED	2541	0x9ed	Cannot add ip address as the address is already owned by cluster or peer node

ERROR CODE	ERROR CODE(DECIMAL)	ERROR CODE(HEX)	ERROR MESSAGE
NSERR_LS_UNABLE_GET_VLAN	2542	0x9ee	Unable to get the current vlan of this linkset.
NSERR_LS_NOT_BOUND_VLAN	2543	0x9ef	Linkset not a member of this vlan
NSERR_CL_UNSUPPORTED_PROTO	2544	0x9f0	Retain connections on joining a new node is not supported for this protocol
NSERR_CL_UNSUPPORTED_IN_NCORE	2545	0x9f1	This feature/option is not supported in n-core
NSERR_NO_SETTING_PROP	2546	0x9f2	This setting will not be propagated to other nodes.
NSERR_OWNERNODE_ALLOWED_ON_CLU	2547	0x9f3	ownerNode setting is allowed only on CLIP.
NSERR_CL_NOT_DFLT_NG	2548	0x9f4	In L2 cluster, adding a node with nodegroup other than DEFAULT_NG is not permitted.

CR ERRORS

CR ERRORS

ERROR CODE	ERROR CODE(DECIMAL)	ERROR CODE(HEX)	ERROR MESSAGE
NSERR_PXY_CACHE_HMG	512	0x200	All caches in a content group should be of the same type
NSERR_PXY_RM_LAST_MEMT	513	0x201	Cannot remove last map entry from the table; remove the map table instead
NSERR_CSW_INS_INVAL_PFX	514	0x202	Invalid prefix specified in the URL
NSERR_PENG_EXPR_IVAL_NAME	515	0x203	Invalid expression name
NSERR_CSW_BIG_URL	516	0x204	URL specified is too long - maximum is 208, including . and *
NSERR_CSW_BIG_PFX	517	0x205	Prefix specified is too long - maximum is 199, excluding *
NSERR_CSW_BIG_SFX	518	0x206	Extension specified is too long, maximum is 8
NSERR_CSW_INVAL_SFX	519	0x207	Extension is invalid
NSERR_EXPR_NOMETHOD	520	0x208	Invalid method name
NSERR_EXPR_NOURLTOKENS	521	0x209	Invalid HTTP URL tokens
NSERR_EXPR_NOVERSION	522	0x20A	Invalid HTTP version

ERROR CODE	ERROR CODE(DECIMAL)	ERROR CODE(HEX)	ERROR MESSAGE
NSERR_EXPR_NOHDR	523	0x20B	Invalid HTTP header
NSERR_EXPR_NOCACNTL	524	0x20C	Invalid cache-control value
NSERR_EXPR_NOPRAG	525	0x20D	Invalid pragma value
NSERR_EXPR_NOQUERY	526	0x20E	Invalid query string
NSERR_EXPR_NOQUAL	527	0x20F	Invalid qualifier
NSERR_ACTION_INUSE	528	0x210	Action name is already in use
NSERR_ACTION_HDR_INVAL	529	0x211	Invalid header
NSERR_URLQ_INVAL	530	0x212	?
NSERR_UNDEF_ACTION	531	0x213	Action directive or qualifier is not valid
NSERR_CPE_INUSE	532	0x214	Policy name is already in use
NSERR_CPE_REQACT_INVAL	533	0x215	Request action is not valid
NSERR_CPE_RSPACT_INVAL	534	0x216	Response action is not valid
NSERR_CPE_REQRULE_INVAL	535	0x217	Request rule is not valid
NSERR_CPE_RSPRULE_INVAL	536	0x218	Response rule is not valid

ERROR CODE	ERROR CODE(DECIMAL)	ERROR CODE(HEX)	ERROR MESSAGE
NSERR_ACTION_DEFINVAL	537	0x219	Default actions cannot be removed
NSERR_ACTION_NOTPRESENT	538	0x21A	Action does not exist
NSERR_PXY_INVAL_SERVICETYPE	539	0x21B	Invalid service type for virtual server
NSERR_CACHEPOLICY_INUSE	540	0x21C	Caching policy name is already in use
NSERR_CACHEGROUP_INTERNAL	543	0x21F	Built-in content groups cannot be removed
NSERR_CPE_INVAL	544	0x220	Policy name is invalid
NSERR_EXPR_NOLEN	545	0x221	URL length is not valid
NSERR_DNSWAIT	546	0x222	?
NSERR_GW_TIMEOUT	547	0x223	?
NSERR_CSWDMN_INUSE	548	0x224	Domain is already hosted by another server
NSERR_CSWDMN_PLCY_EXIST	549	0x225	?
NSERR_ACTION_NOMOD_HDR	550	0x226	Action has missing argument
NSERR_EXPR_INVAL_OPERATOR	551	0x227	Invalid operator for the qualifier

ERROR CODE	ERROR CODE(DECIMAL)	ERROR CODE(HEX)	ERROR MESSAGE
NSERR_EXPR_DEF_REM_INVAL	552	0x228	Default expression cannot be removed
NSERR_EXPR_TOOMANY	553	0x229	Expression limit reached
NSERR_ACTION_TOOMANY	554	0x22A	Action limit reached
NSERR_CSWPOLICY_TOOMANY	555	0x22B	CS policy limit reached
NSERR_CRDPOLICY_TOOMANY	556	0x22C	CR policy limit reached
NSERR_MAPPOLICY_TOOMANY	557	0x22D	Map policy limit reached
NSERR_FILTPOLICY_TOOMANY	558	0x22E	Filter policy limit reached
NSERR_CACHEPOLICY_TOOMANY	559	0x22F	Integrated cache policy limit reached
NSERR_CACHEGROUP_TOOMANY	560	0x230	Integrated cache content group limit reached
NSERR_CACHEPARAM_MEMALLOC_FAILED	561	0x231	Unable to allocate specified amount of memory
NSERR_CACHEGROUP_INUSE	562	0x232	Content group name is already in use
NSERR_CACHEGROUP_EXPCONFLICT	563	0x233	Expiry times conflict
NSERR_CACHEPARAM_INVAL	564	0x234	Cache parameter is invalid

ERROR CODE	ERROR CODE(DECIMAL)	ERROR CODE(HEX)	ERROR MESSAGE
NSERR_CACHEGROUP_PARAM_INVAL	565	0x235	Invalid invalidation parameter
NSERR_CACHEGROUP_QUERY_INVAL	566	0x236	Invalid query string
NSERR_ACTION_INVAL	567	0x237	Action name is invalid
NSERR_EXPR_DEF_SET_INVAL	568	0x238	Default expression cannot be set
NSERR_CACHEGROUP_RESSZ_MIN_GT_MAX	569	0x239	Minimum response size cannot exceed the maximum response size
NSERR_FILTACION_INVALRESPCODE	570	0x23A	Invalid HTTP response code
NSERR_SVPN_CERT_CHALLENGE	571	0x23B	Destination Server challenged for a client certificate
NSERR_CACHEGROUP_HOST_REQ	576	0x240	Host required for a group with invalidation restricted to host
NSERR_CACHEGROUP_HOST_NREQ	577	0x241	Host not required
NSERR_CACHEGROUP_DYNGRP_NEXP	578	0x242	Cannot expire a parameterized content group

ERROR CODE	ERROR CODE(DECIMAL)	ERROR CODE(HEX)	ERROR MESSAGE
NSERR_CACHEGROUP_ONE_GRP_REQ	579	0x243	Only one content group required
NSERR_CACHEGROUP_ONE_ALL_REQ	580	0x244	ALL should be the only content group specified
NSERR_CACHEGROUP_MATCH_PARAM_INVAL	581	0x245	Invalid request hit parameter
NSERR_CACHEGROUP_DYN_RES_CACHE	582	0x246	Cannot specify a parameterized group with a response time CACHE action
NSERR_EXPR_INVAL_VALUE	583	0x247	Invalid expression value
NSERR_CACHEGROUP_PREFETCH_CONFLICT	584	0x248	Prefetch periods conflict
NSERR_CACHEGROUP_PREFETCH_ENABLE	585	0x249	Enable prefetch to set prefetch period
NSERR_CACHEGROUP_CCH_UNKNOWN	586	0x24A	Unknown cache-control header
NSERR_CACHEGROUP_RELEXP_X10MS	587	0x24B	Relative expiry milliseconds must be multiples of 10
NSERR_CACHEGROUP_PREFETCH_X10MS	588	0x24C	Prefetch period milliseconds must be multiples of 10

ERROR CODE	ERROR CODE(DECIMAL)	ERROR CODE(HEX)	ERROR MESSAGE
NSERR_CACHEGROUP_STATIC_TO_DYNAMIC	589	0x24D	This content group cannot be made parameterized
NSERR_CLASSIC_POLICY_DEPRECATED	590	0x24E	Classic policy expressions are deprecated - use Default (Advanced) policy expressions instead; the nspepi utility may be helpful in conversion
NSERR_PLCY_DEF_REM_INVAL	591	0x24F	Default policy cannot be removed
NSERR_CACHEGROUP_PREFETCH_REL_NREQ	592	0x250	Relative expiry and prefetch period should both be seconds or milliseconds
NSERR_EXPR_SET_INVAL_FLOWTYPE	593	0x251	Flow type cannot be changed
NSERR_EXPR_TOO_BIG	594	0x252	Expression too long - maximum length is 1500, in fully-qualified form
NSERR_CPE_INVALID_IDRANGE	595	0x253	Priority should be between 0 and 64000 (inclusive)
NSERR_EXPR_TOO_BIG_EXT	596	0x254	Expression too long - maximum length is 800, in fully-qualified form

ERROR CODE	ERROR CODE(DECIMAL)	ERROR CODE(HEX)	ERROR MESSAGE
NSERR_CACHEGROUP_MATCHCOOKIE_DYN_REQ	597	0x255	Match cookies argument allowed only with parameterized groups
NSERR_CACHEGROUP_MATCHCOOKIE_NREQ	598	0x256	Cannot modify match cookies
NSERR_CACHEFWPXY_PRESENT	599	0x257	Forward proxy is already present
NSERR_CACHE_PREFETCH_REEVAL_NREQ	600	0x258	Cannot configure a content group to both prefetch and evaluate every miss
NSERR_CACHEFWPXY_TOOMANY	601	0x259	Forward proxy limit reached
NSERR_CACHE_PET_FC_NREQ	602	0x25A	Cannot enable both Poll Every Timeand Flash Cache
NSERR_SAVECONFIG_IN_PROGRESS	603	0x25B	Configuration is being saved, please try again later
NSERR_GWSUBNET_NOT_EXIST	604	0x25C	The gateway is not directly reachable
NSERR_GW_REQ_SUBNET	605	0x25D	An existing route relies on the presence of this subnet
NSERR_URLPOL_NO_PRI	606	0x25E	Priority cannot be specified for URL-

ERROR CODE	ERROR CODE(DECIMAL)	ERROR CODE(HEX)	ERROR MESSAGE
			based content switching policy
NSERR_BAD_CR_ATTRIBS	607	0x25F	Bad cache redirection VIP attributes given
NSERR_MAC_NOT_SUPPORTED	608	0x260	MAC redirection not supported
NSERR_POLICY_NOT_SUPPORTED	609	0x261	Policy redirection not supported with NNTP
NSERR_CACHEABILITY_NOT_SUPPORTED	610	0x262	Cannot set cacheability on cache
NSERR_HOST_RT_NOT_ALLOWED	611	0x263	Host route advertisement not permitted for this IP
NSERR_ROUTING_NOT_ALLOWED	612	0x264	Routing protocols can run only on SNIPs or NSIP
NSERR_CONFIG_NOTSAVED	613	0x265	Configuration is not saved
NSERR_DIS_ROUTING_NSIP	614	0x266	Cannot disable dynamic routing on NSIP
NSERR_NETWORK_RT_NOT_ALLOWED	615	0x267	Network route advertisement not permitted on this IP
NSERR_NR_ENABLE	616	0x268	networkRoute can be enabled only for one IP per subnet

ERROR CODE	ERROR CODE(DECIMAL)	ERROR CODE(HEX)	ERROR MESSAGE
NSERR_EXPRMISMATCH	624	0x270	?
NSERR_NO_HOST	627	0x273	No host header and default not set
NSERR_NO_DFLT	628	0x274	Host header present, no match, and default not set
NSERR_DFLTDMN_FIRST	629	0x275	No default domain translation entry
NSERR_PXY_CONF_LOOP	630	0x276	Source and target domains cannot be the same
NSERR_PXY_ME_DUP	632	0x278	Duplicate map entry
NSERR_PXY_IVAL_TGT	633	0x279	Target is invalid during map entry addition
NSERR_CONNECTED	634	0x27A	Proxy connection established
NSERR_AUTHENTICATE	635	0x27B	Authentication required
NSERR_LARGE_DOMAIN	636	0x27C	Domain length overflow
NSERR_PXY_IVAL_URL	640	0x280	Source/destination URL or the combination is not valid
NSERR_PXY_ME_USE	641	0x281	Map entry is already in use

ERROR CODE	ERROR CODE(DECIMAL)	ERROR CODE(HEX)	ERROR MESSAGE
NSERR_PXY_MT_TYPE	642	0x282	Map entries in a table should be homogeneous
NSERR_PXY_MT_USE	643	0x283	Map table is in use
NSERR_PXY_MB_INVAL	644	0x284	Invalid map bind operation
NSERR_PXY_MB_USE	645	0x285	You can bind only one table to a vserver
NSERR_PXY_SETDCDN	646	0x286	Invalid DCDN configuration
NSERR_PXY_SETDFLT	647	0x287	Invalid default domain specification
NSERR_PXY_DFLT_NOTSET	648	0x288	Removing default without setting it
NSERR_PXY_FWD_IVAL	649	0x289	Invalid FWD PXY map entry
NSERR_PXY_MTX_PROT	656	0x290	PXY and cache protocol should be the same
NSERR_PXY_MTX_TRA	657	0x291	Invalid cache type on service
NSERR_PXY_MTX_FWD	658	0x292	Forward Pxy matrix
NSERR_PXY_MTX_REV	659	0x293	Reverse Pxy Matrix

ERROR CODE	ERROR CODE(DECIMAL)	ERROR CODE(HEX)	ERROR MESSAGE
NSERR_PXY_ADD_TRA_NONHTTP	660	0x294	transparent non-HTTP policy is invalid
NSERR_PXY_ADD_TRA_OTHER	661	0x295	transparent non-HTTP map/via should not be specified
NSERR_PXY_OPT_INVAL	662	0x296	ARP/ghost options invalid for REV/FWD proxy
NSERR_PXY_ADD_SVR_OTHER	663	0x297	Simple content based no proxy attribs
NSERR_PXY_MB_DUP	664	0x298	Bind entry already exists
NSERR_PXY_DFLT_DUP	665	0x299	Default already exists
NSERR_SEL_INUSE	672	0x2a0	Selector already exists
NSERR_SEL_PARSE_FAILED	673	0x2a1	Error in selector expression syntax
NSERR_NOSELECTOR	674	0x2a2	No such selector
NSERR_SEL_TOOMANY	675	0x2a3	Selector limit reached
NSERR_CACHEGROUP_NOSELPARAM	677	0x2a5	Cannot change param-based content group to selector-based (or vice-versa)

ERROR CODE	ERROR CODE(DECIMAL)	ERROR CODE(HEX)	ERROR MESSAGE
NSERR_NOCACHEGROUP	678	0x2a6	Content group does not exist
NSERR_NONREQ_SEL	679	0x2a7	Selector expression is not request based
NSERR_INVALARG	680	0x2a8	The action qualifier and the specified argument(s) do not match
NSERR_NOWILD_ALLOWED	681	0x2a9	Wildcard not allowed with BETWEEN operator
NSERR_DATE_INCOMPAT	682	0x2aa	Time arguments incompatible
NSERR_TIMEDATE_INVALID	683	0x2ab	Date/time value invalid
NSERR_CONTENTGROUP_COOKIE_REQ_PARAM	684	0x2ac	Hit parameters or inval parameters needed to enable cookie matching
NSERR_CONTENTGROUP_IGNORECASE_REQ_HITPARAM	685	0x2ad	Hit parameters needed to enable case-insensitive param value matching
NSERR_CONTENTGROUP_INVALPARAM_REQ	686	0x2ae	Inval parameters needed to enable host-based invalidation
NSERR_CLISEC_EXP_TOO_LONG	687	0x2af	Client security expression too long

ERROR CODE	ERROR CODE(DECIMAL)	ERROR CODE(HEX)	ERROR MESSAGE
NSERR_NONHTTP_CSW_BIND_HTTP_SSL_POLICY	688	0x2b0	HTTP/SSL protocol based policies can be bound only to HTTP/SSL CS vserver
NSERR_NONHTTP_CSW_BIND_DOMAIN_POLICY	689	0x2b1	Domain based policies can be bound only to HTTP/SSL CS vserver
NSERR_NONHTTP_CSW_BIND_URL_POLICY	690	0x2b2	URL based policies can be bound only to HTTP/SSL CS vserver
NSERR_POLBOUNDTO_TOO_MANY_VSVRS	691	0x2b3	Domain cannot be set - policy bound to more than one vserver.
NSERR_PI_CSW_URL_DOMAIN	692	0x2b4	Domain/URL options disallowed along with advanced expressions
NSERR_PI_TO_PE_CSW	693	0x2b5	Advanced expressions based policies cannot be bound to a classic CS vserver.
NSERR_PE_TO_PI_CSW	694	0x2b6	Classic expression policies cannot be bound to a advanced CS vserver.
NSERR_PRIORITY_COMP_PI_CSW	695	0x2b7	Priority is mandatory for advanced expressions.

ERROR CODE	ERROR CODE(DECIMAL)	ERROR CODE(HEX)	ERROR MESSAGE
NSERR_PI_TCPCSW	696	0x2b8	Advanced expressions can not be bound to non-HTTP CS Vserver.
NSERR_PE_GOTO	697	0x2b9	Priority Jump expression cannot be bound with a classic expression based policy.
NSERR_PI_TO_PE_POL_CSW	698	0x2ba	Advanced expression based policy cannot be converted to a classic expression based policy and vice versa.
NSERR_CACHEOBJECT_EVICT	699	0x2bb	Cached object removed on expiry.
NSERR_DNSFAIL	700	0x2bc	DNS request failed
NSERR_HC_RET_TYPE_CHANGE	701	0x2bd	Cannot change callout return type.
NSERR_HC_NOT_HTTP_VS	702	0x2be	Not a HTTP or SSL vserver.
NSERR_HC_REQ_CONFIG_XOR	703	0x2bf	Full request expression and other request attributes cannot be set at the same time.
NSERR_HC_SERVICE_CONFIG_XOR	704	0x2c0	

ERROR CODE	ERROR CODE(DECIMAL)	ERROR CODE(HEX)	ERROR MESSAGE
NSERR_HC_RET_TYPE_REQD	705	0x2c1	Return type needs to be configured first.
NSERR_RTSP_CSW_BIND_IP_POLICY	706	0x2c2	Only IP based policies can be bound to RTSP CS vserver
NSERR_CSW_BIND_INCOMPAT_TGT	707	0x2c3	The target vserver is not compatible with the CS verver.
NSERR_CSW_BIND_INCOMPAT_CSPL	725	0x2d5	The target vserver is not compatible with the CS policylabel.
NSERR_AS_BAD_XMLNAMESPACE_PREFIX	708	0x2c4	Invalid XML-Namespace prefix.
NSERR_INVALID_SIP_EXPR	709	0x2c5	The SourceIP Expression specified is invalid.
NSERR_NO_BACKENDVSERVER	710	0x2c6	No backend LB vserver found.
NSERR_CONTENT_GROUP_TOOMANY	711	0x2c7	ContentGroup limit reached
NSERR_CACHE_MEM_SIZE_CHANGED	712	0x2c8	To use new Integrated Cache memory limit, save the configuration and restart the Citrix ADC.
NSERR_CACHE_MEM_SIZE_ZERO	713	0x2c9	No memory limit is configured for Integrated Cache. Use set cache

ERROR CODE	ERROR CODE(DECIMAL)	ERROR CODE(HEX)	ERROR MESSAGE
			parameter command to set the memory limit.
NSERR_L2CONN_NOT_ALLOWED	714	0x2ca	l2conn feature is supported only for ncore.
NSERR_SQL_NOT_ALLOWED	715	0x2cb	Database feature is supported only on nCore
NSERR_IPSEC_NOT_ALLOWED	716	0x2cc	IPSec feature is not supported
NSERR_HTMLPAGE_NOT_ALLOWED	717	0x2cd	Respondwith/ sqlresponse_ok/ sqlresponse_error actions cannot be set to respondwithhtmlpage action, and vice versa.
NSERR_PE_NOT_ALLOWED_DB	718	0x2ce	PE expressions are not allowed for DataStream.
NSERR_CSW_NO_POLICY_NAME	719	0x2cf	Policy name is required with priority.
NSERR_SQL_MSG_EXCEED	720	0x2d0	Message length should be smaller than 512
NSERR_CSW_CLASSIC_APPFW_POL	721	0x2d1	Binding classic appfw policy to CS vserver is not supported.

ERROR CODE	ERROR CODE(DECIMAL)	ERROR CODE(HEX)	ERROR MESSAGE
NSERR_ROUTING_STRIPED_SNIP	722	0x2d2	Routing protocols can run only on a spotted SNIP in a cluster
NSERR_USERNAME_MISSING	723	0x2d3	UserName argument Missing
NSERR_ACTION_EXIST	724	0x2d4	Default Action for this type already exist.
NSERR_WRONG_SERVICENAME	726	0x2d6	Use corresponding gslb servicecommand.
NSERR_PE_NOT_ALLOWED_RULE	727	0x2d7	PE expressions are not allowed for RULE based persistence
NSERR_NODE_NOT_IN_CLUSTER	728	0x2d8	Node is not in the cluster
NSERR_MCMX_CNRES	729	0x2d9	Error in communication with other cores
NSERR_HC_POST_REQ_CONFIG_XOR	730	0x2da	Parameters and body cannot be set at the same time for the POST method
NSERR_HC_CACGRP	731	0X2db	Operation on calloutContentGroup content group not permitted
NSERR_CSW_BIND_INCOMPAT_BKUP	732	0x2dc	The backup vserver of the target vserver is not compatible with the CS vserver.

ERROR CODE	ERROR CODE(DECIMAL)	ERROR CODE(HEX)	ERROR MESSAGE
NSERR_CACHE_MEM_RESIZE_IN_PROGRESS	733	0x2dd	Integrated Cache memory resize is in progress. Please wait a few seconds.
NSERR_CACHE_MEM_RESIZE_FAILED	734	0x2de	Integrated Cache memory resize failed.
NSERR_CSW_BIND_INCOMPAT_TARGET	735	0x2df	The target vserver has to be an LB vserver.
NSERR_ATTRIB_MEM_FREED	736	0x2e0	ATTRIB memory already freed
NSERR_RULE_MISSING	737	0x2e1	EvalRule argument Missing
NSERR_CACHEGROUP_CCH_MAXAGE_MAX	738	0x2e2	max-age value cannot be greater than 2147483647
NSERR_CACHE_MEM_REDUCTION_NOT_ALLOWED	739	0x2e3	Integrated Cache memory limit cannot be reduced.
NSERR_CSW_POL_INCOMPAT_LB	740	0x2e4	The target vserver is not compatible with the CS policy expression.
NSERR_CS_INVAL BACKUP	741	0x2e5	Content-Switching or Cache-Redirection vservers cannot be used as a backupvserver.

ERROR CODE	ERROR CODE(DECIMAL)	ERROR CODE(HEX)	ERROR MESSAGE
NSERR_CACHE_MEM_LIMIT_EXCEEDING	742	0x2e6	The IC memory configured for all the admin partitions that are defined on the appliance, must be limited to the IC memory that is configured on the default partition.
NSERR_PI_TO_PE_CRD	743	0x2e7	Advanced expressions based policies cannot be bound to a classic CR vserver.
NSERR_PE_TO_PI_CRD	744	0x2e8	Classic expression policies cannot be bound to a advanced CR vserver.
NSERR_INVAL_ARG_CRPL_LBVS	745	0x2e9	LB vserver cannot be bound to the CR vserver along with a CR policy.
NSERR_CRPEPOLICY_SETNOACTION	746	0x2ea	Cache-Redirection classic policy does not support action
NSERR_CACHE_MEM_LIMIT_NOT_CONFIGURED	747	0x2eb	Set the IC memory on the default partition before setting the IC memory for an admin partition.
NSERR_INVALID_BIND_PIPOL_CS	748	0x2ec	The Advanced Policy cannot be bound to a CS vserver

ERROR CODE	ERROR CODE(DECIMAL)	ERROR CODE(HEX)	ERROR MESSAGE
NSERR_INVALID_BIND_PIPOL_CR	749	0x2ed	The Advanced Policy cannot be bound to a CR vserver
NSERR_MEDIAC_MEMORY_ALLOC_FAILED	750	0x2ee	Memory allocation for media classification failed
NSERR_CRD_CLASSIC_APPFW_POL	751	0x2ef	Binding classic appfw policy to CR vserver is not supported.
NSERR_CSPOL_VPN_BIND	756	0x2f4	Invalid bind, VPN vserver is already set as a target Vserver on a CS action.
NSERR_CSACT_VPN	757	0x2f5	VPN vserver cannot be set as a target Vserver since CS policies are bound to it.
NSERR_CSPOL_ACT_VPN	759	0x2f7	CS policy cannot be bound to the vserver.CS action must be a LB vserver.
NSERR_VPNSERVER_NOTEXIST	760	0x2f8	The VPN Vserver does not exist
NSERR_VPNREFERRED_IN_CSBINDING	761	0x2f9	Cannot remove VPN Vserver which is referred in CS policy bindings.

ERROR CODE	ERROR CODE(DECIMAL)	ERROR CODE(HEX)	ERROR MESSAGE
NSERR_DEFAULTVPN_NOT_BOUND	762	0x2fa	Default VPN mentioned is not bound
NSERR_HDXINVAL	763	0x2fb	HDX servicetype cannot have WildCard IP and Port or 0 IP.
NSERR_CSW_ACTION_INCOMPTIBLE	764	0x2fc	Incompatible Content Switching action provided. During set operation the old and the new action should have compatible fields. eg, target LB vserver to target LB vserver.
NSERR_FWPXY_LICENSE	3578	0xdfa	Feature not licensed [FORWARDPROXY]
NSERR_GSLB_SVC_GRP_VSRVR_ERR	941	0x3ad	Binding of same GSLB servicegroup to heterogeneous IPV4 and IPV6 GSLB vservers is not allowed.
NSERR_GSLB_SVC_GRP_IP_ERR	942	0x3ae	Same GSLB servicegroup cannot be bound to both IPV4 and IPV6 addresses.
NSERR_GSLB_PUB_IP_PORT	943	0x3af	This configuration change will result in inconsistent private and public ip/port combination across

ERROR CODE	ERROR CODE(DECIMAL)	ERROR CODE(HEX)	ERROR MESSAGE
			GSLB services/ servicegroup items.
NSERR_GSLB_SVCGRP_PUBIP_PUBPORT_EXISTS	1341	0x53d	The Public IP and Public port is being used by another service item in the same GSLB servicegroup.
NSERR_NO_GSLBSVCGRP	1349	0x545	The GSLB servicegroup does not exist
NSERR_RENAME_GSLB_SVCGRP_WARN	1350	0x546	Please fire this command on all the participating GSLB sites manually or by using real time sync otherwise stats exchange between GSLB sites will not work.
NSERR_CS_PERSIST_NT_ALLWD	1688	0x698	Persistence is not supported for PROXY and CS GSLB type of CS vservers.

CQA ERRORS

CQA ERRORS

ERROR CODE	ERROR CODE(DECIMAL)	ERROR CODE(HEX)	ERROR MESSAGE
NSERR_CQA_LABEL_3_NOTSET	3782	0xec6	Network type net3 is not set
NSERR_CQA_WRONG_COEF	3783	0xec7	Coefficient values are not proper Need to give intercept and one feature from list. Check man page
NSERR_CQA_WRONG_CCL	3784	0xec8	Congestion level scale is not proper
NSERR_CQA_WRONG_CSQ	3785	0xec9	Signal Quality scale is not proper
NSERR_CQA_LR1_DEPENDENCY	3786	0xeca	Need to set network type net1 and net2
NSERR_CQA_LR2_DEPENDENCY	3787	0xecb	Need to set network type net2 and net3
NSERR_CQA_WRONG_LOG_COEF	3788	0xecc	Log Coefficient values are not proper
NSERR_CQA_LABEL_1_NOTSET	3789	0xecd	Network type net1 is not set
NSERR_CQA_LABEL_2_NOTSET	3791	0xecf	Network type net2 is not set

CVPN ERRORS

CVPN ERRORS

ERROR CODE	ERROR CODE(DECIMAL)	ERROR CODE(HEX)	ERROR MESSAGE
NSERR_NO_SUCH_PROFILE	3248	0xcb0	Profile does not exist
NSERR_PROFILE_IN_USE	3249	0xcb1	Profile in use
NSERR_NO_SAML PROFILE	3383	0xd37	Please specify a SAML SSO profile.
NSERR_NO_SUCH_PUSH_SERVICE	3003	0xbbb	Push service does not exist

DSK CACHE ERRORS

DSK CACHE ERRORS

ERROR CODE	ERROR CODE(DECIMAL)	ERROR CODE(HEX)	ERROR MESSAGE
NSERR_IC_CACHE_NOT_ENB	752	0x2f0	FEO enabled but IC needs to be enabled for most FEO features to work.
NSERR_DISK_ENGG_NOT_RUNNING	753	0x2f1	Disk Engine is not running, Save the config and reboot.
NSERR_DISK_ENGG_DISABLED	754	0x2f2	To take Disk Engine down, Save the config and reboot.
NSERR_IC_DISK_CACHE_PARAM_NOT_SET	755	0x2f3	Disk Cache not enabled. Set -enableDiskCache parameter to YES.

EV SUBSCRIBER ERRORS

EV SUBSCRIBER ERRORS

ERROR CODE	ERROR CODE(DECIMAL)	ERROR CODE(HEX)	ERROR MESSAGE
NSERR_EV_SUBSCRIBER_EXISTS	3489	0xda1	There exists a subscriber with the same name
NSERR_EV_SUBSCRIBER_NOT_FOUND	3490	0xda2	Subscriber does not exist
NSERR_EV_SUBSCRIBER_MAX_CONFIG	3491	0xda3	Maximum number of subscribers configured already
NSERR_EV_SUBSCRIBER_ALLOC_FAIL	3492	0xda4	Memory allocation failure while adding a subscriber
NSERR_EV_SUBSCRIBER_INVALID_TYPE	3493	0xda5	Invalid subscriber type
NSERR_EV_SUBSCRIBER_INVALID_URI	3494	0xda6	Invalid URI
NSERR_EV_SUBSCRIBER_SINFO_FAIL	3495	0xda7	Failed to create server info for the subscriber
NSERR_EV_SUBSCRIBER_INVALID_EVENT	3496	0xda8	Invalid event type
NSERR_EV_SUBSCRIBER_INVALID_ENTITY	3497	0xda9	Invalid entity type

ERROR CODE	ERROR CODE(DECIMAL)	ERROR CODE(HEX)	ERROR MESSAGE
NSERR_EV_SUBSCRIBER_ALL_TO_SPECIFIC	3498	0xdaa	Trying to (un)bind a specific enitty to an event; ALL entities were bound previously
NSERR_EV_SUBSCRIBER_SPECIFIC_TO_ALL	3499	0xdab	Trying to (un)bind ALL entities to an event; specific entities were bound previously

FWDSESSION ERRORS

FWDSESSION ERRORS

ERROR CODE	ERROR CODE(DECIMAL)	ERROR CODE(HEX)	ERROR MESSAGE
NSERR_FWDSESSION_NAME_EXISTS	928	0x3a0	A forwarding session with same name exists
NSERR_FWDSESSION_NW_EXISTS	929	0X3a1	A forwarding session with same network exists
NSERR_FWDSESSION_ACL_EXISTS	930	0X3a2	This acl is already configured
NSERR_FWDSESSION_ACL_NOT_EXISTS	931	0x3a3	Specified ACL is not configured
NSERR_FWDSESSION_CONN_FAILOVER	932	0x3a4	Connection fail-over not supported for v6 forwarding sessions

GENTOOL ERRORS

GENTOOL ERRORS

ERROR CODE	ERROR CODE(DECIMAL)	ERROR CODE(HEX)	ERROR MESSAGE
NSERR_SSL_CONFFILE	1601	0x641	Error in SSL conf file
NSERR_SSL_NOCONFFILE	1602	0x642	Unable to load SSL configuration info
NSERR_SSL_SIGFAIL	1603	0x643	Signing operation failed
NSERR_SSL_INVALFORMAT	1604	0x644	Invalid format
NSERR_SSL_OUTFILE	1605	0x645	Problem in writing output file
NSERR_SSL_VERIFY_FAIL	1606	0x646	Signature verification failed
NSERR_SSL_FILECREATE	1607	0x647	Unable to create output file
NSERR_SSL_MINKEYSIZE	1608	0x648	Key size less than 1024 is not supported on FIPS Firmware Version 1.1.
NSERR_SSL_INVALID_REQ	1609	0x649	Invalid certificate request
NSERR_SSL_GENTOOL	1610	0x64A	Internal failure in SSL cert/key generation tool

ERROR CODE	ERROR CODE(DECIMAL)	ERROR CODE(HEX)	ERROR MESSAGE
NSERR_SSL_PEM_ONLY	1611	0x64B	Only PEM format private key can be encrypted
NSERR_SSL_PASSMISMATCH	1612	0x64C	Password verification failed
NSERR_SSL_PASSREQ	1613	0x64D	Password required for private key
NSERR_SSL_INVALPASS	1614	0x64E	Invalid password
NSERR_SSL_MAXKEYSIZE	1615	0x64F	Maximum allowed key size is 2048
NSERR_SSL_GNTOOLARG_MISSING	1616	0x650	Required argument missing
NSERR_SSL_INVAL_PUBEXP	1617	0x651	Invalid public exponent value
NSERR_SSL_INVALID_VALUE	1618	0x652	Invalid value
NSERR_SSL_MISSING_KVAL	1619	0x653	Value missing for the argument
NSERR_SSL_INVAL_DHGEN	1620	0x654	Invalid DH generator value
NSERR_SSL_NOFILE_CERTREQ	1621	0x655	No such certificate request file exists
NSERR_SSL_REQ_NOMATCH	1622	0x656	Certificate Request (CSR) and private key do not match
NSERR_SSL_PKCS12	1623	0x657	Problem in input PKCS12 file

ERROR CODE	ERROR CODE(DECIMAL)	ERROR CODE(HEX)	ERROR MESSAGE
NSERR_SSL_NOFILE_CAPVTKEY	1624	0x658	No such CA key file present
NSERR_SSL_NOFILE_CACERT	1625	0x659	No such CA certificate file exists
NSERR_SSL_NOFILE_SERIAL	1626	0x65A	No such serial file exists
NSERR_SSL_NOFILE_PKCS12	1627	0x65B	No such PKCS12 file exists
NSERR_SSL_READ_CACERT	1628	0x65C	In reading input CA certficate file
NSERR_READ_CERT	1629	0x65D	Error in reading input certficate file
NSERR_IMP_EXP_NOT_TOGETHER	1630	0x65E	Import and export can not be specified together
NSERR_NOFILE_CERT	1631	0x65F	No such certificate file exists
NSERR_NOFILE_KEY	1632	0x660	No such key file exists
NSERR_CACERTPKEY_MISMATCH	1633	0x661	CA certificate not matching with CA private-key
NSERR_WRONG_REV_ENTRY	1634	0x662	Invalid revocation entry in CAs database
NSERR_WRONG_REV_DATE	1635	0x663	Invalid revocation data in CAs database

ERROR CODE	ERROR CODE(DECIMAL)	ERROR CODE(HEX)	ERROR MESSAGE
NSERR_WRONG_SERIAL_NO	1636	0x664	Invalid serial number in CAs database
NSERR_CREATE_DATABASE_ENTRY	1637	0x665	Revocation entry creation failed
NSERR_ALREADY_REVKD	1638	0x666	Certificate already revoked
NSERR_LOAD_PKEY	1639	0x667	Private key loading failed
NSERR_CERTPKEY_NEEDED	1640	0x668	Both certificate and private key are needed for export
NSERR_PKCS12_NEEDED	1641	0x669	PKCS12 file missing for import
NSERR_ERR_FILE EXISTS	1642	0x66A	Cannot create output file. File already exists
NSERR_FIPSKEY_KEYFILE_OPTION	1643	0x66B	Both -keyFileName and -fipsKeyName cannot be specified
NSERR_OUTFILELENBIG	1644	0x66C	Output file name too long
NSERR_SSL_DEFPATH	1645	0x66D	Output filepath should be under the partitions default SSL directory
NSERR_SSL_CRL_DEFPATH	1646	0x66E	Output CRL filepath should be under the partitions default CRL directory

ERROR CODE	ERROR CODE(DECIMAL)	ERROR CODE(HEX)	ERROR MESSAGE
NSERR_SSL_DEFLOCATION	1647	0x66F	Input file(s) not present or not accessible in current partition
NSERR_SSL_DNS_DEFPATH	1648	0x670	Output filepath should be under the partitions default DNS directory
NSERR_DHREQD	1791	0x6FF	DH required
NSERR_SSL_CRL_DEFLOCATION	1650	0x672	Input CRL file not present or not accessible in current partition
NSERR_SSL_NOFIPS_KEY	1651	0x673	No such FIPS key
NSERR_SSL_IMPROPER_SERIALFILE	1652	0x674	Invalid serial file
NSERR_SSL_SERIAL_NUM_REVOKED	1653	0x675	Certificate serial number match with another certificate already revoked in the CRL
NSERR_SSL_MAXRSAKEYSIZE	1654	0x676	Maximum allowed key size is 4096
NSERR_SSL_MUSTBEPEM	1655	0x677	The input file must be in PEM or SIM format.
NSERR_SSL_CANTBEPEM	1656	0x678	The input file cannot be in PEM format.

ERROR CODE	ERROR CODE(DECIMAL)	ERROR CODE(HEX)	ERROR MESSAGE
NSERR_SSL_BAD_LABEL	1657	0x679	The keylabel cannot be publicor private.
NSERR_SSL_HSMKEY_LOAD	1664	0x680	Unable to Load Key in external Hardware Security Module (HSM).
NSERR_SSL_HSMKEY_PPE	1665	0x681	Internal error while adding HSM key.
NSERR_ERR_SSL_HSMKEY_NOKEY	1666	0x682	Missing argument [Key].
NSERR_ERR_SSL_HSMKEY_DELETE	1667	0x683	Unable to Delete Key in external Hardware Security Module (HSM).
NSERR_ERR_SSL_HSMKEY_NOMATCH	1668	0x684	Specified HSM Type does not match.
NSERR_ERR_SSL_HSMKEY_REFEXT	1669	0x685	The HSM key is referenced by a certificate
NSERR_ERR_SSL_HSMKEY_NOTEXISTS	1670	0x686	No such HSM key.
NSERR_ERR_SSL_HSMKEY_NOSTORE	1671	0x687	Specified keystore does not exist.
NSERR_ERR_SSL_GW_NOTUP	1672	0x688	Gateway daemon is booting up please wait for 20 to 30 secs.

ERROR CODE	ERROR CODE(DECIMAL)	ERROR CODE(HEX)	ERROR MESSAGE
NSERR_ERR_SSL_HSMKEY_FIPS	1673	0x689	Operation not permitted on a MPX-FIPS System.
NSERR_ERR_SSL_SANCERT_DUP	1674	0x68A	Duplicate domain cert cannot be added.
NSERR_SSL_ECDSA_SUPPORT	1675	0x68B	ECDSA certificates are not supported for this entity/platform.
NSERR_SSL_ECDSA_CURVE_SUPPORT	1676	0x68C	The EC curve in the ECDSA certificate is not supported.
NSERR_SSL_LINK_LOOP	1677	0x68D	Certificate cant be linked. This is creating a loop.
NSERR_SSL_N3FIPSREBOOT	1678	0x68E	N3FIPS InitHSM is not done, Please reboot again
NSERR_SSL_N3FIPS_FILECREATE	1679	0x68F	FIPS config folder /nsconfig/fips is not present, please create it manually and reboot the system, after reboot reset and set fips again
NSERR_SSL_SIGHASH_SUPPORT	1680	0x690	List contains unsupported SigHash type for this entity/platform.

ERROR CODE	ERROR CODE(DECIMAL)	ERROR CODE(HEX)	ERROR MESSAGE
NSERR_SSL_SAN_CERT_ENTRY	1681	0x691	Unsupported OID for SAN entry in certificate
NSERR_CIPHER_BACKEND	1682	0x692	The cipher alias cannot be bound to this entity. It is only applicable for a SSL backend service or a SSL serviceGroup
NSERR_SSL_CIPHER_NOT_SUPPORTED_ON_PLATFORM	1683	0x693	This Cipher/ CipherAlias is not supported on this platform.
NSERR_SSL_I2D_X509_NAME_FAILURE	1684	0x694	Failed to encode X509_NAME into DER format
NSERR_DTLS_CIPHER_SUPPORT	1685	0x695	This cipher alias cannot be bound. DTLS vServer/ Service only supports DEFAULT_DTLS/ DEFAULT_DTLS_BACKEND ciphers respectively.
NSERR_DTLS_CIPHER_BACKEND	1686	0x696	The cipher alias cannot be bound to this entity. It is only applicable for a DTLS backend service or a DTLS serviceGroup.

GSLB ERRORS

GSLB ERRORS

ERROR CODE	ERROR CODE(DECIMAL)	ERROR CODE(HEX)	ERROR MESSAGE
NSERR_BAD_DNS_OPTION	1792	0x700	Invalid DNS option specified
NSERR_DNS_BAD_CACHETYPE	1793	0x701	Only forward cache type can bind to DNS vserver
NSERR_DNS_NOTAVAIL	1794	0x702	No such DNS vserver exists
NSERR_DNS_BAD_VSTYPE	1795	0x703	DNS vserver can only be address-based
NSERR_GSLB_SITELKUP_FAILED	1796	0x704	GSLB site not found
NSERR_GSLBBOUNDTO_CSACT	1797	0x705	The GSLB Vserver referred by Content Switching action cannot be removed
NSERR_GSLBREFFERED_IN_CSBINDING	1798	0x706	Cannot remove GSLB Vserver which is reffered in CS policy bindings
NSERR_NOAREC	1801	0x709	Address record for host name not available
NSERR_INVALTTL	1806	0x70E	Invalid TTL value
NSERR_CNAMEEXISTS	1807	0x70F	Alias name record exists for the host name

ERROR CODE	ERROR CODE(DECIMAL)	ERROR CODE(HEX)	ERROR MESSAGE
NSERR_INVAL_SVCOPTION	1808	0x710	Option not applicable for this service
NSERR_GSLBBIND_EXISTS	1809	0x711	GSLB binding exists for the given host
NSERR_NOT_LOCALREMOTE	1810	0x712	Site information must be specified for a GSLB service
NSERR_COUNT_MISMATCH	1811	0x713	Address record count does not match the number of services bound to the GSLB vserver
NSERR_IP_MISMATCH	1812	0x714	IP addresses of bound GSLB services and the address records do not match
NSERR_SVCIP_REPEAT	1813	0x715	A GSLB service with the same IP address is already bound to the GSLB vserver
NSERR_NOT_REMOTE	1814	0x716	Service bound to GSLB VIP with -gslb remote is a local service
NSERR_NOT_LOCAL	1815	0x717	Service bound to GSLB VIP with -gslb local is not a local service
NSERR_PROXY_REC	1816	0x718	A proxy record cannot be assigned to the domain

ERROR CODE	ERROR CODE(DECIMAL)	ERROR CODE(HEX)	ERROR MESSAGE
NSERR_NOPROXY_AREC	1817	0x719	Proxy address records cannot be bound to a GSLB vserver
NSERR_SVC_GSLBBIND_EXISTS	1819	0x71B	GSLB binding exists - unbind the GSLB vserver first
NSERR_SVCTYPEMISMATCH	1820	0x71C	Service type mismatch with existing service/ vserver
NSERR_INVALID_PUBIP_OPTION	1822	0x71E	The public IP and public port are valid only for GSLB local service
NSERR_INVAL_NAME_SYNTAX	1823	0x71F	Invalid domain name syntax
NSERR_GSLB_BACKUP	1824	0x720	Backup VIP is backup of other GSLB VIP. Please unset the existing association and try again
NSERR_IS_BACKUP	1825	0x721	Backup VIP is backup of other VIP. Please unset the existing association and try again
NSERR_GSLB_HAS_BACKUP	1826	0x722	GSLB VIP has a backup. Please unset the backup and try the operation again

ERROR CODE	ERROR CODE(DECIMAL)	ERROR CODE(HEX)	ERROR MESSAGE
NSERR_GSLB_REQUIRES_INDEPENDENT_BACKUP	1827	0x723	Backup VIP to GSLB VIP must not be backup of any other VIP. Please unset the backup and try the operation again
NSERR_GSLB_NOQUALIFIER	1828	0x724	At least one qualifier should be given for the location
NSERR_GSLB_NONAME	1829	0x725	The qualifier does not exist in the repository
NSERR_GSLB_CUSTOM_NOSPACE	1830	0x726	Too many custom locations
NSERR_GSLB_STATIC_NOSPACE	1836	0x72C	Too many static locations
NSERR_GSLBSITENAME_EXISTS	1831	0x727	A GSLB site with the same name exists
NSERR_GSLBSITEIP_EXISTS	1832	0x728	A GSLB site with the same IP address exists
NSERR_NO_GSLBSITE	1833	0x729	The GSLB site does not exist
NSERR_LOCAL_EXISTS	1834	0x72A	The local site already exists
NSERR_NO_GSLBSVC	1835	0x72B	The GSLB service does not exist
NSERR_NO_GSLBVIP	1837	0x72D	The GSLB vserver does not exist

ERROR CODE	ERROR CODE(DECIMAL)	ERROR CODE(HEX)	ERROR MESSAGE
NSERR_NO_DMN	1839	0x72F	The domain does not exist
NSERR_DMN_NOT BOUND	1840	0x730	The domain is not bound to this VIP
NSERR_SVCGRP_ NOT_SUPPORTED_ ON_HEIRARCHICH AL	1841	0x731	GSLB servicegroup is not supported in parent child topology
NSERR_GSLBDOM AIN_BOUND	1842	0x732	The domain is already bound to a GSLB vserver
NSERR_NOT_GSLB ENT	1844	0x734	Operation not permitted on a non-GSLB entity
NSERR_GSLBDOM AIN_PERM	1845	0x735	Operation not permitted on a GSLB-configured domain
NSERR_GSLB_MAX QUAL	1846	0x736	Maximum qualifiers allowed is six
NSERR_GSLB_MAX LOCLENGTH	1847	0x737	Maximum allowed location length is 256 characters
NSERR_GSLBVIP_ PERM	1848	0x738	Operation not permitted on a GSLB vserver
NSERR_GSLBSVC_ PERM	1849	0x739	Operation not permitted on a GSLB service

ERROR CODE	ERROR CODE(DECIMAL)	ERROR CODE(HEX)	ERROR MESSAGE
NSERR_GSLB_MAXSITES	1850	0x73A	Maximum number of sites is 32
NSERR_GSLB_QUALTOOLONG	1851	0x73B	The qualifier is too long
NSERR_GSLB_PERSISTIDEXISTS	1852	0x73C	The persistence ID is already being used by a GSLB vserver
NSERR_GSLB_INVLD_LDNSTO_INTVL	1854	0x73E	Minimum LDNS entry timeout interval is 30 secs
NSERR_VIP_BACKUP_ISGSLB	1856	0x740	GSLB vserver cannot be backup of a non-GSLB vserver
NSERR_GSLB_PUBIP_PUBPORT_EXISTS	1857	0x741	The public IP and public port are being used by another GSLB service
NSERR_GSLBLOCALSVC_ENADIS_NOTALLOWED	1858	0x742	GSLB local service status not changed - please enable or disable the corresponding LB vserver instead
NSERR_GSLBLOCALSVR_ENADIS_NOTALLOWED	1859	0x743	Server status not changed - the server has a GSLB local service
NSERR_INVALID_PERSISTID	1860	0x744	Invalid persistence ID
NSERR_PERM_PERSISTID	1861	0x745	The persistence ID is required for persistence or

ERROR CODE	ERROR CODE(DECIMAL)	ERROR CODE(HEX)	ERROR MESSAGE
			spillover-persistency to be set
NSERR_INVALID_LOCALSITEIP	1862	0x746	Invalid GSLB local site IP address
NSERR_INVALID_REMOTESITEIP	1863	0x747	Invalid GSLB remote site IP address
NSERR_INVALID_SITETYPE	1864	0x748	Invalid GSLB site type
NSERR_INVALID_BACKUPIP	1865	0x749	Invalid backup IP for the GSLB domain
NSERR_GSLBVIP_MAXSVC	1866	0x74A	Cannot bind more than 32 services to a GSLB vserver
NSERR_INVALID_JITTER	1867	0x74B	RTT tolerance value must be between 1 and 100
NSERR_GSLBLOCSVC_DIS_NOTALLOWD	1868	0x74C	Cannot add GSLB local service with state set to DISABLED
NSERR_DUMMYBKUP_NOTALLOWD	1869	0x74D	A backup vserver with no IP address and port must be of type GSLB
NSERR_GSLBSVC_BIND_NOTALLOWD	1870	0x74E	Cannot bind GSLB service to a vserver with valid IP and port
NSERR_NSMAP_VERSION	1871	0x74F	Unrecognized Citrix ADC file version

ERROR CODE	ERROR CODE(DECIMAL)	ERROR CODE(HEX)	ERROR MESSAGE
NSERR_NSMAP_FORMAT	1872	0x750	Unrecognized file format specified
NSERR_NSMAP_EOF	1873	0x751	End of file has been reached
NSERR_NSMAP_PARSE	1874	0x752	File parsing error
NSERR_NSMAP_READ	1875	0x753	Error reading file
NSERR_NSMAP_WRITE	1876	0x754	Error writing file
NSERR_NSMAP_IOCTL	1877	0x755	Error occurred while transferring information to Citrix ADC
NSERR_NSMAP_OUTPUTFILE	1878	0x756	Error opening output file for writing
NSERR_NSMAP_INPUTFILE	1879	0x757	Error opening input file for reading
NSERR_NSMAP_REQUIRED_FIELD	1880	0x758	Required field missing
NSERR_NSMAP_DBFILE	1881	0x759	Error opening database file
NSERR_NSMAP_DBINSERT	1882	0x75A	Error inserting entry into the database
NSERR_NSMAP_DBSEARCH	1883	0x75B	Error searching the database
NSERR_GSLBPROXM_LICENCE_ABSENT	1884	0x75C	Cannot set proximity methods without

ERROR CODE	ERROR CODE(DECIMAL)	ERROR CODE(HEX)	ERROR MESSAGE
			GSLB proximity license
NSERR_GSLB_SITEPERSISTENCE_HTTPONLY	1885	0x75D	Site persistence is valid only for HTTP or SSL GSLB services
NSERR_GSLB_NOSITEPFX	1886	0x75E	Site prefix compulsory if site persistence is HTTPRedirect
NSERR_GSLB_SITEPREFIX_SIZE	1888	0x760	Site prefix size exceeds maximum
NSERR_GSLB_SITEDOMAIN_SIZE	1889	0x761	Site domain cannot be created - name length exceeds maximum
NSERR_GSLB_SITEDOMAIN_SYNTAX	1890	0x762	Site domain cannot be created - name is invalid
NSERR_GSLB_SITEDOMAIN_REFERS	1891	0x763	Site domain cannot be deleted - it has external DNS references
NSERR_GSLBDOMAIN_REFERS	1892	0x764	GSLB domain cannot be deleted - it has external DNS references
NSERR_GSLB_ARECORD_EXISTS	1893	0x765	GSLB record cannot be created - another A record is configured with the same name

ERROR CODE	ERROR CODE(DECIMAL)	ERROR CODE(HEX)	ERROR MESSAGE
NSERR_GSLB_SITECOOKIE_TIMEOUT_RANGE	1894	0x766	Invalid GSLB site cookie timeout
NSERR_GSLBDOMAIN_SYNTAX	1895	0x767	GSLB domain cannot be created - name is invalid
NSERR_GSLB_SO_PERM	1896	0x768	Spill-over cannot be set on a GSLB vserver
NSERR_GSLB_DEPRCOOKIETOUT	1897	0x769	Ignored deprecated cookietimeout setting since it has been made a per-domain configuration
NSERR_DNS_NAMESVR_SYNTAX	1898	0x76a	Invalid name server syntax
NSERR_DNS_ORIGSVR_SYNTAX	1899	0x76b	Invalid origin server syntax
NSERR_DNS_CONTACT_SYNTAX	1900	0x76c	Invalid contact name syntax
NSERR_DNS_MX_SYNTAX	1901	0x76d	Invalid mail exchange syntax
NSERR_DNS_CNAME_SYNTAX	1902	0x76e	Invalid canonical name syntax
NSERR_DNS_ALIAS_SYNTAX	1904	0x770	Invalid alias name syntax
NSERR_GSLB_NOLOCALVIP	1905	0x771	Cannot create GSLB local service - add the local vserver first

ERROR CODE	ERROR CODE(DECIMAL)	ERROR CODE(HEX)	ERROR MESSAGE
NSERR_GSLB_LOCALSVCEXISTS	1906	0x772	Cannot delete the vserver - corresponding GSLB local service still exists
NSERR_GSLBLOCALSVC_PERM	1907	0x773	Operation not permitted on GSLB local service
NSERR_DNS_INVAL_REVDOMNAME_SYNTAX	1908	0x774	Invalid reverse domain name syntax
NSERR_DNS_ALIASREC	1909	0x775	Alias name cannot have any record
NSERR_GSLBLOCSVC_DELAYEDCLEAN_NOTALLOWD	1910	0x776	Down state flush cannot be set on GSLB local service
NSERR_NO_BKP_VIP	1911	0x777	Set a backup Vserver, to enable it.
NSERR_GSLB_NOSUCH_LDNSENTRY	1912	0x778	LDNS entry not present
NSERR_MAX_DNS_VIEW	1913	0x779	Number of DNS views exceeds limit
NSERR_GSLB_SITE_IP_EXISTS	1914	0x77a	GSLB Site IP already exists
NSERR_GSLB_LAST_MIP	1915	0x77b	MIP is not removed. At least one MIP is required
NSERR_RM_GSLB_SITE	1916	0x77c	IP cannot be removed as it is

ERROR CODE	ERROR CODE(DECIMAL)	ERROR CODE(HEX)	ERROR MESSAGE
			being used by the GSLB local site
NSERR_GSLB_IPRM_LAST_MIP	1917	0x77d	GSLB Site IP removed. MIP is not removed. At least one MIP is required
NSERR_GSLB_SVC_PUB_PORT_ERR	1918	0x77e	Cannot modify public port for remote GSLB service/ service item
NSERR_REC_HAS_REF	1919	0x77f	DNS record has references
NSERR_DNS_VIEW_REF	1920	0x780	The dns view cannot be removed as it has dependencies
NSERR_GSLB_SITE_PERSISTENCE_MATCH	1921	0x781	Mismatch in gslb site persistence type with other bound services
NSERR_GSLB_SITE_PERSISTENCE_CONFLICTS	1922	0x782	Site persistence conflicts with other services/ servicegroups bound to the gslb vserver(s)
NSERR_DNS_POLICY_INVAL	1923	0x783	Invalid DNS policy
NSERR_GSLBVIP_CNAME_MISMATCH	1924	0x784	GSLB vserver cannot be bound with cname based and ip based services at the same time

ERROR CODE	ERROR CODE(DECIMAL)	ERROR CODE(HEX)	ERROR MESSAGE
NSERR_GSLBVIP_CNAME_UNSUPPORTED_LBMETHODS	1925	0x785	GSLB vserver having cname services cannot have dynamic load balancing methods
NSERR_GSLB_SERVICE_MON_NOIPPORT	1926	0x786	Cannot bind monitor with zero ip or port to a cname gslb service
NSERR_GSLB_SERVICE_MON_SET_NOIPPORT	1927	0x787	Cannot set ip or port to zero when monitor is bound to a cname gslb service.
NSERR_GSLB_CNAME_SERVICE_SET	1928	0x788	Setting of these parameters on cname gslb service is not allowed
NSERR_GSLB_CNAME_VSERVER_SET	1929	0x789	Setting of these parameters on gslb vserver having cname services is not allowed
NSERR_GSLBVIP_CNAME_UNSUPPORTED_BKUP_LBMETHODS	1930	0x78a	GSLB vserver having cname services cannot have dynamic backup load balancing methods
NSERR_GSLBVIP_CNAME_UNSUPPORTED_EDR	1931	0x78b	GSLB vserver having cname services cannot have empty down response disabled

ERROR CODE	ERROR CODE(DECIMAL)	ERROR CODE(HEX)	ERROR MESSAGE
NSERR_GSLBVIP_CNAME_UNSUPPORTED_MIR	1932	0x78c	GSLB vserver having cname services cannot have multiple ip response enabled
NSERR_GSLBSVC_VIEW_NEXIST	1933	0x78d	View is not bound to the GSLB service
NSERR_DNS_TTL_MORE_THAN_MAX_ALLOWED	1934	0x78e	DNS RECORD TTL value greater than max TTL allowed
NSERR_GSLB_LB_MAXSITES	1935	0x78f	Maximum number of LB sites is 1024
NSERR_GSLB_HASCHILDREN	1936	0x790	Cannot remove site that has child site(s)
NSERR_GSLB_PARENTISCHILD	1937	0x791	Parent site cannot be a child to another site
NSERR_MAX_STREAM_SELECTORS	1938	0x792	Number of selectors exceeds limit
NSERR_MAX_STREAM_IDENTIFERS	1939	0x793	Number of identifiers exceeds limit
NSERR_NO_SUCH_SELECTOR	1940	0x794	The given selector does not exist
NSERR_NO_SUCH_IDENTIFIER	1941	0x795	The given identifier does not exist
NSERR_RLT_TIMESLICE_INVALID_VAL	1942	0x796	Time slice should be a multiple of 10
NSERR_ILLEGAL_SUBNET_MASK	1943	0x797	Incorrect subnet mask value

ERROR CODE	ERROR CODE(DECIMAL)	ERROR CODE(HEX)	ERROR MESSAGE
NSERR_STREAM_SELECTOR_INUSE	1944	0x798	The selector is being referenced by one or more identifiers
NSERR_IDENTIFIER_INUSE	1945	0x799	The identifier is being referenced by one or more policies
NSERR_SELECTOR_CANNOT_CHANGE_ATTRIB_TYPE	1946	0x79A	Cannot change request type selector to response type
NSERR_GSLB_IGN_TRIGMON	1947	0x79B	Trigger monitor setting assumed ALWAYS for GSLB local site
NSERR_SELECTOR_NOT_MORE_THAN_2_IPV6_EXP	1948	0x79C	Cannot exceed more than two IPV6 expressions in a selector
NSERR_REPEATED_MONITORS	1949	0x79D	Monitors cannot be repeated
NSERR_ALL_MONITORS_DISABLED	1950	0x79E	All monitors in selected list are disabled.
NSERR_GSLB_AAAARECORD_EXISTS	1951	0x79F	GSLB record cannot be created - another AAAA record is configured with the same name
NSERR_GSLBVIP_HETEROGENEOUS_SERVICE_IPVERSION	1952	0x7A0	Heterogeneous service/servicegroup binding is not allowed for GSLB vserver

ERROR CODE	ERROR CODE(DECIMAL)	ERROR CODE(HEX)	ERROR MESSAGE
NSERR_VIEWIP_IPV6	1953	0x7A1	View IP setting is not allowed for IPv6 GSLB service.
NSERR_BACKUPIP_IPV6	1954	0x7A2	IPv4 backup IP setting is not allowed for IPv6 GSLB vserver.
NSERR_BACKUP_VIP_MISMATCH	1955	0x7A3	Both primary and secondary vservers should be of same DNSRecordType
NSERR_GSLB_DOMAIN_CONVERSION	1956	0x7A4	Cannot associate gslb domain to the service.
NSERR_GSLB_IPV4_BACKUPIP	1957	0x7A5	IPv6 service binding is not allowed along with IPv4 backup IP.
NSERR_CANNOT_ADD_SELECTOR	1958	0x7A6	Cannot set a new selector when the identifier with no selector is bound to a policy.
NSERR_SEL_CONFLICTING_ATTRIBUTES	1959	0x7A7	Conflicting attributes in the new and current selector sets.
NSERR_GSLBVIP_CNAME_BACKUPIP	1960	0x7A8	GSLB vserver having cname services cannot have backup IP
NSERR_GSLB_HETEROGENEOUS_SITEIP	1961	0x7A9	Heterogeneous IPv6 and IPv4 GSLB site IPs not allowed.

ERROR CODE	ERROR CODE(DECIMAL)	ERROR CODE(HEX)	ERROR MESSAGE
NSERR_DNS_MAX_AAAA_REC	1962	0x7AA	Cannot exceed more than 5 IPv6 addresses.
NSERR_DNS_MAX_A_REC	1963	0x7AB	Cannot exceed more than 25 IP addresses.
NSERR_GSLB_RECTYPE_SERVICE	1965	0x7AD	Cannot set the record type while services/ servicegroups are bound to the GSLB vserver.
NSERR_GSLB_RECTYPE_DOMAIN	1966	0x7AE	Cannot set the record type while domains bound to the GSLB vserver.
NSERR_GSLB_RECTYPE_BACKUP_VIP	1967	0x7AF	Cannot set the record type while GSLB vserver has backup vserver.
NSERR_GSLB_BKPVIP_RECTYPE	1968	0x7B0	Cannot set the record type while GSLB vserver is a backup vserver.
NSERR_GSLBVIP_NAPTR_UNSUPPORTED_SVCTYPE	1969	0x7B1	GSLB vserver of NAPTR DNS record type can have only ANY as service type.
NSERR_BACKUPIP_IPV4	1970	0x7B2	IPv6 backup IP setting is not allowed for IPv4 GSLB vserver.

ERROR CODE	ERROR CODE(DECIMAL)	ERROR CODE(HEX)	ERROR MESSAGE
NSERR_GSLB_IPV6_BACKUPIP	1971	0x7B3	IPv4 service binding is not allowed along with IPv6 backup IP.
NSERR_GSLB_DB_REQ_TOO_BIG	1972	0x7B4	Static proximity Database request is too big.
NSERR_GSLB_DB_QUEUE_MAXED	1973	0x7B5	Static proximity Database queue is full.
NSERR_GSLB_DB_SERVER	1974	0x7B6	Static proximity database server is not running.
NSERR_GSLB_DB_TIMEOUT	1975	0x7B7	Static proximity database server is not responding.
NSERR_GSLB_DB_CLOSED	1976	0x7B8	Connection to static proximity database server is closed.
NSERR_GSLB_COORDINATES	1977	0x7B9	Both longitude and latitude should be specified or not.
NSERR_INVALID_IPV6_PREFIXLEN	1978	0x7BA	Invalid IPv6 prefix length.
NSERR_GSLBDOMAIN_CANME_RECORD_EXISTS	1979	0x7BB	Cname record cant be created. Another record exists for the configured domain.
NSERR_GSLBDOMAIN_AAAA_CNAME_EXISTS	1980	0x7BC	Cname record exists for the configured domain.

ERROR CODE	ERROR CODE(DECIMAL)	ERROR CODE(HEX)	ERROR MESSAGE
NSERR_GSLB_NON_PARENT_REMOTESITE	1981	0x7BD	Cannot have a gslb remote site, other than the parent site, configured on an LB node
NSERR_ZONE_EXISTS	1982	0x7BE	Zone already configured.
NSERR_INVAL_KEYFLAGS	1983	0x7BF	Invalid flags value in DNSKEY
NSERR_DNSKEY_NONEXISTS	1984	0x7C0	Zone has no DNSKEYs
NSERR_NORECS_IN_ZONE	1985	0x7C1	No records in zone
NSERR_SIGNFAILED	1986	0x7C2	RSA Sign operation returned error
NSERR_DIGESTINIT_FAILED	1987	0x7C3	Digest init before RSA Sign operation returned error
NSERR_DIGESTUPDATE_FAILED	1988	0x7C4	Digest update before RSA Sign operation returned error
NSERR_KEYEXISTS	1989	0x7C5	The DNSKEY is already added for this zone
NSERR_NOPASVKEYS	1990	0x7C6	No Passive DNSKEYs in zone
NSERR_DNS_MAXKEYSIZE	1991	0x7C7	Maximum allowed public key size is 1024 and private key size is 4096

ERROR CODE	ERROR CODE(DECIMAL)	ERROR CODE(HEX)	ERROR MESSAGE
NSERR_NODATA_TO_SIGN	1992	0x7c8	No data to sign the record
NSERR_CNAME_SITE_NOTEXISTS	1993	0x7c9	Site name must be specified for cname based GSLB service.
NSERR_DNS_NOSOA_NONS	1994	0x7cA	No SOA or NS records for the zone
NSERR_DNS_PROXY_ZONE	1995	0x7cB	Cannot sign/unsign a proxy zone
NSERR_NOACTVKEYS	1996	0x7CC	No active DNSKEYs in zone
NSERR_NOTIFYPERIOD	1997	0x7cd	Notification period must be less than the expiry period
NSERR_LOAD_PUBKEY	1998	0x7ce	Public key loading failed
NSERR_DNSKEYGEN_UNSUPPORTED_ALGO	1999	0x7cf	Unsupported algorithm
NSERR_DNSKEYGEN_ERR_PUBFILE_OPEN	2000	0x7d0	Error opening public key file
NSERR_DNSKEYGEN_ERR_PRIVFILE_OPEN	2001	0x7d1	Error opening private key file
NSERR_DNSKEYGEN_ERR_DSFILE_OPEN	2002	0x7d2	Error opening delegation signer (DS) file

ERROR CODE	ERROR CODE(DECIMAL)	ERROR CODE(HEX)	ERROR MESSAGE
NSERR_GSLB_OPT_NOTSUPPORTED	2003	0x7d3	gslb option is not supported, GSLB site is mandatory for GSLB services
NSERR_GSLB_GFS_NOT_SUPPORTED	2004	0x7d4	Graceful shutdown is not supported for GSLB services.
NSERR_NSMAP_IMPORTFILE	2005	0x7d5	Error reading the imported file. See /var/log/ns.log file for more details.
NSERR_DNS_CNAMELOOP	2006	0x7d6	Addition of this CNAME record leads to a CNAME loop
NSERR_NEWSVC_NOT_4GSLB	2007	0x7d7	startup settings not applicable to GSLB vservers
NSERR_NON_CLUSTER_SYSTEM	2008	0x7d8	The option is only supported for clustering systems.
NSERR_PM_MON_NOT_ENABLED	2009	0x7d9	Path monitoring must be enabled to enable this option.
NSERR_GSLB_BACKUP_PERS_DEPRICATED	2010	0x7dA	GSLB Backup Persistence is deprecated. Please use Spillover Persistence instead.
NSERR_DNS_SIGNFAIL	2011	0x7dB	DNSSEC Sign operation failed

ERROR CODE	ERROR CODE(DECIMAL)	ERROR CODE(HEX)	ERROR MESSAGE
NSERR_DNS_NEED_PROXY	2012	0x7dC	Zone needs to be in proxy mode
NSERR_DNSSEC_OFFLOAD_NOSUPPORT	2013	0x7dD	DNNSEC Offload is not supported in this platform.
NSERR_DNS64_ACT_INVAL	2014	0x7dE	Invalid DNS64 action.
NSERR_DNS64_NOT_SUPPORTED_VS	2015	0x7dF	Specified policy can be bound only to DNS/DNS_TCP vserver.
NSERR_DNS64_INVAL_PREFIX	2016	0x7e0	Specified Prefix is either not allowed or Prefix length is not in [96, 64, 56, 48, 40, 32].
NSERR_GSLB_DB_REPEAT_REQUEST	2017	0x7e1	Location database request need to be repeated
NSERR_GSLBSITE_CONN_NOTSUPPORTED	2018	0x7e2	This may cause GSLB site MEP connection issues in some earlier releases.
NSERR_DNS64_TIMEOUT_ROUNDOFF	2019	0x7e3	Timeout threshold is rounded off to nearest multiple of 10 less then the given value.
NSERR_USIP_EV_NO_MEM_WARNING	2020	0x7e4	Service marked unusable due to lack

ERROR CODE	ERROR CODE(DECIMAL)	ERROR CODE(HEX)	ERROR MESSAGE
			of memory to post APBR rules for USIP.
NSERR_GSLB_SITE_PREFIX_CONFLICT	2044	0x7fc	Site prefix conflicts with another service bound to the gslb vip
NSERR_USIP_EV_NO_MEM_WARNING_DEL	2021	0x7e5	APBR rule on switch not deleted due to lack of memory to post APBR rules for USIP.
NSERR_USIP_EV_NO_SVC_GRP_MEM_WARNING	2022	0x7e6	One or more APBR rules were not configured on the switch due to lack of memory.
NSERR_USIP_EV_NO_SVC_GRP_MEM_WARNING_DEL	2023	0x7e7	One or more APBR rules were not deleted on the switch due to lack of memory.
NSERR_DNSKEY_MISMATCH	2024	0x7e8	The Key specified is not valid for this zone
NSERR_INVALID_REMOTE_CLIP	2025	0x7e9	Invalid remote GSLB site CLIP. CLIP belong to the local gslb site
NSERR_IPV6_CLIP_NOTSUPPORTED	2026	0x7ea	IPV6 CLIP is not supported
NSERR_NON_DEFAULT_DBFILE_LOC	2027	0x7eb	In case of HA setup and cluster system, the command might fail on other nodes if an identical copy of

ERROR CODE	ERROR CODE(DECIMAL)	ERROR CODE(HEX)	ERROR MESSAGE
			the file is not present at the same location.
NSERR_LOC64_FROM_TO_TYPE	2028	0x7ec	IP From and IP To in location specification are of different IP type.
NSERR_DNS_NOSOA_NONS_ZONE_NO_PROXY_MODE	2029	0x7ed	Cannot delete SOA or NS records for a zone that is not deployed in proxy mode.
NSERR_DNS_ACTION_DROP_DEPRECATED	2031	0x7ef	Drop is deprecated. Please use Drop action type under Responder feature.
NSERR_DNS_ACTION_REWRITE_RESPONSE_DEPRECATED	2039	0x7f7	Rewrite Response is deprecated. Please use replace_dns_answer_section action type under Rewrite feature.
NSERR_IDENTIFIER_INVAL_THRESHOLD	2040	0x7f8	Stream Identifier min and max threshold cannot both be zero and min threshold cannot be greater than max threshold
NSERR_IDENTIFIER_INVAL_SORT	2041	0x7f9	Stream Identifier sort option is invalid
NSERR_NO_SUCH_DNSPROF	2042	0x7fa	DNS profile does not exist

ERROR CODE	ERROR CODE(DECIMAL)	ERROR CODE(HEX)	ERROR MESSAGE
NSERR_DNS_PROFILE_SVC_TYPE	2043	0x7fb	DNS profile cannot be set to this service type
NSERR_CSGSLB_POLICY_SET_NOTALLOWED	2046	0x7fe	Policy bound to CS GSLB vserver must have a gslb targetvserver
NSERR_CS_POLICY_SET_NOTALLOWED	2045	0x7fd	Policy bound to non CS GSLB vserver must not have a gslb targetvserver
NSERR_GSLB_ECS_DISABLED_VALIDATION_ENABLED	2987	0xbab	ECS is disabled: Enabling only ECS address validation will not have any effect if ECS is not enabled.
NSERR_ECS_INVALID_ADDR_LEN	2989	0xbad	ECS address uses more octets than required or the address has non-zero bits set beyond SOURCE PREFIX-LENGTH
NSERR_GSLB_ECS_MIR_BOTH_SET	765	0x2fd	ECS option is mutually exclusive with MIR
NSERR_GSLB_ECS_IPV6_DISABLED	766	0x2fe	IPv6PT is disabled, DNS queries with an IPv6 ECS address will be dropped

ERROR CODE	ERROR CODE(DECIMAL)	ERROR CODE(HEX)	ERROR MESSAGE
NSERR_GSLB_ECS_LBMETHOD_NOT_SP	767	0x2ff	Enabling ECS without setting LB method to Static Proximity will not result in optimal responses for ECS-enabled queries
NSERR_REM_FAIL_NOMEM	1499	0x5DB	Removal is unsuccessful because of an internal error
NSERR_EN_AUTOSYNC_DEFAULT_PARTITION_WARNING	1500	0x5DC	Enabling automaticConfigSync parameter in default partition will have no effect

GSLB2 ERRORS

GSLB2 ERRORS

ERROR CODE	ERROR CODE(DECIMAL)	ERROR CODE(HEX)	ERROR MESSAGE
NSERR_DNS_VIEW_NOENT	3008	0xbc0	DNS View does not exist
NSERR_GSLB_LOC_NOMEM	3009	0xbc1	Memory allocation for location failed.
NSERR_DNS_PRIO_INVAL_VALUE	3010	0xbc2	Cannot bind DNS Policy with priority > 65535
NSERR_DNSPROF_EXISTS	3011	0xbc3	DNS Profile already exists.

ERROR CODE	ERROR CODE(DECIMAL)	ERROR CODE(HEX)	ERROR MESSAGE
NSERR_DNSPROF_DEF	3012	0xbc4	Cannot remove default DNS Profile.
NSERR_GSLB_LOC_EXISTS	3013	0xbc5	Custom Location Overlaps with an already existing location.
NSERR_GSLB_LOC_NOENT	3014	0xbc6	Custom Location does not exist.
NSERR_GSLB_NOMEM	3015	0xbc7	Not enough memory for GSLB operation.
NSERR_DNS_INT_POL	3016	0xbc8	Internal Policy is not allowed to bind if builtin parameter is not set with any value.
NSERR_GSLB_MEP	3017	0xbc9	Cannot enable MEP (Network Metric / Persistency Exchange) on a Child Site.
NSERR_IP_ROUTE_EXISTS	3018	0xbca	Host route enabled on this IP, cannot be used for GSLB Site IP.
NSERR_DNS_VIEW_EXISTS	3019	0xbcb	DNS View already exists
NSERR_NAME_VSVR_PERM	3020	0xbcc	Operation not permitted on a Vserver based Nameserver.

ERROR CODE	ERROR CODE(DECIMAL)	ERROR CODE(HEX)	ERROR MESSAGE
NSERR_DNS_POL_INTERNAL	3021	0xbcd	Cannot remove or unbind internal DNS policy
NSERR_DNS_NOMEM	3022	0xbce	Not enough memory for DNS Operation
NSERR_GSLB_NAPTR_SERVICE_SET	3023	0xbcf	Setting of these parameters on naptr gslb service is not allowed
NSERR_GSLB_NAPTR_REPLACEMENT_UNSET	3025	0xbd1	Unsetting of naptr replacement is not allowed if the gslb service is bound to gslb vserver of NAPTR dnsrectype.
NSERR_GSLBVIP_NAPTR_UNSUPPORTED_LBMETHODS	3026	0xbd2	GSLB vserver having NAPTR services can have only CUSTOM LOAD as lb method.
NSERR_GSLB_NAPTR_VSERVER_SET	3027	0xbd3	Setting of these parameters on natpr gslb vserver is not allowed.
NSERR_GSLB_NAPTR_DOMAIN_SIZE	3028	0xbd4	NAPTR domain cannot be created - name length exceeds maximum
NSERR_GSLB_NAPTR_DOMAIN_SYNTAX	3029	0xbd5	NAPTR domain cannot be created - name is invalid

ERROR CODE	ERROR CODE(DECIMAL)	ERROR CODE(HEX)	ERROR MESSAGE
NSERR_GSLB_NAPTR_REPLACEMENT_SUFFIX_SYNTAX	3030	0xbd6	Invalid NATPR Replacement Suffix
NSERR_GSLB_NON_NAPTR_SERVICE	3031	0xbd7	Operation permitted only on NAPTR GSLB service
NSERR_GSLB_USE_FORCESYNC	3032	0xbd8	Force sync option must be used in the sync gslb configcommand to synchronize the configuration to other GSLB sites, or manually add the requisite configuration to other sites.
NSERR_INVALID_WILDCARD_CONFIG	3033	0xbd9	Invalid Wildcard Configuration

HTML INJECTION ERRORS

HTML INJECTION ERRORS

ERROR CODE	ERROR CODE(DECIMAL)	ERROR CODE(HEX)	ERROR MESSAGE
NSERR_PREBODY_FILE_NOT_FOUND	3840	0xF00	Prebody file not found
NSERR_PREBODY_FILE_TOO_BIG	3841	0xF01	Prebody file too big
NSERR_PREBODY_FILE_READ_ERR	3842	0xF02	Prebody file read error

ERROR CODE	ERROR CODE(DECIMAL)	ERROR CODE(HEX)	ERROR MESSAGE
NSERR_POSTBODY_FILE_NOT_FOUND	3843	0xF03	Postbody file not found
NSERR_POSTBODY_FILE_TOO_BIG	3844	0xF04	Postbody file too big
NSERR_POSTBODY_FILE_READ_ERR	3845	0xF05	Postbody file read error
NSERR_HTML_INJECTION_VAR_EXISTS	3846	0xF06	HTML injection variable already exists
NSERR_HTML_INJECTION_VAR_NEXISTS	3847	0xF07	HTML injection variable does not exist
NSERR_HTML_INJECTION_VAR_NAME_TOO_LONG	3851	0xF0B	variable name too long
NSERR_HTML_INJECTION_INVALID_VAR_NAME	3852	0xF0C	Invalid variable name
NSERR_HTML_INJECTION_INCOMLETE_VAR_DEFINE	3853	0xF0D	Unpaired %% for variable name

ICA CLIENT VERSION ERRORS

ICA CLIENT VERSION ERRORS

ERROR CODE	ERROR CODE(DECIMAL)	ERROR CODE(HEX)	ERROR MESSAGE
NSERR_ICA_CLIENT_VERSION_UNKNOWN_CLIENT_TYPE	3536	0xdd0	Unknown ICA client type.
NSERR_ICA_CLIENT_VERSION_DEFAULT_WHITELISTING	3537	0xdd1	Failed to initialize the ICA whitelist: Switched to default static ICA whitelisting
NSERR_ICA_CLIENT_VERSION_NOT_FOUND_IN_EXCEPTION_LIST	3538	0xdd2	The ICA client version you are requesting to remove from the exception list is not present in the exception list.
NSERR_ICA_CLIENT_VERSION_INCORRECT_BASE_VERSION_FORMAT	3539	0xdd3	Invalid format for the ICA client base version entered.
NSERR_ICA_CLIENT_VERSION_INCORRECT_EXCEPTION_LIST_FORMAT	3540	0xdd4	Invalid format for the ICA exception list entered.
NSERR_ICA_CLIENT_VERSION_DUPLICATES_RECEIVED_TO_REMOVE	3541	0xdd5	Two or more of the same client versions entered in the ICA exception list.

ICA LATENCY PROFILE ERRORS

ICA LATENCY PROFILE ERRORS

ERROR CODE	ERROR CODE(DECIMAL)	ERROR CODE(HEX)	ERROR MESSAGE
NSERR_ICA_LATENCY_PROFILE_ACTIVE_CONNECTIONS	3542	0xdd6	Cannot remove the Latency profile as there are active ICA connections associated with this profile.

INAT ERRORS

INAT ERRORS

ERROR CODE	ERROR CODE(DECIMAL)	ERROR CODE(HEX)	ERROR MESSAGE
NSERR_PUBLICIP	3888	0xf30	Invalid Public IP address
NSERR_PRIVATEIP	3889	0xf31	Invalid Private IP address
NSERR_PROXYIP	3890	0xf32	Invalid Proxy IP address
NSERR_WILDCARDTCPVIP	3891	0xf33	TCPPROXY enabled when wildcard TCP Vserver already present
NSERR_WILDCARDANYVIP	3892	0xf34	INAT rule with wildcard ANY vserver, not allowed
NSERR_FTPVIPPRESENT	3893	0xf35	FTP enabled when FTP vserver already present

ERROR CODE	ERROR CODE(DECIMAL)	ERROR CODE(HEX)	ERROR MESSAGE
NSERR_V46_USIP	3894	0xf36	USIP is not allowed in mixed topology
NSERR_INCOMPATIBLE_IP	3895	0xf37	Private and Proxy IP should share the same IP Type
NSERR_ONLY_NAT46_SUPPORTED	3896	0xf38	Stateless mode only supported for IPv4->IPV6
NSERR_NAT46_PROXY_IP	3897	0xf39	Proxy IP is not supported in stateless mode
NSERR_NAT46_INVALID_PREFIXLEN	3898	0xf3A	Prefix length should be 32/40/48/56/64/96
NSERR_NAT46_INVALID_SETTING	3899	0xf3B	Invalid configuration for nat46
NSERR_NAT46_PRIVATEIP_EXISTS	3900	0xf3C	Private IP already configured in another nat46 rule
NSERR_TFTPVIPPRESENT	3901	0xf3d	TFTP enabled when TFTP vserver already present
NSERR_SETUNSETINAT_NOTSUPPORTED	3902	0xf3E	Setting and Unsetting mode of INAT not supported

INTERNAL NAMESERVER ERRORS

INTERNAL NAMESERVER ERRORS

ERROR CODE	ERROR CODE(DECIMAL)	ERROR CODE(HEX)	ERROR MESSAGE
NSERR_NAME_SVR_EXISTS	2561	0xa01	Name server already exists.
NSERR_NAME_SVR_IDNS_NEMPTY	2562	0xa02	Name servers already configured.
NSERR_NAME_SVR_IVALPOLICY	2563	0xa03	Invalid LB method for vserver-based name server.
NSERR_NAME_SVR_IVALPROTO	2564	0xa04	Invalid service type for vserver-based name server.
NSERR_NAME_SVR_ADD_NEXIST_VIP	2565	0xa05	vserver does not exist.
NSERR_NAME_SVR_NEXIST	2566	0xa06	Name server does not exist.
NSERR_NAME_SVR_IDNS_PERM	2567	0xa07	Operation on internal entity not permitted
NSERR_NAME_SVR_IP_EXISTS	2568	0xa08	Name server exists with this IP address
NSERR_NAME_SVR_TYPE_NEXIST_VIP	2569	0xa09	vserver of given type does not exist.
NSERR_NAME_SVR_INVAL_PROTOTYPE	2570	0xa0a	Invalid protocol type for local nameserver.
NSERR_NAME_SVR_DYN_VSRVR_EXISTS	2571	0xa0b	Dynamic vserver of same type exists.

INTF ERRORS

INTF ERRORS

ERROR CODE	ERROR CODE(DECIMAL)	ERROR CODE(HEX)	ERROR MESSAGE
NSERR_1GSFPSPEEDLIMIT	3968	0xf80	1G SFPs are restricted to speed 1000 or AUTO only
NSERR_1GSFPDUPLEXLIMIT	3969	0xf81	1G SFPs are restricted to duplex FULL or AUTO only
NSERR_1GSPEEDLIMIT	3970	0xf82	1G port can not be configured as 10G speed
NSERR_1GDUPLEXLIMIT	3971	0xf83	1G port can not be configured as HALF duplex
NSERR_10GSPEEDLIMIT	3972	0xf84	10G ports on this platform can only be configured as speed 1G, 10G or AUTO only
NSERR_10GDUPLEXLIMIT	3973	0xf85	10G ports on this platform can only be configured as duplex FULL or AUTO only
NSERR_10GAUTONEGLIMIT	3974	0xf86	10G ports on this platform do not support AUTONEG
NSERR_10GSPEEDONLY	3975	0xf87	10G ports on this platform are restricted to speed 10000 or AUTO only

ERROR CODE	ERROR CODE(DECIMAL)	ERROR CODE(HEX)	ERROR MESSAGE
NSERR_FULLDUPLEXONLY	3976	0xf88	These ports do not support HALF duplex mode
NSERR_AUTOINVALSPEED	3977	0xf89	Static CLAG does not support speed as AUTO
NSERR_CLAGSPEEDREQ	3978	0xf8a	Static CLAG explicitly requires speed to be configured other than AUTO
NSERR_THROUGHPUTLIMIT	3979	0xf8b	Throughput limit error. Possible values: 1G port - 1000Mbps, 10G port - 10000Mbps, 40G port - 40000Mbps, Link Aggregation - 80000Mbps
NSERR_BANDWIDTHLIMIT	3980	0xf8c	Bandwidth limit error. Possible values: 1G port - 1000Mbps, 10G port - 10000Mbps, 40G port - 40000Mbps, Link Aggregation - 80000Mbps
NSERR_10GDAC_SPEEDLIMIT	3981	0xf8d	DAC port speed must be configured as 10G or AUTO
NSERR_VIF_LACP_CAUTION	3982	0xf8e	Ensure that the physical interface is not shared with any other VMs.

ERROR CODE	ERROR CODE(DECIMAL)	ERROR CODE(HEX)	ERROR MESSAGE
NSERR_LA_SLAVE_LIMIT	3983	0xf8f	Trying to bind more interfaces than allowed limit
NSERR_SLAVE_MTU_MISMATCH	3987	0xf93	Interface MTU should be same as Channel MTU.
NSERR_BKPLANE_MTU_MISMATCH	3988	0xf94	Interface MTU should be default or same as Backplane MTU.
NSERR_MTU_NOT ALLOWED	3989	0xf95	MTU change is not allowed on this interface.
NSERR_BACKPLANE_MTU_LIMIT	3990	0xf96	Backplane Interface, Minimum user configured value for MTU is 1579.
NSERR_INTERFACE_SLAVE_MTU_LIMIT	3991	0xf97	Interface is a part of channel, MTU change is not allowed.
NSERR_CHANNEL_RESTRICT	447	0x1BF	LA channel configuration on VPX running on SDX should be done only from Service VM
NSERR_PTP_CLUS_L3	3992	0xf98	Cluster INC Mode Enabled, PTP state change is not allowed.
NSERR_40GDUPLEXLIMIT	3993	0xf99	40G ports on this platform can only be

ERROR CODE	ERROR CODE(DECIMAL)	ERROR CODE(HEX)	ERROR MESSAGE
			configured as duplex FULL or AUTO only
NSERR_40GSPEED ONLY	3994	0xf9a	40G ports on this platform are restricted to speed 40000 or AUTO only
NSERR_LRSET_FC _NOT_ALLOWED	3996	0xf9c	Flow Controle settings not allowed for link redundant set interface
NSERR_LRSET_LA CP_SET_NOT_ALL OWED	3997	0xf9d	Dynamic LA(LACP) settings are not allowed for link redundant set interface
NSERR_LRSET_LL DPSET_NOT_ALLO WED	3998	0xf9e	LLDP settings are not allowed for link redundant set interface
NSERR_LRSET_SP EED_NOT_ALLOWE D	3999	0xf9f	speed setting not allowed for link redundant set interface
NSERR_LRSET_PR IORITY_NOT_ALLO WED	4026	0xfba	lrset priority not allowed for link redundant set master interface
NSERR_LRSET_TH ROUGHPUT_NOT_ ALLOWED	4027	0xfbb	throughput setting not allowed for link redundant set master interface

ERROR CODE	ERROR CODE(DECIMAL)	ERROR CODE(HEX)	ERROR MESSAGE
NSERR_LRSET_BW_SET_NOT_ALLOWED	4028	0xfbc	Bandwidth settings are not allowed for link redundant set interface
NSERR_LRSET_LAG_TYPE_NOT_ALLOWED	4029	0xfbd	Lagtype setting not allowed for link redundant set interface
NSERR_LRSET_DUPLEX_NOT_ALLOWED	4030	0xfbe	Duplex setting not allowed for link redundant set interface
NSERR_LRSET_AUTO_NEG_NOT_ALLOWED	4031	0xfbf	Disabling autonegotiation not allowed for link redundant set interface
NSERR_JF_ESX_VERSION_SUPPORT	4070	0xfe6	Jumbo Frames are supported on ESX 5.5 version and onward
NSERR_TP_SETTINGS_NOT_ALLOWED	3385	0xd39	Trunk Port settings on this interface is not supported
NSERR_LACP_SETTINGS_ON_TP_NOT_ALLOWED	3386	0xd3a	LACP settings on Trunk Port is not allowed
NSERR_IP6_ADDRESS_SETTING_FAILED	3387	0xd3b	IPv6 Address can not be added because MTA table got exhausted

ERROR CODE	ERROR CODE(DECIMAL)	ERROR CODE(HEX)	ERROR MESSAGE
NSERR_50GSPEEDLIMIT	3388	0xd3c	50G ports on this platform can only be configured as speed 10G, 25G, 40G, 50G or AUTO
NSERR_100GSPEEDLIMIT	3389	0xd3d	100G ports on this platform can only be configured as speed 10G, 25G, 40G, 50G, 100G or AUTO

LB ERRORS

LB ERRORS

ERROR CODE	ERROR CODE(DECIMAL)	ERROR CODE(HEX)	ERROR MESSAGE
NSERR_SLESSLB_LBMETHOD_NOTSUPPORTED	1280	0x500	Incompatible LB method for sessionless vserver
NSERR_SLESSLB_PERSIST_NOTSUPPORTED	1281	0x501	Incompatible persistence method for sessionless vserver
NSERR_SLESSLB_TYPE_NOTSUPPORTED	1282	0x502	Sessionless vserver must be of type ANY, DNS or UDP
NSERR_SLESSLB_MODE_NOTSUPPORTED	1283	0x503	Sessionless vserver must have MAC or IPTUNNEL mode set
NSERR_SLESSLB_SVC_USIPNOTSET	1284	0x504	Service bound to a sessionless vserver

ERROR CODE	ERROR CODE(DECIMAL)	ERROR CODE(HEX)	ERROR MESSAGE
			must have UseSourceIP mode enabled
NSERR_WILDCARDVIP_LBMETHOD_INVAL	1286	0x506	Invalid LB method for a wildcard vserver
NSERR_WILDCARDVIP_PERSIST_INVAL	1287	0x507	Invalid persistence policy for a wildcard vserver
NSERR_LB_SO_THRESHOLD	1288	0x508	Spill-over threshold should be a non-zero value
NSERR_CONNFAILOVER_USIP	1289	0x509	Service must have Use Source IP option set in order to be bound to a connection failover enabled virtual server
NSERR_CONNFAILOVER_SERVICE	1290	0x50a	Connection failover can only be enabled on a virtual server of service type ANY
NSERR_LB_SO_DYNAMICCON_THRESHOLD	1291	0x50B	Spill-over threshold cannot be set for dynamic connection spill-over method
NSERR_LB_SO_ADDRVIP	1292	0x50C	DYNAMICCONNECTION spill-over cannot be set on a non-LB vserver

ERROR CODE	ERROR CODE(DECIMAL)	ERROR CODE(HEX)	ERROR MESSAGE
NSERR_CONNFAILOVER_NOTFOR_SLESS	1293	0x50d	Connection Failover is not supported for sessionless lb vserver\\n
NSERR_NEWSVC_PERCENT_CAPPED	1294	0x50e	Percentage value of requests capped at 100
NSERR_SIP_NOCALLID	1296	0x510	Missing Call-ID header field
NSERR_SIP_NOVIA	1297	0x511	Missing Via header field
NSERR_SIP_NOCSEQ	1298	0x512	Missing CSeq header field
NSERR_SIP_NOTO	1299	0x513	Missing To header field
NSERR_SIP_NOFROM	1300	0x514	Missing From header field
NSERR_SIP_NOMAX_FORWARDS	1301	0x515	Missing Max-Forwards header field
NSERR_SIP_SERVICE_UNAVAILABLE	1302	0x516	Service unavailable
NSERR_SVC_NOT_BOUND	1303	0x517	Service not bound to Vserver
NSERR_SVCTYPE_MISMATCH	1304	0x518	Service type mismatch between the vserver and service

ERROR CODE	ERROR CODE(DECIMAL)	ERROR CODE(HEX)	ERROR MESSAGE
NSERR_CH_CON_NOT_ALLOWED	1305	0x519	CONNECTION header cannot be set
NSERR_LBPROFILE_DBS	3572	0xdf4	DBSLB setting of LB profile cannot be changed with vservers associated with it
NSERR_LBPROFILE_EXIST	3573	0xdf5	LB profile already exists
NSERR_LBPROFILE_NOENT	3574	0xdf6	LB profile doesnt exist
NSERR_LBPROFILE_NOMEM	3575	0xdf7	Cannot allocate memory for LB profile
NSERR_LBPROFILE_VSERVERSET	3576	0xdf8	Cannot set vserver params which is associated with LB profile
NSERR_CH_INCOMPLETE_HDR	1312	0x520	Custom Header string should end with CRLF
NSERR_SPILLOVERDISABLED	1313	0x521	No spillover on the vserver due to max client setting 0on a bound service
NSERR_MAXTIMEEXCEEDED	1314	0x522	Maximum time value for interval or responsetimeout or downtime is exceeded, rounding to maximum

ERROR CODE	ERROR CODE(DECIMAL)	ERROR CODE(HEX)	ERROR MESSAGE
NSERR_SCINVALPROTO	1315	0x523	Cannot bind SC policy to non-HTTP service or vserver
NSERR_ROOTREC	1316	0x524	Atleast one nameserver record should exist for the root domain
NSERR_ROOT_GLUEREC	1317	0x525	Root domain nameserver address record should have atleast one ip address
NSERR_VIPINSERT_NOT_SUPPORTED	1356	0x54c	Vserver IP and port header insertion option not supported on this vserver.
NSERR_VSVR_ALRDY_BOUND	1318	0x526	This vserver is already bound to a WLM
NSERR_INVALID_WLM_BINDING	1319	0x527	Only LB vservers are allowed to bound to WLM
NSERR_VSVR_NOT_BOUND	1320	0x528	Vserver not bound to WLM
NSERR_WLM_EXISTS	1321	0x529	WLM already configured on this IP, port
NSERR_INVALID_TIMEOUT	1322	0x52a	Invalid persistence timeout: default timeout assigned

ERROR CODE	ERROR CODE(DECIMAL)	ERROR CODE(HEX)	ERROR MESSAGE
NSERR_CONNFAILOVER_INCMODE	1323	0x52b	Stateful connection failover cannot be enabled in INC mode
NSERR_CONNFAILOVER_TCPB	1324	0x52c	Stateful connection failover cannot be enabled on a vserver that binds services with TCP buffering enabled
NSERR_CONNFAILOVER_TCPB_SVC_BIND	1326	0x52e	Service with TCP buffering enabled cannot be bound to a vserver with stateful connection failover enabled
NSERR_CONNFAILOVER_TCPB_SVC_PARAM	1327	0x52f	TCP buffering cannot be enabled on a service bound to a vserver with stateful connection failover enabled
NSERR_CONNFAILOVER_SERVICE_STATEFULL	1328	0x530	Stateful connection failover cannot be enabled on a vserver of this service type
NSERR_CONNFAILOVER_SSL_SVC_BIND	1329	0x531	SSL TCP service cannot be bound to a TCP vserver with stateful connection failover enabled
NSERR_CONNFAILOVER_HA_INC_NODE	1330	0x532	Ha node with INC mode enabled cannot be added to the system that has

ERROR CODE	ERROR CODE(DECIMAL)	ERROR CODE(HEX)	ERROR MESSAGE
			vserver(s) with stateful connection failover enabled
NSERR_CONNFAILOVER_SSL_SVC	1331	0x533	Stateful connection failover cannot be enabled on a vserver that binds services of type SSL_TCP
NSERR_INVALID_PERSISTENCE	1332	0x534	Invalid Persistence type
NSERR_RTSPSESSION_INVALID	1333	0x535	Invalid RTSP session entry
NSERR_LB_METHOD_SAMEAS_BACKUP_METHOD	1334	0x536	Backup load balancing method cannot be the same as the primary load balancing method
NSERR_SERVER_EXIST	1335	0x537	Server already exists
NSERR_LB_SO_HEALTH	1336	0x538	HEALTH spill-over cannot be set on a non-LB vserver
NSERR_REQ_RULE_MISSING	1337	0x539	Request Rule is required, when Response rule is specified.
NSERR_CONNFAILOVER_BINDLO	1338	0x53a	Loopback service cannot be bound to vserver with connection failover enabled \\n

ERROR CODE	ERROR CODE(DECIMAL)	ERROR CODE(HEX)	ERROR MESSAGE
NSERR_CONNFAILOVER_SETWITHLO	1339	0x53b	Connection Failover cannot be set to STATEFUL/ STATELESS on vserver with bound loopback service\\n
NSERR_REDIRECT_URL_NOT_APPLICABLE	1340	0x53c	Redirect URL cannot be configured on this vserver.
NSERR_CONNFAILOVER_NOT_SUPPORTED	1342	0x53e	Connection Failover is not supported for pattern based vservers
NSERR_NOBIND_ONPUSH	1343	0x53f	Bindings not allowed on push vserver
NSERR_PUSH_VSVR	1344	0x540	Option cannot be set on push vserver
NSERR_ONLY_HTTPSSL	1345	0x541	Option can only be set on http/ ssl vserver
NSERR_PUSH_BIND_EXISTS	1346	0x542	A push vserver is already bound. Unbind before binding new vserver
NSERR_SET_PUSHVS_ONLY	1347	0x543	Push vserver should be of type push or ssl_push only
NSERR_SLESS_NO_SUPPORT	1348	0x544	Not supported for sessionless vservers
NSERR_LISTEN_POLICY_BIND	1351	0X547	Error in binding listen policy to vserver(normal

ERROR CODE	ERROR CODE(DECIMAL)	ERROR CODE(HEX)	ERROR MESSAGE
			vserver already present)
NSERR_NO_LISTEN_POLICY_FOR_DUMMYVS	1352	0x548	Cannot bind listen policy to dummy vservers
NSERR_LISTEN_POLICY_BIND_PRESENT	1353	0X549	Error in binding listen policy to vserver(vserver with new priority already present)
NSERR_NO_LISTEN_PRI	1354	0X54a	Error in binding listen priority to normal vserver
NSERR_NORMAL_VS_NONE_LISTEN POL	1355	0x54b	The vserver already has None Listen Policy\\n
NSERR_NO_LISPRI_FOR_NONE_LISPOL	1357	0x54d	None listenpolicy doesnot have listen priority\\n
NSERR_INVALID_POLICY_STRING	1358	0x54e	Invalid listenpolicy entered\\n
NSERR_NODATALENOFFFORVS	1359	0x54f	Dataoffset and datalength are not required for this vserver\\n
NSERR_LBGRPPERSISWITHRDP	1285	0x505	Persistency is not applicable for RDP vserver or LB group which has RDP vserver part of it
NSERR_CS_SOTHRESHOLD	1370	0x55a	Spillover threshold must be given with

ERROR CODE	ERROR CODE(DECIMAL)	ERROR CODE(HEX)	ERROR MESSAGE
			spillover method in Content Switching vserver
NSERR_INVALID_INSERTVSERVERIPPORT	1392	0x570	V6TOV4MAPPING is not applicable to ipv4 vserver
NSERR_LB_SERVER_NOTEXIST	1388	0x56c	The Target LB Vserver does not exist
NSERR_BOUNDTO_CSACT	1371	0x55b	The Vserver referred by Content Switching action cannot be removed
NSERR_BOUNDTO_CSPOLICY	1372	0x55c	The Content Switching Action bound to Content Switching Policy cannot be removed
NSERR_INVALID_MULTIPLEBIND	1373	0x55d	Multiple Bind not supported for Content Switching Advanced policies without action
NSERR_CSPOLICY_HASACTION	1374	0x55e	LB vserver/ gotoPriorityExpression/invoke not allowed during bind when content swtiching policy already has an action.
NSERR_CS_NOTSUPPORT_LB	1375	0x55f	The Protocol of given LB vserver is not supported by

ERROR CODE	ERROR CODE(DECIMAL)	ERROR CODE(HEX)	ERROR MESSAGE
			Content Switching action
NSERR_ACTION_NOTNOOP_CS	1386	0x56a	Do not provide gotoPriorityExpression/invoke because policy has action
NSERR_CSPOLICY_NOACTION	1387	0x56b	Either LB vserver or GotoPriorityExpression or Invoke must be provided while binding a Content Switching policy without action.
NSERR_CSPEPOLICY_SETNOACTION	1389	0x56d	Content Switching classic policy does not support action
NSERR_POLICY_USED_FOR_INVOKE	1390	0x56e	Do not give target LB beacuse the policy has been bound to multiple entities
NSERR_SPECIFY_BACKUP_NODE	1393	0x571	Please specify the backup node.
NSERR_SAME_NODES	1394	0x572	Owner and backup nodes need to different.
NSERR_INCOMPATIBLE_POLICY_SERVER_TYPE	1391	0x56f	Content switching vserver service type is not compatible with given policy expression
NSERR_PEPOLICY_BIND_NOTALLOWED	1395	0x573	Binding of classic content switching policy to content

ERROR CODE	ERROR CODE(DECIMAL)	ERROR CODE(HEX)	ERROR MESSAGE
			switching policylabel is not allowed
NSERR_TCPKA_IDLETIME_TOO_LOW	1396	0x574	Connection idle time cannot be lower than keep-alive probe interval
NSERR_OWNER_NODE_NOT_IN_CLUSTER	1397	0x575	Owner node is not in the cluster config or is NOT ACTIVE
NSERR_BACKUP_NODE_NOT_IN_CLUSTER	1398	0x576	Backup node is not in the cluster config or is NOT ACTIVE
NSERR_SPOTTED_VSERVER_NOT_SUPPORTED_IN_STANDALONE	1399	0x577	Spotted vserver configuration not supported in standalone or HA deplyoment
NSERR_SPOTTED_CONFIG_NO_MATCH	1400	0x578	Spotted config of the backup verser does not match the vserver
NSERR_OPTION_NOT_SUPPORTED	1401	0x579	Skip persistency is supported only for ANY and UDP service types
NSERR_OPTION_NOT_SUPPORTED_IP	1402	0x57a	BYPASS is not supported for IP based vservers
NSERR_OWNER_NODE_NOT_SET	1403	0x57b	Owner node should be specified for cluster

ERROR CODE	ERROR CODE(DECIMAL)	ERROR CODE(HEX)	ERROR MESSAGE
NSERR_INVALID_RENAME	1404	0x57c	Rename is supported only for Advanced policies
NSERR_INVAL_LOCAL_NAMSVR_TYPE	1405	0x57d	Only UDP based local nameservers are allowed
NSERR_INVALID_TFTPORT	1406	0x57e	Invalid port for tftp vserver
NSERR_NO_RENAME	1407	0x57f	No Rename Supported
NSERR_USE_BIND CR_CMD	88	0x58	Use Bind CR command to bind CR Vserver instead of Bind CS
NSERR_NOENT_AS_PROFILE	1408	0x580	No such Autoscale profile
NSERR_LBREFFERED_IN_CSBINDING	1409	0x581	Cannot remove LB Vserver which is reffered in CS policy bindings
NSERR_DEFAULTLB_NOT_BOUND	1410	0x582	default LB mentioned is not bound
NSERR_NG_BIND_ENTITY	1411	0x583	Error binding the node group to an entity
NSERR_ENTITY_BOUND_TO_ANOTHER_NG	1412	0x584	The specified entity is already bound to a different node group

ERROR CODE	ERROR CODE(DECIMAL)	ERROR CODE(HEX)	ERROR MESSAGE
NSERR_ENTITY_ALREADY_BOUND	1413	0x585	The specified entity is already bound to this node group
NSERR_ENTITY_NOT_BOUND_TO_NG	1414	0x586	The specified entity is not bound to any node group
NSERR_NG_NOT_FOUND	1415	0x587	Nodegroup is not found
NSERR_NG_HAS_BINDINGS	1416	0x588	Nodegroup cannot be deleted, it has entities bound to it
NSERR_VSERVER_CANNOT_BIND	1417	0x589	Vserver cannot bind to the nodegroup
NSERR_VSERVER_CANNOT_UNBIND	1418	0x58a	Vserver cannot unbind to the nodegroup
NSERR_NG_CANNOT_BE_STRICT	1419	0x58b	Nodegroup cannot be made strict as there are no nodes bound to it, but entities are bound
NSERR_NG_STRICT_AND_ZERODEPTH	1420	0x58c	Cannot bind an entity to a strict nodegroup with no nodes bound to it
NSERR_UNBINDING_LAST_NODE_FROM_STRICT_NG	1421	0x58d	Cannot unbind the last node from a strict nodegroup with entities bound to it
NSERR_NG_ALREADY_ADDED	1422	0x58e	Nodegroup is already added

ERROR CODE	ERROR CODE(DECIMAL)	ERROR CODE(HEX)	ERROR MESSAGE
NSERR_NG_ENTITY_NOMEM	1423	0x58f	No memory to establish entity to nodegroup association
NSERR_NG_MAX_CONFIGURED	1424	0x590	Cannot add nodegroup; maximum number of nodegroups already added
NSERR_INVAL_CS_ACTION	1425	0x591	Action given is not a CS action
NSERR_SET_CSACTION_DISALLOWED	1426	0x592	Set disallowed for CS action from Target LB/ targetVserver to Target Expression and vice versa
NSERR_RETAIN_VLAN	1427	0x593	macmoderetainvlan applicable only for wildcard(IP and port) macmode vserver. Disable macmoderetainvlan attribute to change the mode of vserver
NSERR_BIND_BDGGROUP_SYNCVLAN	1428	0x594	Sync vlan cannot be bound to bridgegroup
NSERR_DEL_SYNCVLAN	1429	0x595	Sync vlan cannot be deleted
NSERR_CSPEPOL_PARTITION_NOT_ALLOWED	1430	0x596	Current NON-default partition only

ERROR CODE	ERROR CODE(DECIMAL)	ERROR CODE(HEX)	ERROR MESSAGE
			supports Advanced Polices.
NSERR_LESSTHAN_PPE	1438	0x59e	Max/Min Reusepool connections should be greater than or equal to running Packet Engines.
NSERR_MCLIMITS	1295	0x50f	Dummy error to avoid issues when upgrade done from 9.3.
NSERR_SIP_BAD_REQUEST	1431	0x597	Invalid SIP request.
NSERR_SIP_NO_CONTENT_LEN	1432	0x598	No content length present.
NSERR_SIP_RET_MAX_INCOM_LEN	1433	0x599	SIP InComp max limit reached.
NSERR_SIP_RET_RETRY	1434	0x59a	SIP packet retry.
NSERR_SIP_RET_NO_MEM	1435	0x59b	SIP no memory available.
NSERR_SIP_RET_HEADER_NOT_COMPLETE	1436	0x59c	Incomplete SIP headers.
NSERR_SIP_RET_FAILURE	1437	0x59d	SIP operation failed.
NSERR_CS_GSLB_DEFAULT	1439	0x59f	Default vserver type mismatch
NSERR_CS_GSLB_DNSRT_MISMATCH	1306	0x51a	DNSrecordType of CS GSLB vserver

ERROR CODE	ERROR CODE(DECIMAL)	ERROR CODE(HEX)	ERROR MESSAGE
			and Target Vserver should be same.
NSERR_CS_GSLB_INVALID_TARGET	1307	0x51b	CS Vserver of type GSLB can only have GSLB vserver as Target.
NSERR_CS_GSLB_POLICY_NO_ACTION	1308	0x51c	CS Policy has no action or CS action doesnt have targetVserver
NSERR_GSLB_SVC_ALREADY_BOUND_TO_CS	1309	0x51d	GSLB service bound to targetVserver is already bound to CS vserver
NSERR_CS_POLICY_NO_RULE	1310	0x51e	Binding CS policy to CS GSLB vserver requires rule
NSERR_VIP_BACKUP_ISCSGSLB	1311	0x51f	CS GSLB vserver cannot be backup of a non- CS GSLB vserver
NSERR_VSR_BOUND	1325	0x52d	Service/servicegroup is bound to one or more vserver. Please unbind and retry the operation. Also check allowBoundSvcRemovaloption in lb parameter to change this behaviour.
NSERR_SG_AUTOSCALE_REQ_SRV	1504	0x5e0	Autoscale DNS must be enabled on servicegroup for

ERROR CODE	ERROR CODE(DECIMAL)	ERROR CODE(HEX)	ERROR MESSAGE
			binding server of SRV querytype
NSERR_SG_PORT_REQ	1505	0x5e1	Port is not specified
NSERR_SRV_PORT_NREQ	1506	0x5e2	Port should not be specified for DBS server of SRV querytype
NSERR_DBS_SPECIFIC_PARAMS	1507	0x5e3	This parameter can only be used for DBS type servers
NSERR_LW_CPX	1687	0x697	Operation not permitted

LB2 ERRORS

LB2 ERRORS

ERROR CODE	ERROR CODE(DECIMAL)	ERROR CODE(HEX)	ERROR MESSAGE
NSERR_ADDRPRESENT	1376	0x560	Address(s) is invalid or already present in this action, will be discarded.
NSERR_PERSISTENCE_PERSISTAVP_NOTDEFINED	1377	0x561	Persist Avp Number Must be defined For diameter Persistency
NSERR_MAX_DIAMETER_AVP_EXCEEDED	1378	0x562	Maximum number of diameter Persist Avp Codes exceeded

ERROR CODE	ERROR CODE(DECIMAL)	ERROR CODE(HEX)	ERROR MESSAGE
NSERR_AUTOSCALE_MIN_GT_MAX	1379	0x563	Minimum Autoscale member count exceeds the maximum value
NSERR_MEMBERPORT_REQD	1380	0x564	Member Port option has to be specified when autoScale option POLICY is chosen
NSERR_COOKIEINSERT_NOTSET	1381	0x565	Persistence type is not COOKIEINSERT.

LSN ERRORS

LSN ERRORS

ERROR CODE	ERROR CODE(DECIMAL)	ERROR CODE(HEX)	ERROR MESSAGE
NSERR_LSN_NOPOOL	3960	0xf78	No such pool
NSERR_LSN_NOTRANSPRFL	3961	0xf79	No such transportprofile
NSERR_LSN_NOAPPSPRFL	3866	0xf1a	No such applicationprofile
NSERR_LSN_POOL NOTBOUND	4052	0xfd4	Pool not bound to group
NSERR_LSN_PRFL NOTBOUND	4053	0xfd5	Profile not bound to group
NSERR_LSN_NOTREADY	4054	0xfd6	LSN not initialized. Please check

ERROR CODE	ERROR CODE(DECIMAL)	ERROR CODE(HEX)	ERROR MESSAGE
			memory allocated to LSN.
NSERR_LSN_MAXBINDINGS	4055	0xfd7	Maximum bindings allowed exceeded
NSERR_LSN_ACL_DETERMINISTICNAT	4056	0xfd8	Client with ACLs not allowed for deterministic nat
NSERR_LSN_ACLTYPE	4057	0xfd9	ACL with action other than ALLOW or establishedflag set
NSERR_LSN_ACLINUSE	4058	0xfda	ACL already bound to a LSN/RNAT rule
NSERR_64LIB_MEM_NOTSUFFICIENT	4059	0xfdb	Active memory less than minimum required for LSN and Subscriber Session Store
NSERR_LSN_NOBINDING	4060	0xfdc	No such binding
NSERR_LSN_NONETWORK	4061	0xfdd	No such network
NSERR_LSN_CLIENTGROUP	4062	0xfde	Please remove clients group first
NSERR_LSN_NOCLIENT	4063	0xfdf	No such client
NSERR_LSN_NOGLBLLOG	4072	0xfe8	Logging is disabled at Global level
NSERR_LSN_CLIENTINUSE	4080	0xff0	Client already present in a group

ERROR CODE	ERROR CODE(DECIMAL)	ERROR CODE(HEX)	ERROR MESSAGE
NSERR_LSN_PORTBLOCKSIZE	4081	0xff1	Portblocksize should be multiple of 256
NSERR_LSN_WRONG_PORTBLOCKSIZE	855	0x357	Portblocksize should not exceed 64512
NSERR_LSN_BINDINGEXIST	4082	0xff2	Binding already exists
NSERR_LSN_POOLGROUPTYPE	4083	0xff3	Both pool and group should be of same nat type
NSERR_LSN_POOLREUSE_DETERMINISTICNAT	4084	0xff4	Pool can be bound to only one group for determinstic nat
NSERR_LSN_PORT_APPSPROFEXIST	4085	0xff5	Profile already exist for one or more ports in this/bound group(s)
NSERR_LSN_TCPONLY	4086	0xff6	Allowed only for TCP profile
NSERR_LSN_MAPPINGFILTERINGTYPE	4087	0xff7	Filtering should be more or equally strict as mapping
NSERR_LSN_ICMPAPPSPROF_MAPPINGFILTERINGPORT	4088	0xff8	Port dependent Filtering or Mapping not allowed for ICMP
NSERR_LSN_ICMPAPPSPROF_PORT	4089	0xff9	Port bindings not allowed for ICMP applicationprofile
NSERR_LSN_STATIC_NOGROUP	4090	0xffa	Client ip not part of any active lsn group

ERROR CODE	ERROR CODE(DECIMAL)	ERROR CODE(HEX)	ERROR MESSAGE
NSERR_LSN_STATIC_INVPOOL	4091	0xffb	No natip available or natip given is invalid
NSERR_LSN_FINTIMGTSESSTIM	4092	0xffc	Finrsttimeout should not be more than session timeout
NSERR_LSN_SETPORTBLOCK_DETERMINISTICNAT	4093	0xffd	Port block size cannot be modified for deterministic nat
NSERR_LSN_DETERMINISTICNAT_SUBSCR_LIMIT	4094	0xffe	Maximum allowed subscribers for deterministic nat group exceeeded
NSERR_64LIB_MEM_INCOMPLETE	3943	0xf67	Active memory could be different from configured memory limit, Please check show extendedmemory parameter
NSERR_64LIB_MEM_EXCEEDSMAX	3944	0xf68	Configured memory cannot exceed maximum limit, Configured memory is set to maximum available limit
NSERR_64LIB_MEM_SIZE2M	3945	0xf69	Configured memory should be in multiples of 2MB
NSERR_LSN_IPINSERT	3957	0xf75	LSN IP addition failed

ERROR CODE	ERROR CODE(DECIMAL)	ERROR CODE(HEX)	ERROR MESSAGE
NSERR_LSN_ICMP_TIMEOUT_EXCEEDED	3958	0xf76	Maximum value of ICMP session timeout is 240 seconds
NSERR_LSN_TCP_UDP_TIMEOUT_EXCEEDED	3867	0xf1b	Maximum value of TCP/UDP session timeout is 600 seconds
NSERR_LSN_APPSPROF_PORTZERO	4073	0xfe9	Cannot bind/unbind port 0 to Application Profile
NSERR_LSN_MEM_DECR	3995	0xf9b	Active memory cannot be decreased without reboot
NSERR_LSN_NOSIPALGPRFL	3547	0xddb	No such sipalgprofile
NSERR_LSN_REM_APPPROFILE_ALGPROFILE_BOUND	3548	0xddc	Can not remove Appfprofile, SIP ALG profile and RTSP ALG Profile should be unbound fron the groups bound to the Appprofile
NSERR_LSN_UNBIND_APPPROFILE_ALGPROFILE_BOUND	3549	0xddd	Can not unbind to the Appprofile, SIP ALG profile and RTSP ALG Profile should be unbound fron the groups bound to the Appprofile
NSERR_LSN_NORTSPALGPRFL	4074	0xfea	No such rtspalgprofile

ERROR CODE	ERROR CODE(DECIMAL)	ERROR CODE(HEX)	ERROR MESSAGE
NSERR_LSN_SIPALGPROF_PORTZERO	4075	0xfeb	Cannot add port 0 to SIP ALG Profile
NSERR_LSN_RTSPALGPROF_PORTZERO	4076	0xfec	Cannot add port 0 to RTSP ALG Profile
NSERR_LSN_IP6PROF_DSLITE	4077	0xfed	With DS-Lite add only Network6
NSERR_LSN_IP6PROF_NAT64	4078	0xfee	With NAT64 add only IP6Prefix
NSERR_LSN_NETWORK6_EXISTS	3546	0xdda	network 6 based LSN client already exists for this network 6 address(Duplicate config)
NSERR_LSN_CLIENTNETWORK_INCORRECT_BINDING	3551	0xddf	Both Client and Network type MUST be of the same type, IPv4, IPv6 or DS-Lite Deterministic
NSERR_LSN_CLIENTGROUPTYPE	3553	0xde1	Both client and group should be of same type
NSERR_LSN_DSLITEPREFIXLEN	3554	0xde2	Network6 MUST have a prefix length 128 for a DS-Lite DETERMINISTIC type client or client associated with STATIC mapping

ERROR CODE	ERROR CODE(DECIMAL)	ERROR CODE(HEX)	ERROR MESSAGE
NSERR_LSN_NOIP6PROFILE	3555	0xde3	No such IP6 profile
NSERR_LSN_IP6GROUP	4079	0xfef	Please remove associated group first
NSERR_LSN_DSLITE_INVALIDIP6	3556	0xde4	With DS-Lite add only SNIP6.
NSERR_LSN_DSLITE_SNIP6_REQD	3557	0xde5	DS-Lite needs SNIP6 in relevent TDs.If not already present, add SNIP6 relevent to all tds belonging to a LSN group
NSERR_LSN_DUP_IP6PROFILE	3552	0xde0	Either IPv6 address or NAT Prefix already been used
NSERR_LSN_SIPALG_CANNOT_BIND	3882	0xf2a	Cannot bind SIPALG profile to the group, either the group is not in full cone mode or IP Pool pair is not enabled
NSERR_LSN_RTSPALG_CANNOT_BIND	3883	0xf2b	Cannot bind RTSP profile to the group, either the group is not in full cone mode or IP Pool pair is not enabled
NSERR_LSN_SIPALG_TRANSPORT_PROTOCOL_BINDING EXIST	3884	0xf2c	Can not bind more than one SIPALG profile of same

ERROR CODE	ERROR CODE(DECIMAL)	ERROR CODE(HEX)	ERROR MESSAGE
			transport protocol to the lsn group
NSERR_LSN_RTSPALG_TRANSPORT_PROTOCOL_BINDINGEXIST	3885	0xf2d	Can not bind more than one RTSPALG profile of same transport protocol to the lsn group
NSERR_LSN_CANNOT_UNBIND_APPPROFILE	3886	0xf2e	Can not unbind appprofile unless all SIP ALG Profiles, RTSP ALG Profiles and IPSEC ALG Profile are unbound from all the groups to which appsprofile is bound
NSERR_LSN_RTSPALGPROF_TRANSPORT_SUPPORTED	3887	0xf2f	RTSP ALG profile for only TCP is supported
NSERR_LSN_IPSECALG_CANNOT_BIND	3520	0xdc0	Cannot bind IPSECALG profile to the group, IP Pool pair is not enabled
NSERR_LSN_SIPALGPROF_PORT_NOT_ADDED	3353	0xd19	Atleast sipsrcportrange or sipdstportrange should be added with positive portrange
NSERR_LSN_GRP_SESS_LIMIT	852	0x354	Group session limit must be greater than the number of Packet Engines

ERROR CODE	ERROR CODE(DECIMAL)	ERROR CODE(HEX)	ERROR MESSAGE
NSERR_LSN_DETER_POOL	3903	0xf3f	Cannot enable portblock allocation on deterministic pool
NSERR_LSN_SIP_RTSP_ALG_MAPPING_FILTERING_IPPOOLING	853	0x355	Can not set appsprofile unless all SIP ALG profiles and all RTSP ALG profiles are unbound from all the groups to which appsprofile is bound
NSERR_LSN_PORTBLOCKSIZE_NOTALLOWED	856	0x358	Portblock size should not be changed to zero from nonzero and to nonzero from zero
NSERR_LSN_POOLGROUP_PORTBLOCK_MISMATCH	857	0x359	Pools portblock allocation and groups portblocksize fields are not compatable
NSERR_LSN_GRPID_PERM	3818	0xeea	GroupID should not be explicitly specified
NSERR_LSN_INVALID_DSLITE_PREFIX_LEN	3819	0xeeb	Invalid Ipv6 Prefix length for dslite client ipv6 network(only /128 prefix is allowed)
NSERR_LSN_NOLOGPRFL	3384	0xd38	No such log profile
NSERR_LSN_NOHTTPHDRLOGPRFL	3392	0xd40	No such HTTP header log profile

ERROR CODE	ERROR CODE(DECIMAL)	ERROR CODE(HEX)	ERROR MESSAGE
NSERR_LSN_DOHASYNC	3396	0xd44	Force HA Sync should be given to sync Primarys pre-flush operation LSN sessions
NSERR_LSN_STATICWILD_INV_PORT	3397	0xd45	invalid port for static nat wild card port config, subscr port and nat port must be *
NSERR_LSN_STATICWILD_INV_NATIP	3398	0xd46	NAT ip must be given for static nat wildcard port config
NSERR_LSN_STATICWILD_INV_DSTIP	3399	0xd47	Dest ip must not be given for static nat wildcard port config
NSERR_LSN_STATICWILD_INV_WILDPORT	3400	0xd48	wild card port(* or 65535) is valid only for trasport protocol type ALL(i.e only static nat wild card port config)
NSERR_LSN_STATICWILD_NATIP_INUSE	3401	0xd49	NAT ip is already in use for static nat wildcard port config.
NSERR_LSN_STATICWILD_NATIP_NO_FREE_PORTS	3402	0xd4a	NAT ip is not valid or not all ports on NAT ip are free, select different natip which has all ports free.
NSERR_LSN_NAT64_MAP_INS_FAIL	3406	0xd4e	Operation on nat64 prefix failed due to low system memory

ERROR CODE	ERROR CODE(DECIMAL)	ERROR CODE(HEX)	ERROR MESSAGE
NSERR_LSN_DSLITE_NO_FTP_CM	3577	0xdf9	FTP connection mirroring in LSN is not supported with DS-Lite nattype

MAP DOMAIN ERRORS

MAP DOMAIN ERRORS

ERROR CODE	ERROR CODE(DECIMAL)	ERROR CODE(HEX)	ERROR MESSAGE
NSERR_MAP_IPV6_PREFIX_INVALID	3529	0xdc9	MAP IPv6 Prefix is invalid, it must be global
NSERR_MAP_IPV6_PREFIX_REQ	3530	0xdca	MAP IPv6 Prefix is required
NSERR_MAP_PSID_INVALID	3531	0xdcb	MAP BMR Port Set ID is invalid
NSERR_MAP_PSID_OFFSET_INVALID	3532	0xdcc	MAP BMR Port Set ID offset is invalid
NSERR_MAP_IPV4_NW_REQ	3533	0xdcd	MAP BMR rule ipv4 network and netmask
NSERR_MAP_BMR_NW_MISSMATCH	3534	0xdce	MAP BMR rule and ipv4 network mismatch
NSERR_MAP_BMR_PSID_LEN_INVALID	3535	0xdcf	MAP BMR rule sharing ration is invalid it must be 1 to 16

ERROR CODE	ERROR CODE(DECIMAL)	ERROR CODE(HEX)	ERROR MESSAGE
NSERR_MAP_BMR_EABIT_LEN_INVALID	3543	0xdd7	MAP BMR rule EABIT length invalid
NSERR_MAP_NSB_CONSUMED	3544	0xdd8	NSB is consumed by MAP-T, error return to source or packet forwarded
NSERR_MAP_NO_CONFIG	3545	0xdd9	NO MAP T config, continue with other config

NAT64 ERRORS

NAT64 ERRORS

ERROR CODE	ERROR CODE(DECIMAL)	ERROR CODE(HEX)	ERROR MESSAGE
NSERR_NAT64_ACL_EXISTS	3959	0xf77	ACL is bound to another NAT64 rule

NET ERRORS

NET ERRORS

ERROR CODE	ERROR CODE(DECIMAL)	ERROR CODE(HEX)	ERROR MESSAGE
NSERR_ACL_TIMED_OUT	977	0x3d1	ACL timed out
NSERR_MSR_ADVT_NULL_RT	978	0x3d2	Cannot monitor or advertise NULL route

ERROR CODE	ERROR CODE(DECIMAL)	ERROR CODE(HEX)	ERROR MESSAGE
NSERR_DEF_RT_UNSET	979	0x3d3	Unset operation on default route not allowed
NSERR_ACTIVE_PERM_DIR_NULL_RT	981	0x3d5	Unset operation on active or permanent or direct or null routes not allowed
NSERR_DHCP_RT	982	0x3d6	Unset operation on DHCP route not allowed
NSERR_LB_RT	983	0x3d7	Unset operation on LB route not allowed
NSERR_MSR_STATIC_RT	984	0x3d8	Cannot set MSR on a non-static route
NSERR_ADV_DEF_RT	985	0x3d9	Advertise on default route not allowed
NSERR_VLAN_RT	986	0x3da	Cannot delete this route. Network is bound to VLAN
NSERR_PERM_RT	987	0x3db	A permanent route cannot be deleted
NSERR_NOENTRY_RT	988	0x3dc	Entry not found in the routing table
NSERR_DIR_RT	989	0x3dd	Cannot add a direct route
NSERR_LOOP_RT	990	0x3de	Cannot add a route in loopback network

ERROR CODE	ERROR CODE(DECIMAL)	ERROR CODE(HEX)	ERROR MESSAGE
NSERR_GW_NWID_BROADADDR	991	0x3df	Gateway cannot be network id or broadcast address
NSERR_MON_ADV_NULL_RT	992	0x3e0	Cannot set MSR or advertise on null gateway
NSERR_HA_PEER_IP	994	0x3e2	Cannot add IP address. This is peers NSIP
NSERR_NON_LOCAL_IP	995	0x3e3	IP address is already configured on a node in the cluster
NSERR_ADD_NSIP	996	0x3e4	Adding NSIP not allowed
NSERR_VRID_VIP_ONLY	997	0x3e5	VRID can be set only on VIP
NSERR_ARP_ICMP_SNIP_NSIP	998	0x3e6	ARP and ICMP cannot be disabled on SNIP/NSIP/MIP/CLIP
NSERR_IPADDR_DISABLE	999	0x3e7	IP address state of SNIP/NSIP/CLIP cannot be disabled
NSERR_NODE_NOEXIST	1000	0x3e8	Cluster node does not exist
NSERR_CLIP_SPOTTED_IP	1001	0x3e9	CLIP should be striped
NSERR_IP_MGMTACCESS	1002	0x3ea	Management Access cannot be enabled on this IP address

ERROR CODE	ERROR CODE(DECIMAL)	ERROR CODE(HEX)	ERROR MESSAGE
NSERR_IPTUN_MULTICAST	1003	0x3eb	Vlan on which to send the packet needs to be specified for multicast remote endpoints
NSERR_IPTUNNEL_INVALIDNETMASK	1004	0x3ec	IP tunnel netmask should be 255.255.255.255.
NSERR_IPTUNNEL_VLAN	1005	0x3ed	Vlan can be specified only for multicast remote endpoints
NSERR_IPTUN_INV_LIP	1006	0x3ee	Invalid local IP address. Either the local IP address is not configured or local IP address has to be one among SNIP|VIP|MIP
NSERR_VXLAN_LIP_NOT_SNIP	1007	0x3ef	Invalid local IP address. For VXLAN local IP address has to be SNIP
NSERR_INVALID_IPTUNNEL_NAME	1008	0x3f0	Invalid IPTunnel name
NSERR_IPTUNNEL_PREFXLEN_GT_128	1009	0x3f1	IP tunnel prefix length should not be more than 128
NSERR_TUN_PBR_DEP	1010	0x3f2	Cannot remove tunnel. A PBR is depending on this tunnel

ERROR CODE	ERROR CODE(DECIMAL)	ERROR CODE(HEX)	ERROR MESSAGE
NSERR_TUN_NOEXIST	1011	0x3f3	Tunnel does not exist
NSERR_INV_DROPFRAGCPUTHRESH	1012	0x3f4	Invalid dropFragCpuThreshold parameter. Should >=1 and <=100
NSERR_SRCIP_NOEXIST	1013	0x3f5	SrcIP does not exist
NSERR_TUN_ON_DEF_ROUTE	1014	0x3f6	Cannot add tunnel on default route
NSERR_VLAN_NOT_SNIP_MIP_VIP	1015	0x3f7	IP address has to be either SNIP, MIP or VIP
NSERR_ACL_PROTO_TCP	1016	0x3f8	Protocol should be TCP for ESTABLISHED flag to be set
NSERR_HA_VERSION_MISMATCH	1017	0x3f9	HA version mismatch between primary and secondary
NSERR_HA_SYNC_VLAN	1018	0x3fa	Syncvlan should be L2 VLAN. No IP address should be bound to this VLAN
NSERR_LOOP_FIS	1019	0x3fb	Loopback interface cannot be bound to Failover Interface Set (FIS)

ERROR CODE	ERROR CODE(DECIMAL)	ERROR CODE(HEX)	ERROR MESSAGE
NSERR_VLAN_LOOP	1020	0x3fc	Loopback interface cannot be unbound from default VLAN
NSERR_VLAN_TAGG_INTF_NOSUPP	1021	0x3fd	VLAN tagging is not supported on this platform/interface
NSERR_VLAN_BDG_NOTBOUND	1022	0x3fe	The VLAN is not bound to specified bridgegroup
NSERR_PERMANENT_ARP_ND6_ENTRY	1023	0x3ff	Cant remove permanent ARP/ND6 entry
NSERR_HA_VER_MISMATCH_PROP_SYNC	1447	0x5a7	HA version mismatch. Prop and sync not allowed
NSERR_VXLAN_TUNN_BOUND	1458	0x5b2	Cannot remove IPtunnel. One or more VXLAN is bound to this iptunnel.
NSERR_HOSTRT_UNSET	1489	0x5d1	Unset of OSPFArea and HostRtGw is allowed only for VIP
NSERR_BDGGRP_L3VLAN	2412	0x96c	L3VLAN cannot be bound to a bridgegroup
NSERR_BDGGRP_ROUTING	2505	0x9c9	Routing is enabled on the specified VLAN. Cannot bind this VLAN to the bridgegroup

ERROR CODE	ERROR CODE(DECIMAL)	ERROR CODE(HEX)	ERROR MESSAGE
NSERR_RM_RT_DIR_PERM_DHCP	2506	0x9ca	DIRECT/ PERMANENT/DHCP routes cannot be removed.
NSERR_ADD_IP_STRIPED_EXISTS	2507	0x9cb	Trying to add a striped IP address as a spotted IP.
NSERR_DIS_IP_NOT_USR_DEF	2508	0x9cc	Enable/Disbaling this IP address not allowed as VIP is not user-defined
NSERR_DIS_IP_NOT_VIP	2509	0x9cd	Enable/Disabling this IP address not allowed as it is not VIP
NSERR_DEL_NSIP	2510	0x9ce	Cannot delete NSIP
NSERR_DEL_DHCP_AQRD	2511	0x9cf	Cannot delete DHCP acquired IP address
NSERR_USR_DEF_NO	2618	0xa3a	Not a user-defined IP address
NSERR_TUNL_DEPENDENCY	2619	0xa3b	Existing tunnel depends on this IP address
NSERR_MGMT_NSIP_CLIP	2620	0xa3c	Management access on NSIP and CLIP should always be enabled
NSERR_INV_ICMP_RESP	2621	0xa3d	Invalid icmpResponse value

ERROR CODE	ERROR CODE(DECIMAL)	ERROR CODE(HEX)	ERROR MESSAGE
NSERR_INV_ARP_RESP	2622	0xa3e	Invalid arpResponse value
NSERR_INTERNAL_ERR_REGN	2623	0xa3f	Internal error while registration of service. Service name in use
NSERR_CANT_SET_IP_DEL_IN_PROG	2540	0x9ec	Delete IP address is in progress.
NSERR_INAT_USE_PROXY	2394	0x95a	Useproxyport cannot be disabled when either of publicIP or private IP is IPV6
NSERR_GSLB_ADNS_CONFGRD	2395	0x95b	GSLB/ADNS/ RADIUS is already configured on this IP address
NSERR_NW_ADDR_ZERO	2396	0x95c	Network address cannot be zero
NSERR_NW_ADDR_LOOP	2397	0x95d	Network address cannot be loopback
NSERR_UNABLE_DEFVLAN_INFO	2398	0x95e	Unable to get default VLAN info.
NSERR_HA_INC_SYNC	2399	0x95f	SyncVlan is not supported in HA-INC mode
NSERR_IP_NWADDR	2400	0x960	IPaddress cannot be network address
NSERR_IP_BCASTADDR	2401	0x961	IPaddress cannot be broadcast address

ERROR CODE	ERROR CODE(DECIMAL)	ERROR CODE(HEX)	ERROR MESSAGE
NSERR_ACL_BOUND_LSN	2402	0x962	This ACL is already bound to LSN
NSERR_SUBNET_BOUND_ANTHR_VLAN	2403	0x963	Either the subnet is not directly connected or subnet already bound to another VLAN
NSERR_BGP_AS_NO_EXIST	2404	0x964	BGP instance with given AS number does not exist.
NSERR_BGP_RUNNING	2405	0x965	Another BGP instance is already running.
NSERR_DEL_ADNS_IP	2406	0x966	Cannot delete this IP address as ADNS is configured on this.
NSERR_DEL_REF_IP	2407	0x967	Cannot delete this IP address as it is used in GSLB/IPSET/ Other entities
NSERR_SNMP_CANNOT_EN_ON_IP	2409	0x969	SNMP cannot be enabled on this IP address
NSERR_CANNOT_EN_ON_IPADDR	2410	0x96a	SNMP/TELNET/FTP/SSH/GUI/SECURE GUI/RESTRICTED-IP cannot be enabled on this IP address
NSERR_ACL_ESTABLISHED_SET	2414	0x96e	For RNAT/NAT64 ACL established flag cannot be set

ERROR CODE	ERROR CODE(DECIMAL)	ERROR CODE(HEX)	ERROR MESSAGE
NSERR_DEL_RADIUS_IP	2415	0x96f	Cannot delete this IP address as RADIUS is configured on this.
NSERR_CANNOT_EN_ON_IP	2416	0x970	TELNET/FTP/SSH/GUI/SECURE GUI/RESTRICTED-IP cannot be enabled on this IP address
NSERR_BFD_SESSION_CLEAR_INPROGRESS	2418	0x972	BFD session cannot be added now as old session clear is in progress
NSERR_BFD_SESSION_EXISTS_DIFF_FLAGS	2419	0x973	BFD session exists with different flags (SINGLEHOP AND MULTIHOP)
NSERR_CLASS_E	2426	0x97a	Class-E IPv4 and IPv6PT are mutually exclusive configs. Disable one to enable the other

NETBRIDGE ERRORS

NETBRIDGE ERRORS

ERROR CODE	ERROR CODE(DECIMAL)	ERROR CODE(HEX)	ERROR MESSAGE
NSERR_TNL_NETBRIDGED	3920	0xf50	Tunnel bound to netbridge
NSERR_TNL_GRE	3921	0xf51	Tunnel protocol is not GRE

ERROR CODE	ERROR CODE(DECIMAL)	ERROR CODE(HEX)	ERROR MESSAGE
NSERR_TNL_RIPMASK	3922	0xf52	Remote IP mask of the tunnel should be 255.255.255.255
NSERR_VLAN_NETBRIDGED	3923	0xf53	Vlan already bound to some netbridge
NSERR_MAX_NETBRIDGE_TUNNEL	3924	0xf54	reached maximum netbridge tunnel bindings
NSERR_TUN_BOUND	3925	0xf55	Tunnel already bound to this netbridge
NSERR_SUBNET_BOUND	3926	0xf56	Subnet already bound to this netbridge
NSERR_TUN_CONF_NETBRIDGE	3927	0xf57	Tunnel is not configured
NSERR_VLAN_CONF_NETBRIDGE	3928	0xf58	Vlan is not configured
NSERR_IP_INVAL_NETBRIDGE	3929	0xf59	IP is not binded to netbridge
NSERR_IPV6_INVAL_NETBRIDGE	3930	0xf5a	IPV6 is not binded to netbridge
NSERR_INV_NETBRIDGE_NAME	3931	0xf5b	Netbridge name should be >=1 and <=32
NSERR_GRE_IPSEC_LIP_NOT_GIVEN	3932	0xf5c	Local ip address has to be given for GRE, IPSEC and broadcast VXLAN tunnels

ERROR CODE	ERROR CODE(DECIMAL)	ERROR CODE(HEX)	ERROR MESSAGE
NSERR_NETBRIDGE_TUNNELBOUNDTOPBR	3933	0xf5d	Tunnel is a nexthop for PBR
NSERR_NETBRIDGE_UNSUPPORTEDTUNNELPAYLOAD	3934	0xf5e	GRE Payload should be ETHERNET or ETHERNETwithDOT1Q
NSERR_NETBRIDGE_ONEVLAN_ETHERNET	3935	0xf5f	Only one VLAN can be bound when GRE carries plain ETHERNET
NSERR_NETBRIDGE_TUNNELPAYLOAD_DIFFERENT	851	0x353	All bound tunnels should have same GRE payload

NETPROFILE ERRORS

NETPROFILE ERRORS

ERROR CODE	ERROR CODE(DECIMAL)	ERROR CODE(HEX)	ERROR MESSAGE
NSERR_IPSET_REF	3952	0xf70	IP set must be unbound from netprofile before it can be removed
NSERR_NETPROFILE_REF	3953	0xf71	Netprofile must be unbound before it can be removed
NSERR_IPSET_BOUND	3954	0xf72	IP set is already bound to the network profile

ERROR CODE	ERROR CODE(DECIMAL)	ERROR CODE(HEX)	ERROR MESSAGE
NSERR_NO_SUCH_IPSET	3955	0xf73	IP set does not exist
NSERR_NO_SUCH_NETPROF	3956	0xf74	Netprofile does not exist
NSERR_OVERLAPPING_PORTRANGE	3558	0xde6	This range is overlapping the existing portrange.
NSERR_NO_SUCH_PORTRANGE	3559	0xde7	Port range does not exist
NSERR_NO_SUCH_NATRULE	3560	0xde8	natRule does not exist
NSERR_NATRULE_EXIST	3561	0xde9	natrule already exist
NSERR_IP_VSERVER_IPSET_FREE_ERROR	3562	0xdea	Freeing Listening services failed. Please check failure counters

NITRO ERRORS

NITRO ERRORS

ERROR CODE	ERROR CODE(DECIMAL)	ERROR CODE(HEX)	ERROR MESSAGE
NSERR_NITRO_INVALID_OBJECTNAME	1232	0x4d0	Invalid object name
NSERR_NITRO_INVALID_JSON_INPUT	1233	0x4d1	Invalid JSON input

ERROR CODE	ERROR CODE(DECIMAL)	ERROR CODE(HEX)	ERROR MESSAGE
NSERR_NITRO_INVALID_JSON_DATATYPE	1234	0x4d2	Invalid JSON data type
NSERR_NITRO_INVALID_XML_INPUT	1235	0x4d3	Invalid XML input
NSERR_NITRO_INVALID_DATATYPE	1236	0x4d4	Invalid NITRO data type
NSERR_NITRO_INVALID_METHOD	1237	0x4d5	Invalid method name. It should be either post, put, get, stat or delete
NSERR_NITRO_PARSE_ERROR	1238	0x4d6	NITRO parse error
NSERR_NITRO_CMDEXEC_FAILED	1239	0x4d7	NITRO command Execution failed
NSERR_NITRO_INVALID_ACTION	1240	0x4d8	Invalid nitro action or operation
NSERR_NITRO_ROLLBACK_FAILED	1241	0x4d9	Rollback failed. Manually correct the configuration
NSERR_NITRO_ROLLBACK_NOT_SUPPORTED	1242	0x4da	Rollback not supported for this action
NSERR_NITRO_BULK_FAILED	1243	0x4db	Bulk operation failed
NSERR_NITRO_BASIC_AUTH_DISABLED	1244	0x4dc	Authentication Failed: Use of Basic Authentication is Disabled.

ERROR CODE	ERROR CODE(DECIMAL)	ERROR CODE(HEX)	ERROR MESSAGE
NSERR_NITRO_TECHSUPPORT_ARCHIVE_CREATION_FAILED	1246	0x4de	Creation of techsupport collector archive failed.
NSERR_NITRO_TECHSUPPORT_ARCHIVE_UPLOAD_FAILED	1247	0x4df	Failed to upload the techsupport collector archive.
NSERR_NSAPP_TEMPLATE_EXISTS	1248	0x4e0	Template already exists
NSERR_NSAPP_INVALID_TEMPLATE	1249	0x4e1	Invalid Template
NSERR_NSAPP_DIR_ERROR	1250	0x4e2	Couldnot open directory
NSERR_NSAPP_FILE_ERROR	1251	0x4e3	Can not open the file
NSERR_NSAPP_INVALID_APP_INPUT	1252	0x4e4	Invalid Application input
NSERR_NSAPP_EXCEEDED_FILELENGTH	1253	0x4e5	File length exceeded 256 characters
NSERR_NSAPP_ENDPOINT_INUSE	1254	0x4e6	Public endpoint in use
NSERR_NSAPP_PROTOCOL_MISMATCH	1255	0x4e7	Protocol mismatch with existing configuration
NSERR_NSAPP_INVALID_VARNAME	1256	0x4e8	Invalid variable name

ERROR CODE	ERROR CODE(DECIMAL)	ERROR CODE(HEX)	ERROR MESSAGE
NSERR_NSAPP_ZIP_FILE	1257	0x4e9	Error in zip format
NSERR_NSAPP_MAXEP_REACHED	1258	0x4ea	Only one endpoint can be configured
NSERR_NSAPP_NOT_EXIST	1259	0x4eb	Application does not exist
NSERR_NSAPP_INVALID_SERVICETYPE	1260	0x4ec	Invalid servicetype - HTTP/HTTPS/SSL are only allowed
NSERR_NSAPP_TEMPLATE_FORMAT_ERROR	1261	0x4ed	Template format error
NSERR_NSAPP_APP_WITHOUT_APPUNITS	1262	0x4ee	Application without appunits can not be exported
NSERR_NSAPP_SERVICRGROUP_EXISTS	1263	0x4ef	Service group with this name already exists
NSERR_NSAPP_VARIABLE_NOT_EXISTS	1264	0x4f0	Variables do not exist in this template file
NSERR_NITRO_CMD_NOTATTEMPTED	1265	0x4f1	command not attempted for configuration
NSERR_NITRO_INVALIDREQ	1266	0x4f2	Invalid nitro request

NNM ERRORS

NNM ERRORS

ERROR CODE	ERROR CODE(DECIMAL)	ERROR CODE(HEX)	ERROR MESSAGE
NSERR_NNM_FAILED	2528	0x9e0	General NNM failure
NSERR_NNM_CONN_CLEANED	2529	0x9e1	NNM connection was cleaned
NSERR_NNM_CONN_FAILED	2530	0x9e2	NNM connection failed
NSERR_NNM_NO_CONN	2531	0x9e3	NNM connection was not found
NSERR_NNM_CONN_DROPPED	2532	0x9e4	NNM connection dropped
NSERR_NNM_EINJECT	2533	0x9e5	NNM error injection
NSERR_NNM_VERSION_UNSUPPORTED	2534	0x9e6	NNM group payload version is unsupported
NSERR_NNM_TIMEOUT_MSG	2535	0x9e7	NNM timeout unable to send message
NSERR_NNM_TIMEOUT_ACK	2536	0x9e8	NNM timeout waiting for ACK
NSERR_NNM_MCAST	2537	0x9e9	NNM multicast/ broadcast failed
NSERR_NNM_PENDING	2538	0x9ea	NNM operation is in a pending state (e.g., ACK held)

PARTITION ERRORS

PARTITION ERRORS

ERROR CODE	ERROR CODE(DECIMAL)	ERROR CODE(HEX)	ERROR MESSAGE
NSERR_MAX_PARTITION	2752	0xac0	Maximum allowed partition limit reached
NSERR_PARTITION_NAME_EXIST	2753	0xac1	Partition name already exist
NSERR_NO_PARTITION	2755	0xac3	Partition does not exist
NSERR_DEL_PARTITION_BUSY	2756	0xac4	Partition is busy, cant delete
NSERR_DEL_PARTITION_INUSE	2757	0xac5	Partition is in use, cant delete
NSERR_NO_DEL_DFLT_PARTITION	2758	0xac6	Default Partition cant be delete
NSERR_RM_PARTITION_CONFIRM	2759	0xac7	Are you sure you want to remove partition (Y/N)? [N]:
NSERR_NOT_SUPPORTFORPARTITION	2760	0xac8	This feature is not supported on partition
NSERR_AP_CONFIG_NOT_ALLOWED	2761	0xac9	This configuration option is not supported on non-default partition
NSERR_VLAN_BDGGRP_PARTITION_EXIST	2762	0xaca	VLAN or VXLAN or BridgeGroup bound to a partition can not be deleted.

ERROR CODE	ERROR CODE(DECIMAL)	ERROR CODE(HEX)	ERROR MESSAGE
NSERR_AP_DEFVLAN	2763	0xacb	Default VLAN cannot be bound/unbound to/from a partition.
NSERR_VLAN_BDGGRP_PARTITION	2764	0xacc	The specified VLAN or VXLAN or BridgeGroup is bound to other partition.
NSERR_PQ_ARG_UNSUPPORTED_PARTITION	2765	0xacd	PQ feature is not supported in non-default partition.
NSERR_SC_ARG_UNSUPPORTED_PARTITION	2766	0xace	SC feature is not supported in non-default partition.
NSERR_MAX_PARTITIONS_ZEBOS	2767	0xacf	Maximum allowed partition limit for routing reached
NSERR_AP_SYNCVLAN	2769	0xad1	SYNC VLAN cannot be bound/unbound to/from a partition.
NSERR_AP_VLAN_BDG_EXIST	2770	0xad2	The specified VLAN cannot be bound to a partition because it is bound to a bridge group.
NSERR_AP_SDXMGMTVLAN	2771	0xad3	The SDX management VLAN cannot be bound/ unbound to/from a partition.

ERROR CODE	ERROR CODE(DECIMAL)	ERROR CODE(HEX)	ERROR MESSAGE
NSERR_AP_DEFAULT_PARTITION_CONFIG	2772	0xad4	This configuration is allowed only from the default partition.
NSERR_AP_MEM_DECR	2773	0xad5	Partitions memory limit cannot be reduced less than currently used memory
NSERR_AP_MISMATCH	2774	0xad6	The specified entity is configured in some other partition.
NSERR_AP_PVID_IF_NOT_ALLOWED	2775	0xad7	The specified VLAN cannot be configured as untagged member of interface because it is bound to partition.
NSERR_AP_VLAN_BINDING_NOT_ALLOWED	2776	0xad8	The specified VLAN cannot be bound to a partition because it is configured as untagged member of interface.
NSERR_AP_NOT_SUPPORTFORSHAREDVLAN	2777	0xad9	This feature is not supported on shared vlan.
NSERR_AP_SHAREDVLAN_CONFIG_NOT_ALLOWED_INCLUSTER	2778	0xada	Configuring vlan as shared is not supported on cluster.
NSERR_AP_SHAREDVLAN_BDG	2779	0xadb	Shared vlan cannot be bound/unbound

ERROR CODE	ERROR CODE(DECIMAL)	ERROR CODE(HEX)	ERROR MESSAGE
			to/from a bridge group.
NSERR_AP_SHAREDVLAN_TD	2780	0xadc	Shared vlan cannot be bound/unbound to/from a Traffic domain.
NSERR_AP_MCAST_PMAC	2781	0xadd	Partition MAC address cannot have multicast bit set.
NSERR_AP_VMAC_EXIST	2782	0xade	Another admin-partition with the same partition MAC address exist.
NSERR_AP_VMAC_BUSY	2783	0xadf	Partitions MAC address can not be unset when shared vlans are bound to it.
NSERR_AP_XS_CONN_FAILED	2784	0xae0	Unable to communicate with the XenStore.
NSERR_AP_XS_NO_VMAC	2785	0xae1	No partition MAC address is configured for the VPX on SDX.
NSERR_AP_XS_VMAC_NOTEXIST	2786	0xae2	The specified partition MAC address is not configured for the VPX on SDX.
NSERR_AP_XS_VMAC_EXHAUSTED	2787	0xae3	All partition MAC addresses configured for the

ERROR CODE	ERROR CODE(DECIMAL)	ERROR CODE(HEX)	ERROR MESSAGE
			VPX on SDX are in use.
NSERR_AP_ARP_VLAN_NOT_SPECIFIED	2788	0xae4	VLAN must be specified while adding ARP entry within partition.
NSERR_AP_SHAREDVLAN_PART_EXIST	2789	0xae5	This VLAN already bound to the specified partition.
NSERR_PARTITION_ID_EXIST	2790	0xae6	Partition-id is already in use by another partition.
NSERR_SYNCNSVLAN_AP	2791	0xae7	VLAN bound to Admin Partition cannot be configured as SYNC/NS VLAN.
NSERR_AP_NSVLAN	2792	0xae8	NSVLAN cannot be bound to Admin Partition.
NSERR_VXLAN_VLAN_DIFF_PARTITION	2799	0xaef	VXLAN and VLAN belong to different partitions.
NSERR_AP_VXLAN_TUNNEL_NOT_ALLOWED	2800	0xaf0	IPtunnel of type vxlan cannot be configured in non default partition
NSERR_VXLAN_CONFIG_INPART	2801	0xaf1	This vxlan config not supported in non-default partition.
NSERR_VXLAN_OTHER_PARTITION	2802	0xaf2	Specified vxlan is already bound to some other partition.

ERROR CODE	ERROR CODE(DECIMAL)	ERROR CODE(HEX)	ERROR MESSAGE
NSERR_VXLAN_VLAN_BIND	2803	0xaf3	When vxlan extends vlan, these operations should be done on vlan
NSERR_VXLAN_IP_BINDING_CURPART	2804	0xaf4	IP binding to vxlan should be done in vxlan partition alone
NSERR_VXLAN_BOUNDTO_PARTITION	2805	0xaf5	Specified vxlan is already bound to the partition.
NSERR_DYNROUTING_PARTITION	2806	0xaf6	Dynamic Routing should be enabled or disabled in vlan/ vxlan partition
NSERR_AP_SHAREDVLAN_VXLAN	2807	0xaf7	Shared vlan cannot be bound/unbound to/from a vxlan.
NSERR_AP_VLAN_VXLAN_EXIST	2808	0xaf8	The specified VLAN cannot be shared to a partition because it is extended by a vxlan .
NSERR_AP_STATIC_ARP_ND6_VXLAN_NOT_ALLOWED	2809	0xaf9	Static arp/nd6 entries cannot be added with vxlan in non default partition
NSERR_AP_MEM_MIN_LIMIT	2810	0xafa	Partitions memory limit cannot be less than 5MB
NSERR_AP_BDGGRP_BINDING_NOT_ALLOWED	2811	0xafb	The specified BridgeGroup cannot be bound to a partition because

ERROR CODE	ERROR CODE(DECIMAL)	ERROR CODE(HEX)	ERROR MESSAGE
			one of its member VLAN is configured as untagged member of interface.
NSERR_AP_NOT_LOCAL_ADMIN_PMAC	2812	0xafc	Local-admin bit must be set for Partition MAC address.
NSERR_AP_NOMEM	2813	0xafd	Cannot allocate memory. The total memory budgeted across all partitions will exceed the system capacity.

PBR ERRORS

PBR ERRORS

ERROR CODE	ERROR CODE(DECIMAL)	ERROR CODE(HEX)	ERROR MESSAGE
NSERR_PBR_NEXTHOP_NOTDIRECT	912	0x390	PBR Nexthop should be direct
NSERR_PBR_NOLOOPBACK	913	0x391	PBR cannot be configured on the loopback interface
NSERR_PBRDELERROR	914	0x392	This PBR has already been removed
NSERR_PBR_IPPIP_EXISTS	915	0x393	PBR with identical parameter specification already exists

ERROR CODE	ERROR CODE(DECIMAL)	ERROR CODE(HEX)	ERROR MESSAGE
NSERR_PBR_PRIORITY_EXISTS	916	0x394	PBR with this priority already exists
NSERR_PBRMODCFGINFO	918	0x396	PBR modified, use apply pbrsto commit this operation
NSERR_PBR_NEXTHOP_REQD	919	0x397	PBR Nexthop is required
NSERR_PBR_L2_CONFIG_INFO	920	0x398	L2 based PBRs work only for routed traffic
NSERR_PBRMODCFG_L2_INFO	921	0x399	L2 based PBRs work only for routed traffic: PBR modified, use apply pbrsto commit this operation
NSERR_PBR_NO_MONITOR_GATEWAY	922	0x39a	Monitor cant be configured when nexthop is configured as gateway name
NSERR_PBR_INVALID_IPORGATEWAY	923	0x39b	Invalid Nexthop IPaddress/Gateway name
NSERR_PBR_MAX_RULE_EXCEEDED	924	0x39c	Number of PBRs on the system exceeds Maximum
NSERR_PBR_NEXTHOP_TUNNEL_REQD	925	0x39d	PBR Nexthop or IpTunnel is required
NSERR_PBR_DENY_MSR	926	0x39e	MSR cannot be enabled on PBR with DENY config

ERROR CODE	ERROR CODE(DECIMAL)	ERROR CODE(HEX)	ERROR MESSAGE
NSERR_ACL_MAC_MASK_INVALID	927	0x39f	SrcMacMask should contain only 1/0s, use 0s for matching and 1s for not matching
NSERR_PBR_IPSEC_ALREADY_BOUND	1445	0x5a5	Use IKEv1 in IPSec profile. For IPSEC tunnels with IKEv2, Only one PBR is allowed to be bound to a tunnel
NSERR_PBR_REMOVED	1446	0x5a6	PBR is removed
NSERR_PBR_L2PARAM_GRE	1456	0x5b0	PBR is having L2PARAM. Cannot have GRE as nexthop.
NSERR_PBR_VXLAN	1457	0x5b1	VXLAN cannot be a nexthop.
NSERR_PBR_TUNNELBOUNDTONETBRIDGE	933	0x3a5	Tunnel is already bound to Netbridge
NSERR_PBR_SETNEXTHOP_TUNNEL	1217	0x4c1	Tunnel is bound to PBR. Cannot set NextHop

PBR6 ERRORS

PBR6 ERRORS

ERROR CODE	ERROR CODE(DECIMAL)	ERROR CODE(HEX)	ERROR MESSAGE
NSERR_PBR6_IPPIP_EXISTS	880	0x370	PBR6 with identical parameter specification already exists
NSERR_PBR6_DELERROR	881	0x371	PBR6 has already been removed
NSERR_PBR6_ADDERROR	882	0x372	Port can be specified only if protocol is TCP (6) or UDP (17)
NSERR_PBR6_PRIORITY_EXISTS	883	0x373	PBR6 with this priority already exists
NSERR_PBR6_ICMP_REQD	884	0x374	ICMPv6 type/code can be specified only if protocol is ICMPv6 (1)
NSERR_PBR6_UNSPECADDR	885	0x375	unspecified address (::) can not be configured in PBR6
NSERR_PBR6_MODCFGINFO	886	0x376	PBR6 modified, apply pbr6 to activate change
NSERR_PBR6_PREFIXLEN	887	0x377	Prefix length should not be configured in PBR6. (Use range instead)
NSERR_PBR6_SPECIFY_NEXTHOPVLAN	888	0x378	Nexthop VLAN should be specified with Link local gateway address

ERROR CODE	ERROR CODE(DECIMAL)	ERROR CODE(HEX)	ERROR MESSAGE
NSERR_PBR6_INVALID_GATEWAY	889	0x379	Invalid PBR6 Gateway
NSERR_PBR6_INVALID_NEXTHOP_VLAN	890	0x37a	Invalid Nexthop VLAN ID
NSERR_PBR6_MAX_RULE_EXCEEDED	891	0x37b	Number of PBR6s on the system exceeds Maximum

PCP ERRORS

PCP ERRORS

ERROR CODE	ERROR CODE(DECIMAL)	ERROR CODE(HEX)	ERROR MESSAGE
NSERR_PCP_PROFILE_BOUND	3521	0xdc1	Cannot remove PCP Profile. To remove PCP profile, first unset it from PCP servers
NSERR_PCPPROFILE_NOT_EXIST	3522	0xdc2	PCP profile does not exist, use show pcp profileto check available pcp profiles
NSERR_LSNPCP_BINDING_EXIST	3523	0xdc3	Please unbind existing PCP server from lsn group
NSERR_PCPSERVER_NOT_EXIST	3524	0xdc4	PCP server does not exist, use show pcp serverto check available pcp servers

ERROR CODE	ERROR CODE(DECIMAL)	ERROR CODE(HEX)	ERROR MESSAGE
NSERR_PCPSERVER_BIND_EXIST	3525	0xdc5	Can not Remove PCP server. Please unbind pcp server from all bindings
NSERR_PCPSERVER_INVALID_IP	3526	0xdc6	PCP Server IP Should be SNIP.
NSERR_PCP_LIFE_TIME	3527	0xdc7	PCP Min life time is greater than Max life time.
NSERR_PCPSERVER_LSNGROUP_TYPE_DIFF	3528	0xdc8	PCP Server being bound is incompatible with the LSN group.

PI ERRORS

PI ERRORS

ERROR CODE	ERROR CODE(DECIMAL)	ERROR CODE(HEX)	ERROR MESSAGE
NSERR_PI_LONGVSVRNAME	3040	0xbe0	The name of the label must be less than 32 characters
NSERR_PI_PRIORITY_EXISTS	3041	0xbe1	A policy is already bound to the specified priority
NSERR_PICON_GOTO_NOT_END	3042	0xbe2	GotoPriorityExpression for this policy has to be END
NSERR_INVAL_THRESHOLD	3043	0xbe3	Invalid threshold size value. Threshold size

ERROR CODE	ERROR CODE(DECIMAL)	ERROR CODE(HEX)	ERROR MESSAGE
			must be lesser than the sample size
NSERR_DEF_INVALID_UNBIND	3044	0xbe4	Default binding cannot be unbinded
NSERR_BAD_INIT_VALUE	3045	0xbe5	Invalid initial value
NSERR_INIT_VALUE_TOO_BIG	3046	0xbe6	Initial value is larger than the maximum allowed text length and the ifValueTooBig action is undef
NSERR_INIT_VALUE_TRUNCATED	3047	0xbe7	Initial value has been truncated to the maximum allowed text length because the ifValueTooBig action is truncate
NSERR_BAD_ASSIGN_LHS	3048	0xbe8	Invalid -var parameter: expect $variable-name for a ulong or text variable or $variable-name[key-expression] for a map variable.
NSERR_BAD_OP_FOR_TYPE	3049	0xbe9	Op is not allowed for the variable type.
NSERR_BAD_VAR_TYPE	3050	0xbea	Invalid type: expected one of ulong, text, or map.

ERROR CODE	ERROR CODE(DECIMAL)	ERROR CODE(HEX)	ERROR MESSAGE
NSERR_ASSIGNMENT_LOG_UNSUPPORTED	3051	0xbeb	Log action is not supported with assignment action
NSERR_UNDEF_EXCEPTION	3052	0xbec	Undef exception raised.
NSERR_NO_VARIABLE_VALUE	3053	0xbed	No value exists for the variable and key.
NSERR_Q_S_DEPRECATED	3054	0xbee	The Q and S prefixes are deprecated - use HTTP.REQ and HTTP.RES instead
NSERR_UNSUPPORTED_VARIABLE_USE	3055	0xbef	Variables are not supported for use with DNS.
NSERR_BAD_KEY_TYPE	3056	0xbf0	Invalid key type: expected type text
NSERR_BAD_SINGLETON_VAR	3057	0xbf1	Not a singleton variable
NSERR_BAD_MAP_VAR	3058	0xbf2	Not a map variable
NSERR_UNKNOWN_VARIABLE	3059	0xbf3	Variable is not configured.
NSERR_MAP_WITHOUT_KEY	3060	0xbf4	Map variable used without a key.
NSERR_SINGLETON_WITH_KEY	3061	0xbf5	Singleton variable used with a key.
NSERR_BAD_TEXT_MAX_LENGTH	3062	0xbf6	Invalid text type: expected text(max-length), where max-length is an unsigned

ERROR CODE	ERROR CODE(DECIMAL)	ERROR CODE(HEX)	ERROR MESSAGE
			integer >= 1 and <= 64000.
NSERR_BAD_MAP_TYPE_SYNTAX	3063	0xbf7	Invalid map type: expected map(key-type, value-type, max-entries), where max-entries is an unsigned long value <= 18446744073709551615 and greater than 0.
NSERR_BAD_MAP_KEY_TYPE	3064	0xbf8	Map key type must be text.
NSERR_BAD_MAP_VALUE_TYPE	3065	0xbf9	Map value type must be ulong or text.
NSERR_BAD_KEY_VALUE_LENGTH	3066	0xbfa	Sum of the maximum map key and value lengths must be <= 64000.
NSERR_BAD_MAP_SET_TYPE	3067	0xbfb	Invalid type parameter for set.
NSERR_TRANSACTION_MAPS_NOT_SUPPORTED	3068	0xbfc	Maps with transaction scope are not supported.
NSERR_FILTER_DEPRECATED	3069	0xbfd	Content Filtering is deprecated - use Responder (for ERRORCODE, or DROP or RESET on the request side), Rewrite (for ADD or CORRUPT, or DROP or RESET on the

ERROR CODE	ERROR CODE(DECIMAL)	ERROR CODE(HEX)	ERROR MESSAGE
			response side), or Content Switching (for FORWARD) instead
NSERR_TRANSACTION_EXPIRE_NOT_SUPPORTED	3070	0xbfe	Expiration with transaction scope variables is not supported.
NSERR_INVOKE_NOT_SUPPORTED	3071	0xbff	Policy does not support invoke.
NSERR_INVALPIEXPR	3073	0xc01	Invalid expression
NSERR_INVALINTOP	3074	0xc02	Expecting numeric arguments in this context for -, +, *or /
NSERR_INVALCOMPARE	3075	0xc03	Invalid arguments to compare operation
NSERR_INVALBOOLOP	3076	0xc04	The !, || and ;;operators can have only boolean argument(s)
NSERR_ARGNEG	3077	0xc05	Argument cannot be negative
NSERR_ARGEXCEED	3078	0xc06	Argument exceeds maximum allowed value
NSERR_ARGZERO	3079	0xc07	Operand or function argument should be non-zero
NSERR_MAXHEADER	3080	0xc08	Number of custom header exceeds limit

ERROR CODE	ERROR CODE(DECIMAL)	ERROR CODE(HEX)	ERROR MESSAGE
NSERR_INVALPICEXPR	3081	0xc09	Expression syntax error
NSERR_CEXPRLIMIT	3082	0xc0a	Expression too long
NSERR_EXPRLIMIT	3083	0xc0b	Expression too long
NSERR_SEXPRLIMIT	3084	0xc0c	String expression too long
NSERR_INVALPISEXPR	3085	0xc0d	String expression syntax error
NSERR_NOBIDI	3086	0xc0e	Bi-directional expression not allowed
NSERR_NOLAB	3087	0xc0f	Policy Label does not exist
NSERR_INTERNAL_LABEL_RM	3088	0xc10	Default Policy labels cannot be removed
NSERR_POLICY_FLOWTYPE_NONE	3089	0xc11	Policy should either be a request or response policy
NSERR_INVAL_PRIORITY	3090	0xc12	Priority is mandatory for advanced expressions and should be in the range from 1 to 2147483647
NSERR_NUMERICEXPR	3091	0xc13	Expression should evaluate to numeric value

ERROR CODE	ERROR CODE(DECIMAL)	ERROR CODE(HEX)	ERROR MESSAGE
NSERR_BOOLEXPR	3092	0xc14	Expression should evaluate to true or false
NSERR_NOTBOUND	3093	0xc15	Cannot unbind a policy that is not bound
NSERR_POLICY_INUSE	3094	0xc16	Policy name already in use
NSERR_POLICY_SET_NOTALLOWED	3095	0xc17	Invalid rule/action for bound policy
NSERR_INVALID_BIND	3096	0xc18	Policy cannot be bound to specified policy label
NSERR_USE_INVOKE_RESULT	3097	0xc19	USE_INVOCATION_RESULT can be used only with invoke
NSERR_INVAL_EXPR_DATA_TYPE	3098	0xc1a	Invalid expression data type
NSERR_EXPR_CONST_ONLY	3099	0xc1b	Only constant parameters are allowed for function
NSERR_ALT_EXPR_STRING_ONLY	3100	0xc1c	Only string expressions are allowed for ALT
NSERR_ACTION_DEF_SET_INVAL	3101	0xc1d	Default action cannot be modified
NSERR_POLICY_DEF_SET_INVAL	3102	0xc1e	Default policy cannot be modified

ERROR CODE	ERROR CODE(DECIMAL)	ERROR CODE(HEX)	ERROR MESSAGE
NSERR_POLICY_LABEL_DEF_SET_INVAL	3103	0xc1f	Default policylabel cannot be modified
NSERR_INVAL_UNBIND	3104	0xc20	Policy not bound to specified policy label
NSERR_REGEX_INVALID	3105	0xc21	Invalid regular expression
NSERR_REGEX_BACKREF	3106	0xc22	Backreference in regular expression
NSERR_REGEX_RECURSIVE	3107	0xc23	Recursive regular expression present
NSERR_PIXL_EXPR_UNSAFE	3108	0xc24	Input expression is unsafe
NSERR_PI_ACTION_MAX_REF_REACHED	3109	0xc25	Cannot create policy. Binding limit for action reached (65536)
NSERR_PI_INVALID_UNSET	3110	0xc26	Cannot unset rule or action. No changes done
NSERR_EMPTY_STRING	3111	0xc27	Zero length string not allowed
NSERR_INVAL_BTW_ARGS	3112	0xc28	Arguments to BETWEEN are incorrect
NSERR_INVAL_ARG_SPECIFIED	3113	0xc29	GotoPriorityExpression, flowtype and invoke apply only to rewrite, responder and cache policies

ERROR CODE	ERROR CODE(DECIMAL)	ERROR CODE(HEX)	ERROR MESSAGE
NSERR_INVAL_ARG_CSPL	3854	0xf0e	Target vserver cannot be specified along with GotoPriorityExpression and invoke.
NSERR_FLOWTYPE_NOT_APPLICABLE	3114	0xc2a	Flowtype applies only to rewrite, responder and cache policies
NSERR_INVAL_ARG_SPECIFIED_CS	3115	0xc2b	Flowtype and invoke apply only to Advanced policies.
NSERR_INVAL_GOTOEXPR_CS	3116	0xc2c	GotoPriorityExpression applies only to Advanced policies
NSERR_REBIND_FAILED	3117	0xc2d	Rebinding of policy with new bind parameters failed. The policy is unbound.
NSERR_SECOND_BIND	3118	0xc2e	A policy can be bound only once
NSERR_NONHTTP_VS	3119	0xc2f	Specified policy can be bound only to HTTP/SSL/PROXY vserver
NSERR_PI_LEX_FAILURE	3344	0xd10	Error in parsing
NSERR_PI_INVALID NAME	3856	0xF10	Invalid policy entity name: names must begin with an ASCII alphabetic character or underscore and

ERROR CODE	ERROR CODE(DECIMAL)	ERROR CODE(HEX)	ERROR MESSAGE
			must contain only ASCII alphanumerics or underscores; words reserved for policy use may not be used
NSERR_PI_RST_CONN	3855	0xF0F	Message processing reset connection
NSERR_PI_EXPR_CSMSG	3857	0xF11	Advanced expression cannot have client security message
NSERR_INVAL_XPATH_EXPR	3858	0xf12	XPath Expression Compilation Failed
NSERR_XPATH_EXPR_NOT_SUPPORTED	3859	0xf13	Unsupported XPath
NSERR_TARGET_INVAL	3860	0xf14	Target Vserver cannot be specified for this policy
NSERR_NON_RESP_HTTP_VS_LIST	3861	0xf15	Specified policy can be bound only to HTTP/SSL/TCP/SSL_BRIDGE/FTP/DNS_TCP/RTSP/SSL_TCP/NNTP/MSSQL/MySQL/ORACLE/DIAMETER/RADIUS/ANY/PROXY vserver

ERROR CODE	ERROR CODE(DECIMAL)	ERROR CODE(HEX)	ERROR MESSAGE
NSERR_INVAL_VSERVER_BIND	3862	0xf16	Policy can be bound only to LB/CS vserver types
NSERR_APP_FREED_NSB	3863	0xf17	Application reset/ drop connection and freed nsb
NSERR_MULT_BIND_INVAL	3864	0xf18	CVPN Policies cannot be bound to multiple entities
NSERR_CLASSIC_POLICY_ALREADY_BOUND	3865	0xF19	Cannot bind an advanced policy to a virtual server to which a classic policy is already bound
NSERR_ADVANCED_POLICY_ALREADY_BOUND	3872	0xF20	Cannot bind a classic policy to a virtual server to which an advanced policy is already bound
NSERR_MULT_UNBIND	3873	0xF21	Policy is bound to multiple global bindpoints. Please specify the appropriate bindpoint to unbind it.
NSERR_UNKNOWN_CERTKEY	3874	0xF22	The specified certkey object has not been configured
NSERR_NO_RSA_PUBLIC_KEY	3875	0xF23	The specified certkey object does not include a certificate

ERROR CODE	ERROR CODE(DECIMAL)	ERROR CODE(HEX)	ERROR MESSAGE
			with an RSA public key
NSERR_NO_RSA_PRIVATE_KEY	3876	0xF24	The specified certkey object does not include an RSA private key
NSERR_CERTKEY_NAME_TOO_LONG	3877	0xF25	The certkey name exceeds the maximum 31 characters
NSERR_INVALID_UTF_8	3878	0xF26	Invalid UTF-8 string
NSERR_UNRECOG_CHARSET	3879	0xF27	Unrecognized character set
NSERR_INCOMPATIBLE_NAME_EXPR_CHANGE	3880	0xF28	Incompatible named expression change for in-use expression.
NSERR_HEADER_EXPR_ONLY	3881	0xF29	Only classic expressions for headers allowed in EVAL_CLASSIC_EXPR
NSERR_RW_NOT_SUPPORTED_VS	3904	0xF40	Specified policy can be bound only to HTTP/SSL/TCP/SSL_TCP/DIAMETER/SSL_DIAMTER/SIP_UDP/RADIUS/PROXY vserver

ERROR CODE	ERROR CODE(DECIMAL)	ERROR CODE(HEX)	ERROR MESSAGE
NSERR_INVALID_KEYVALUE	3905	0xf41	The keyvalue argument is incorrect.
NSERR_INVALID_RESOURCE_NAME	3906	0xF42	Resource of the given name does not exist
NSERR_INVALID_ACTION	3907	0xF43	Invalid action
NSERR_ASCII_ONLY_FUNCTION	3908	0xF44	Advanced expression function does not accept non-ASCII arguments
NSERR_INVALID_ENUM_VALUE	3909	0xF45	Invalid enumeration value
NSERR_INVALID_BINDPOINT	3910	0xF46	Invalid bind point.
NSERR_NO_FIPS_FOR_POLICY_CRYPTO	3911	0xF47	A FIPS key cannot be used for policy crypto functions.
NSERR_NO_CONFIGDB_DATA_SELECTED	3912	0xF48	No data selected from the configuration database.
NSERR_HEADER_NAME_LENGTH_LIMIT	3913	0xF49	Custom header name is too long.
NSERR_UNSUPPORTED_CVPN_SEARCH	3914	0xF4A	Patset, dataset, and xpath not supported for clientless VPN search.

ERROR CODE	ERROR CODE(DECIMAL)	ERROR CODE(HEX)	ERROR MESSAGE
NSERR_INVALID_HEADER_NAME	3915	0xF4B	Header name contains invalid characters (one or more of ( ) < > @ , ; : \\\\ \\" / [ ] ? = { } and space) as per the HTTP spec RFC 2616. This is allowed to for non-conformant HTTP messages, but may not be what you intended.
NSERR_HEADER_NAME_TOO_LONG	3916	0xF4c	Header name is too long (maximum of 32 characters allowed).
NSERR_CSACTION_NOT_SUPPORTED	3918	0xF4E	Content Switching actions are not suported for classic content switching policies.
NSERR_PI_EXPR_TOO_BIG	3919	0xF4F	Expression too long - maximum length is 8191
NSERR_PI_INVALID_FUNCTION_NAME	3946	0xF6A	Cannot use Advanced Expression reserved word as a function name.
NSERR_PI_FUNCTION_NAME_EXISTS	3947	0xF6B	Function with the same name already exists.

ERROR CODE	ERROR CODE(DECIMAL)	ERROR CODE(HEX)	ERROR MESSAGE
NSERR_EXTENSION_INVALID_UPDATE_NAME	3948	0xF6C	Add the extension before updating.
NSERR_PI_INVALID_EXTENSION_NAME	3949	0xF6D	Extension is not added.
NSERR_EXTENSION_FUNCTION_PROTOTYPE_CHANGED	3950	0xF6E	Extension has different prototype of a function than that which is in use.
NSERR_EXTENSION_FUNCTION_UPDATE_PROBLEM	3951	0xF6F	Problem in updating function that is in use.
NSERR_EXTENSION_COMPILE_FAILURE	3962	0xF7A	Extension compilation error.
NSERR_EXTENSION_LOAD_FAILURE	3963	0xF7B	Extension loading error.
NSERR_PI_USED_FUNCTION_REMOVED	3964	0xF7C	Update attempted to remove a function that is in use.
NSERR_EXTENSION_FUNCTION_MULTIPLE_DEFINITIONS	3965	0xF7D	Extension has multiple definitions of a function.
NSERR_EXTENSION_INVALID_FUNCTION_PROTOTYPE	3966	0xF7E	Extension functions prototype is invalid.
NSERR_EXTENSION_MAX_FUNCTION_ARGUMENT	3967	0xF7F	Maximum number of function arguments can be 3.

ERROR CODE	ERROR CODE(DECIMAL)	ERROR CODE(HEX)	ERROR MESSAGE
NSERR_BAD_EXTENSION_TRACE_NAME	4038	0xFC6	Function or variable name can contain only alphanumerics and underscores and cannot begin with a digit.
NSERR_BAD_EXTENSION_TRACE_ARRAYS	3024	0xBD0	traceFunctions or traceVariables cannot be specified with trace off.
NSERR_PI_INVALID_GLOBAL_BINDTYPE	3828	0xEF4	Invalid global bindtype.
NSERR_INVALID_POLICY_GLOBAL_BINDING	3829	0xEF5	Policy cannot be bound/unbound to/ from the given global bindtype.
NSERR_INVALID_PI_ANNOTATE	1465	0x5b9	Invalid Advanced Expression annotation value
NSERR_INVALID_PI_ANNOTATE_LOGINSCHEMA	1466	0x5ba	Invalid loginSchema corresponding to Advanced Expression annotation value
NSERR_ANGLE_BRACKETS_IN_TAGNAME	3830	0xEF6	Invalid syntax, angle brackets in tag specified
NSERR_INVALID_ENCRYPTION_KEY_ARGUMENT	3831	0xEF7	Invalid encryptionKey argument

ERROR CODE	ERROR CODE(DECIMAL)	ERROR CODE(HEX)	ERROR MESSAGE
NSERR_INVALID_HMAC_KEY_ARGUMENT	3832	0xEF8	Invalid hmacKey argument
NSERR_ENCRYPTION_KEY_NOTPRESENT	3833	0xEF9	Encryption key does not exist.
NSERR_HMAC_KEY_NOTPRESENT	3834	0xEFA	HMAC key does not exist.
NSERR_WARN_SHORT_HMAC_KEY	3835	0xEFB	Recommend that the HMAC keyValue be at least as long as the digest blocksize
NSERR_MULT_TYPE_UNBIND	3836	0xEFC	Policy is bound to multiple bindpoints. Please specify the appropriate bindpoint to unbind it.
NSERR_EMPTY_CERTKEY	3837	0xEFD	The specified certkey name is empty or NULL.
NSERR_NON_PROXY_HTTP_VS	3838	0xEFE	Specified policy can be bound only to PROXY/HTTP/SSL vserver
NSERR_EXTENSION_HAS_GENERIC_MBLB_API_NOT_FUNCTION_API	3839	0xEFF	Extension code already used for an Advanced Expression extension function cannot be used for a user protocol implementation.

ERROR CODE	ERROR CODE(DECIMAL)	ERROR CODE(HEX)	ERROR MESSAGE
NSERR_EXTENSION_HAS_FUNCTION_API_NOT_GENERIC_MBLB_API	3424	0xD60	Extension code already used for a user protocol implementation cannot be used for an Advanced Expression extension function.
NSERR_EXTENSION_INCORRECT_USER_PARAM_VALUE	3435	0xD6B	Incorrect parameter fields value. Value should be in
NSERR_EXTENSION_USER_PROTOCOL_IN_USE	1502	0x5DE	Cant update the extension as the user protocol using this extension is in use
NSERR_USER_VSERVER_EXCEED_PARAM_NAME_VALUE_LENGTH	1503	0x5DF	Parameters name or value length exceeds maximum allowed length of 63 characters
NSERR_USER_VSERVER_INVALID_DEFAULT_LB_PROTOCOL_TYPE	2832	0xB10	Default Load Balancing virtual servers protocol type should be a user type.
NSERR_USER_VSERVER_DIFFERENT_TRANSPORT_DEFAULT_LB_PROTOCOL_TYPE	2831	0xB0F	Transport type and default Load Balanocing virtual servers protocol type are not compatible.
NSERR_INTERIM_UPDATE_SUPPORT_IN_RADIUS_ONLY	3436	0xD6C	Treat Radius Interim message as Start message is only

ERROR CODE	ERROR CODE(DECIMAL)	ERROR CODE(HEX)	ERROR MESSAGE
			supported in Radius-Only mode.
NSERR_WARN_VAR_SET_INIT	3437	0xD6D	Set of the init parameter will not affect existing variable values.
NSERR_CI_UNSUPPORTED_VS	3454	0xD7E	A Content Inspection policy can be bound only to LB vservers of type HTTP and SSL, and CS vservers of type HTTP, SSL and PROXY
NSERR_CI_ICAP_FLOWTYPE_MISMATCH	3470	0xD8E	This Content Inspection policys ICAP action is incompatible with the specified bindpoint
NSERR_INVALID_POLICY_BINDING	3438	0xD6E	Policy cannot be bound/unbound to/ from this vserver for the given bindtype.
NSERR_BODY_WITHOUT_ARGUMENT_DEPRECATED	3455	0xD7F	BODY method without argument is deprecated - use BODY method with argument instead
NSERR_HTTP_REQ_USER_DEPRECATED	3456	0xD80	HTTP.REQ.USER has been deprecated. Use AAA.USER instead.

ERROR CODE	ERROR CODE(DECIMAL)	ERROR CODE(HEX)	ERROR MESSAGE
NSERR_ASSIGN_CONTEXT_PCB_NOT_PRESENT	3471	0xD8F	Assignment attempted when not in a connection context.
NSERR_SELFAUTH_DEPRECATED	3481	0xD99	Selfauth as ssotype has been deprecated.
NSERR_RHS_TARGET_REQUIRED	3482	0xD9A	TARGET is required in the rewrite action stringBuilder expression when a refineSearch espression contains a REGEX or XPATH function.
NSERR_AAA_GRP_AS_XML_DEPRECATED	3483	0xD9B	GROUPS_AS_XML has been deprecated. Use AAA.USER.GROUPS instead.
NSERR_REDIRECT_URL_SAML	3484	0xD9C	Redirect URL is a required argument for adding SAML action. Please specify it explicitly, or enter Metadata URL from which it can be imported.
NSERR_CI_IPS_FLOWTYPE_MISMATCH	3486	0xD9E	Policys action is incompatable with specified bindpoint or flow type

POL ERRORS

POL ERRORS

ERROR CODE	ERROR CODE(DECIMAL)	ERROR CODE(HEX)	ERROR MESSAGE
NSERR_INVALIDPOL	2053	0x805	Binding invalid policy
NSERR_NOPOL	2054	0x806	No such policy exists
NSERR_RULEURL	2055	0x807	Rule or URL required
NSERR_DELMC	2056	0x808	Delay or maxConn argument required
NSERR_ACP	2057	0x809	Alternate content path not required
NSERR_ACS	2058	0x80A	Alternate content service not required
NSERR_NOSET_CEXP	2059	0x80B	No support to set compound expression value
NSERR_CEXP_DEPTH	2060	0x80C	Maximum recursive depth reached
NSERR_NOVPNAPP	2061	0x80D	No such intranet application exists
NSERR_NOSET_CSE	2062	0x80E	Changing rule to/ from client security expression is not allowed when policy is in use
NSERR_INVAL_POLNAME	2063	0x80F	Invalid policy Name
NSERR_UNBIND_INVALIDPOL	2097	0x831	Policy not bound

ERROR CODE	ERROR CODE(DECIMAL)	ERROR CODE(HEX)	ERROR MESSAGE
NSERR_EXCEED_MAX_POL_LIMIT	2098	0x832	32 authentication policies are already bound
NSERR_INTERNAL_PI_ERROR	2103	0x837	Internal policy error
NSERR_NEG_POL_VIOL1	2104	0x838	Only negotiate and certificate policies can be bound together
NSERR_NEG_POL_VIOL2	2105	0x839	Negotiate policy in primary can be bound only along with ldap policy (with authentication turned off) in secondary
NSERR_NEG_POL_VIOL3	2106	0x83A	Negotiate policy in secondary can be bound only along with certificate policy in primary
NSERR_SAML_POL_VIOL1	2122	0x84A	SAML policy can only be the last in the cascade or the only one in the cascade
NSERR_SAML_POL_VIOL2	2123	0x84B	SAML policy in primary can be bound only along with ldap policy (with authentication turned off) in secondary
NSERR_SAML_POL_VIOL3	2124	0x84C	SAML policy cannot be bound to secondary cascade

ERROR CODE	ERROR CODE(DECIMAL)	ERROR CODE(HEX)	ERROR MESSAGE
NSERR_SAML_DISABLED	2125	0x84D	SAML policy binding is disabled
NSERR_LDAP_POL_VIOL1	2088	0x828	Only LDAP policies may be bound for group extraction.
NSERR_LDAP_POL_VIOL2	2089	0x829	An LDAP policy bound for group extraction must set -authentication DISABLED.
NSERR_NON_SELECTOR_POLICY	2107	0x83B	Only selector based policies can be bound to sql vserver
NSERR_BAD_SELECTOR	2108	0x83C	Selector does not match content group type
NSERR_BAD_CONTENTGROUP	2109	0x83D	Contentgroup not compatible
NSERR_SQL_POL_GLOBAL_BINDPOINT	2110	0x83E	Datastream cache policies cannot have global/override bindpoint
NSERR_APPQOE_INVALID_ACP	2111	0x83F	Alternate content path is invalid
NSERR_APPQOE_CLTRATE	2086	0x826	Client detect rate is not required
NSERR_APPQOE_DOSTHRESH_NREQ	2087	0x827	DoS attack threshold is not required

ERROR CODE	ERROR CODE(DECIMAL)	ERROR CODE(HEX)	ERROR MESSAGE
NSERR_APPQOE_DOSTHRESH_REQ	2090	0x82a	DoS attack threshold is required
NSERR_APPQOE_ACS_REQ	2091	0x82b	Alternate content service is required
NSERR_APPQOE_ACP_REQ	2092	0x82c	Alternate content path is required with alternate content service
NSERR_CMPTYPE_MISMATCH	2093	0x82d	Compression policy may not take effect due to global parameter type mismatch
NSERR_APPQOE_VSVR_CONFIG_CONFLICT	2094	0x82E	AppQoE policy cannot be bound to vserver configured for PQ/SC
NSERR_APPQOE_SVC_CONFIG_CONFLICT	2095	0x82F	AppQoE policy cannot be bound to vserver with services configured for SC/HDOSP
NSERR_APPQOE_UNSUPPORTED_PROTO	2142	0x85E	Specified policy can be bound only to HTTP/SSL vserver
NSERR_APPQOE_UNSUPPORTED_VSVR	2143	0x85F	Specified policy can be bound only to LB vserver types
NSERR_CLASSIC_POLICY_INVALID_ARG	2145	0x861	Invalid arguments for classic policy binding

ERROR CODE	ERROR CODE(DECIMAL)	ERROR CODE(HEX)	ERROR MESSAGE
NSERR_AUTHN_ADVANCE_INVALID_ARG	2146	0x862	Invalid arguments for advanced authentication policy binding
NSERR_RENAME_NOT_PERMITTED	2147	0x863	Rename operation not permitted
NSERR_PI_AUTHZPLCY_NOT_SUPPORTED	2148	0x864	advanced authorization policy cannot be bound to aaa user/group
NSERR_SAML_2FACTOR_CONFLICT	2149	0x865	SAML 2-Factor policy should always be at the beginning of cascade
NSERR_AUDIT_SYSADVPLCY_BOUND	2150	0x866	Cannot bind Classic Syslog Policy to System global if Advanced Syslog Policy is already bound to SyslogGlobal
NSERR_AUDIT_SYSCLASSICPLCY_BOUND	2151	0x867	Cannot bind Advanced Syslog Policy to SyslogGlobal if Classic Syslog Policy is already bound to System Global
NSERR_CLASSIC_TMSESSPOL_ALREADY_BOUND	2152	0x868	Advanced TM Session Policy cannot be bound if Classic TM Session Policy is already bound to any entity

ERROR CODE	ERROR CODE(DECIMAL)	ERROR CODE(HEX)	ERROR MESSAGE
			(i.e. aaa user, aaa group, authentication vserver, tm global)
NSERR_ADVANCED_TMSESSPOL_ALREADY_BOUND	2153	0x869	Classic TM Session Policy cannot be bound if Advanced TM Session Policy is already bound to any entity (i.e. aaa user, aaa group, authentication vserver, tm global)
NSERR_AUDIT_NSADVPLCY_BOUND	2311	0x907	Cannot bind Classic Nslog Policy to System global if Advanced Nslog Policy is already bound to NslogGlobal
NSERR_AUDIT_NSCLASSICPLCY_BOUND	2312	0x908	Cannot bind Advanced Nslog Policy to NslogGlobal if Classic Nslog Policy is already bound to System Global
NSERR_CLASSIC_AUTHNPOL_ALREADY_BOUND	2219	0x8ab	Advanced authentication Policy cannot be bound if Classic authentication Policy is already bound to system global

ERROR CODE	ERROR CODE(DECIMAL)	ERROR CODE(HEX)	ERROR MESSAGE
NSERR_ADVANCED_AUTHNPOL_ALREADY_BOUND	2220	0x8ac	Classic authentication Policy cannot be bound if Advanced authentication Policy is already bound to system global
NSERR_NEXTFACTOR_ATTRIBMAP_MISMATCH	2221	0x8ad	Bind functionality not supported for current bind-entity and nextFactor
NSERR_CLASSIC_VPNTRAFFICPOL_ALREADY_BOUND	2222	0x8ae	Advanced VPN Traffic Policy cannot be bound if Classic VPN Traffic Policy is already bound to any entity (i.e. aaa user, aaa group, vpn vserver, vpn global)
NSERR_ADVANCED_VPNTRAFFICPOL_ALREADY_BOUND	2223	0x8af	Classic VPN Traffic Policy cannot be bound if Advanced VPN Traffic Policy is already bound to any entity (i.e. aaa user, aaa group, vpn vserver, vpn global)
NSERR_CLASSIC_AUTHPOL_ALREADY_BOUND	2224	0x8b0	Advanced Authorization Policy cannot be bound if Classic Authorization Policy is already bound to any entity (i.e. aaa user, aaa group)

ERROR CODE	ERROR CODE(DECIMAL)	ERROR CODE(HEX)	ERROR MESSAGE
NSERR_ADVANCED_AUTHPOL_ALREADY_BOUND	2225	0x8b1	Classic Authorization Policy cannot be bound if Advanced Authorization Policy is already bound to any entity (i.e. aaa user, aaa group)
NSERR_CLASSIC_VPNSESSPOL_ALREADY_BOUND	2226	0x8b2	Advanced VPN Session Policy cannot be bound if Classic VPN Session Policy is already bound to any entity (i.e. aaa user, aaa group, vpn vserver, vpn global)
NSERR_ADVANCED_VPNSESSPOL_ALREADY_BOUND	2227	0x8b3	Classic VPN Session Policy cannot be bound if Advanced VPN Session Policy is already bound to any entity (i.e. aaa user, aaa group, vpn vserver, vpn global)
NSERR_INVALID_AUTHN_SET_ACTION_SUBTYPE	2228	0x8b4	Setting action for advanced authentication policy not allowed for different action subtype
NSERR_CLASSIC_AUTH_POLICY_DEPRECATED	2229	0x8b5	Classic authentication policies are deprecated. Please use advanced authentication

ERROR CODE	ERROR CODE(DECIMAL)	ERROR CODE(HEX)	ERROR MESSAGE
			policies (i.e. add/set authentication policy)
NSERR_CLASSIC_PREAUTH_POLICY_DEPRECATED	2230	0x8b6	Classic Pre-authentication policies are deprecated. Please use advanced authentication policies (i.e. add/set authentication epaaction, add/set authentication policy)
NSERR_INVALID_POLICY_CONTAIN_CSEC	2231	0x8b7	Binding policy with Client Security expression is not allowed for ICAOnly VPN Virtual Server
NSERR_CLASSIC_AUDITLOGPOL_ALREADY_BOUND	2232	0x8b8	Advanced Audit Policy cannot be bound if Classic Audit Policy is already bound to Vserver
NSERR_ADVANCED_AUDITLOGPOL_ALREADY_BOUND	2233	0x8b9	Classic Audit Policy cannot be bound if Advanced Audit Policy is already bound to Vserver
NSERR_MAX_AUDITLOGPOL_REACHED	2234	0x8ba	Maximum auditlog policy binding to vserver reached
NSERR_PI_ADVANCED_POLICY_ALREADY_ADDED	2235	0x8bb	Failed to bind classic policy because

ERROR CODE	ERROR CODE(DECIMAL)	ERROR CODE(HEX)	ERROR MESSAGE
			advanced policy(s) is already bound.
NSERR_PI_CLASSIC_POLICY_ALREADY_ADDED	2236	0x8bc	Failed to bind advanced policy because classic policy(s) is already bound.
NSERR_CLASSIC_CLIENT_SECURITY_DEPRECATED	2237	0x8bd	Client security expressions are deprecated using this command. Please use advanced authentication policies (i.e. add/set authentication epaaction, add/set authentication policy)
NSERR_CLASSIC_VPNURLPOL_ALREADY_BOUND	2240	0x8c0	VPN URL Policy cannot be bound if Classic VPN URL is already bound to any entity
NSERR_ADVANCED_VPNURLPOL_ALREADY_BOUND	2241	0x8c1	Classic VPN URL cannot be bound if VPN URL Policy is already bound to any entity

POOLED LIC ERRORS

POOLED LIC ERRORS

ERROR CODE	ERROR CODE(DECIMAL)	ERROR CODE(HEX)	ERROR MESSAGE
NSERR_POOLED_LIC_SET_CAP	3569	0xdf1	Unable to set capacity
NSERR_POOLED_LIC_LPED_COMM	3570	0xdf2	Connection to LPED failed
NSERR_POOLED_LIC_SERVER_CONN	3571	0xdf3	Unable to connect to licenseserver

PQ ERRORS

PQ ERRORS

ERROR CODE	ERROR CODE(DECIMAL)	ERROR CODE(HEX)	ERROR MESSAGE
NSERR_PQ_BINDVIP	1360	0x550	Cannot bind same priority policy to VIP
NSERR_PQ_INVALPRIO	1361	0x551	Invalid priority value
NSERR_PQ_2BIGRULE	1362	0x552	Rule argument too big
NSERR_PQ_INVALWT	1363	0x553	Invalid weight value
NSERR_PQ_POLEXIST	1364	0x554	Policy name already in use
NSERR_PQ_NOPOL	1365	0x555	No such PQ policy exists

ERROR CODE	ERROR CODE(DECIMAL)	ERROR CODE(HEX)	ERROR MESSAGE
NSERR_PQ_2MANYREF	1366	0x556	Cannot remove a policy which is bound to VIP
NSERR_PQ_NOLBVIP	1367	0x557	No such LB vserver exists
NSERR_PQ_PHSCONFIG	1368	0x558	Cannot configure PQ on VIP if physical service has PQ configuration
NSERR_PQ_NOBIND	1369	0x559	Cannot unbind PQ Policy which is not bound

QOS ERRORS

QOS ERRORS

ERROR CODE	ERROR CODE(DECIMAL)	ERROR CODE(HEX)	ERROR MESSAGE
NSERR_QOS_64BIT_NOT_SUPPORTED	3776	0xec0	Libqos is not currently supported on 64 bit platforms.
NSERR_QOS_FEATURE_LOCKED	3777	0xec1	QoS featuremask parameter cannot be used with the videooptimization feature enabled.

REWRITE ERRORS

REWRITE ERRORS

ERROR CODE	ERROR CODE(DECIMAL)	ERROR CODE(HEX)	ERROR MESSAGE
NSERR_RW_ACT_INVAL	2817	0xb01	Invalid rewrite action
NSERR_RW_UNDEFACT_INVAL	2818	0xb02	Invalid undef action or log action
NSERR_ACTFLOWMISMATCH	2819	0xb03	Flow types of target and string expression are incompatible
NSERR_RONLY_TAR_EXPR	2820	0xb04	Target must be a simple reference to an unmodified portion of the message
NSERR_DATASET_BINDFAIL	2821	0xb05	Unable to bind the pattern to dataset/ patset
NSERR_DATASET_UNBINDFAIL	2822	0xb06	Pattern does not exist in dataset/ patset
NSERR_DATASET_NOTPRESENT	2823	0xb07	Dataset/Patset does not exist
NSERR_RSP_ACT_INVAL	2824	0xb08	Invalid responder action
NSERR_RSP_POLICY_FLOWTYPE_REQ	2825	0xb09	Responder policy must be a request policy
NSERR_TAR_FLOWTYPE_NRES	2826	0xb0a	Flow type of target should not be response type

ERROR CODE	ERROR CODE(DECIMAL)	ERROR CODE(HEX)	ERROR MESSAGE
NSERR_RSP_CONFIG_LOCK	2827	0xb0b	Responder configuration is temporarily disabled
NSERR_RSP_ACT_MUST_BE_NOOP	2828	0xb0c	Non-terminating policy must have NOOP action
NSERR_PATSET_INVALID_REGEX	2829	0xb0d	Patset contains invalid regex.
NSERR_DATASET_BINDFAIL_DUP_INDEX	2830	0xb0e	Pattern index already in use, try using other index
NSERR_PATSET_BINDFAIL_PATLEN_LT_WU_MINLEN	2833	0xb11	Patset search on a body target not allowed if any of the patset patterns are < 3 or > 112 bytes in length
NSERR_TARGET_NOT_ALLOWED_IN_RULE	2834	0xb12	Expression involving Target not allowed in rule.
NSERR_RW_TARGET_NOT_ALLOWED_IN_STRBUILDER	2835	0xb13	Expression involving Target not allowed in StringBuilder expression.
NSERR_PIT_INVAL	2836	0xb14	Invalid Packet data
NSERR_INVAL_INVOKEPOINT	2837	0xb15	Invalid invoke point
NSERR_PIT_MAX_PACKET	2838	0xb16	Packet size exceeds maximum size

ERROR CODE	ERROR CODE(DECIMAL)	ERROR CODE(HEX)	ERROR MESSAGE
NSERR_REFINE_SEARCH_INVALID	2839	0xb17	Regular expression for patterns not allowed when refine search is specified
NSERR_EXTEND_INVALID	2840	0xb18	Extend not allowed for non-body expressions
NSERR_NON_EXTEND_EXPR	2841	0xb19	Non extend expressions are not allowed
NSERR_EXTEND_INVAL_PIRL	2842	0xb1a	Extend not allowed in the string builder expression
NSERR_INVAL_TAR_EXPR	2843	0xb1b	Invalid target expression
NSERR_INVAL_PATTERN_SEARCH_SET	2844	0xb1c	Cannot set both pattern and search
NSERR_INVAL_SEARCH_ARGS	2845	0xb1d	Invalid argument: search supports text, xpath, xpath_json, xpath_html, regex and patset
NSERR_RSP_ACT_MUST_BE_RESET_DROP	2846	0xb1e	Policy action must be DROP|RESET
NSERR_INVAL_SEARCH_SYNTAX	2847	0xb1f	Invalid search syntax
NSERR_INVAL_SEARCH_XPATH_SYNTAX	2848	0xb20	Invalid xpath syntax

ERROR CODE	ERROR CODE(DECIMAL)	ERROR CODE(HEX)	ERROR MESSAGE
NSERR_INVAL_SEARCH_PATSET_SYNTAX	2849	0xb21	Invalid patset syntax
NSERR_INVAL_SEARCH_REGEX_SYNTAX	2850	0xb22	Invalid regex syntax
NSERR_RW_REQ_BODY_NOT_ALLOWED	2851	0xb23	Request body based expression is not allowed
NSERR_RENAME_NOTSUPPORTED	2852	0xb24	Renaming this entity is not supported as entity based expression is configured
NSERR_ENTITY_REMOVAL_NOTALLOWED	2853	0xb25	Removing this entity is not allowed as entity based expression is configured
NSERR_PI_ENTITY_EXISTS	2854	0xb26	Advanced expression entity with same name already exists.
NSERR_INCOMPATIBLE_CALLOUT_CHANGE	2855	0xb27	Incompatible callout change for in-use callout.
NSERR_STRINGMAP_NOTPRESENT	2856	0xb28	String map does not exist
NSERR_HTTP_PROFILE_ACTION_NOT_ALLOWED	2857	0xb29	Responder action using HTTP protocol expressions is not

ERROR CODE	ERROR CODE(DECIMAL)	ERROR CODE(HEX)	ERROR MESSAGE
			allowed in HTTP profile
NSERR_RSP_ACT_MUST_BE_DROP_NOOP_RESPONDWITH	2858	0xb2A	Policy action must be DROP|NOOP|RESPONDWITH
NSERR_RSP_ACT_MUST_BE_DROP_NOOP	2899	0xb53	Policy action must be DROP|NOOP
NSERR_RSP_UNDEF_ACT_MUST_BE_DROP_NOOP	2859	0xb2B	Policy undef action must be DROP|NOOP
NSERR_INCOMPATIBLE_UNDEF	2860	0xb2C	Incompatible global undef action for SIP policy
NSERR_PIT_ACTION_EVAL_INVAL	2861	0xb2d	Action cannot be evaluated due to wrong input type.
NSERR_RW_PATTERN_DEPRECATED	2862	0xb2e	-pattern is deprecated, use -search instead
NSERR_STRLIT_MAXLEN	2863	0xb2f	String literal length exceeds maximum. Allowed maximum length is 32767 bytes or the maximum expression length, whichever is less. In most situations the maximum expression length is either 8191 or 1499 bytes.

ERROR CODE	ERROR CODE(DECIMAL)	ERROR CODE(HEX)	ERROR MESSAGE
NSERR_DATASET_INVALID	2866	0xb32	Invalid dataset element
NSERR_DATASET_BINDFAIL_DUP_PATTERN	2867	0xb33	Pattern already bound to dataset/ patset, try using other pattern
NSERR_TIMER_ACT_INVAL	2868	0xb34	Invalid timer action
NSERR_TIMER_ENTITY_INUSE	2869	0xb35	Timer entity name already in use
NSERR_TIMER_ENTITY_GLOBAL_BINDPOINT_INVAL	2870	0xb36	Invalid global bindpoint
NSERR_PATSET_BUILTIN	2871	0xb37	Built-in Patsets cannot be modified or deleted
NSERR_CROSS_MAX_LIMIT	2872	0xb38	Maximum value can be 2147483647
NSERR_INVALID_IPV4_FORMAT	2873	0xb39	Value should be in IPv4 format
NSERR_INVALID_IPV4_MAX_FIELD_VALUE	2880	0xb40	Maximum value of an IPv4 field can be 255
NSERR_INVALID_IPV6_FORMAT	2881	0xb41	Invalid IPv6 address format
NSERR_INVALID_NUMERIC_FORMAT	2882	0xb42	Value should be in decimal or hexadecimal format

ERROR CODE	ERROR CODE(DECIMAL)	ERROR CODE(HEX)	ERROR MESSAGE
NSERR_VALID_PREFIX_NUMERIC_FORMAT	2048	0x800	Value is valid decimal or hexadecimal format, but is only a prefix of the string
NSERR_DATASET_PATTERN_ALREADY_BOUND	2883	0xb43	Specified pattern is already bound to dataset/patset
NSERR_PI_CIRCULAR_REFERENCE_DETECTED	2884	0xb44	Circular reference detected
NSERR_INVALID_ULONG_FORMAT	2885	0xb45	Value should be in unsigned long format like 23445, 88888999
NSERR_INVALID_MAC_ADDRESS_FORMAT	2886	0xb46	Value should be in mac address format like ff:ff:ff:ff:ff:ff
NSERR_INVALID_DOUBLE_FORMAT	2887	0xb47	Value should be in double format like 223, 22.3, 234e2
NSERR_CROSS_ULONG_MAX_LIMIT	2888	0xb48	Maximum value can be 18446744073709551615
NSERR_CA_ACT_INVAL	2889	0xb49	Invalid ContentAccelerator action
NSERR_INVAL_AVP_INSERT_EXPR	2890	0xb4a	Rewrite policy has invalid action. For Diameter bind points use DIAMETER.NEW_A

ERROR CODE	ERROR CODE(DECIMAL)	ERROR CODE(HEX)	ERROR MESSAGE
			VP() in action and for Radius bind points use RADIUS.NEW_AVP( ) in action as stringBuilderExpr expression to insert/ replace AVP
NSERR_INVAL_AVP _LOOKUP_EXPR	2891	0xb4b	Rewrite policy has invalid action. For Diameter/Radius bind points, use AVP lookup expression as target expression. e.g. RADIUS.REQ.AVP(1 27)/ DIAMETER.REQ.AV P(9000)/ AVP_START/ AVP_END
NSERR_INVAL_DEL ETE_AVP_EXPR	2892	0xb4c	Rewrite policy has invalid action. Deletion of AVP should be done using AVP based expression like RADIUS.REQ.AVP(3 3)/ DIAMETER.REQ.AV P(9999)
NSERR_INVAL_NE W_MESSAGE_EXP R	2893	0xb4d	Responder policy has invalid action. Use expressions like DIAMETER.NEW_A NSWER/ DIAMETER_NEW_R

ERROR CODE	ERROR CODE(DECIMAL)	ERROR CODE(HEX)	ERROR MESSAGE
			EDIRECT as target expression to create new diameter message and RADIUS.NEW_ANSWER/ RADIUS.NEW_ACCESSREJECT for RADIUS message
NSERR_INVAL_DIAMETER_REDIRECT_ACTION	2894	0xb4e	Binding Invalid Diameter action. Use DIAMETER.NEW_REDIRECT as target expression to create new redirect message
NSERR_INVAL_DIAMETER_ACTION	2895	0xb4f	Invalid action for Diameter bind point.
NSERR_CA_URL_INVAL	2896	0xb50	URL length more than 2000
NSERR_PIT_CONNECTION_ERROR	2897	0xb51	Connection error
NSERR_INVAL_SEARCH_AVP	2898	0xb52	AVP code should be a number
NSERR_DATASET_PATTERN_LIMIT_OVERFLOW	2900	0xb54	Cannot bind more than 5000 patterns to a patset/dataset
NSERR_INVAL_RADIUS_AVPCODE	2901	0xb55	RADIUS AVP code should be in 0-255 number range
NSERR_INVAL_RADIUS_RESPONSE	2902	0xb56	Invalid RADIUS response code

ERROR CODE	ERROR CODE(DECIMAL)	ERROR CODE(HEX)	ERROR MESSAGE
NSERR_RESP_ACT_REASON_PHRASE_REQUIRED	2903	0xb57	Please also specify a reason phrase with a user-defined HTTP status code.
NSERR_RESP_ACT_RESPONSE_CODE_REASON_PHRASE_NOT_ALLOWED	2904	0xb58	responseStatusCode and reasonPhrase options are allowed only for respondwithhtmlpage and redirect action types.
NSERR_RESP_ACT_INVALID_REDIRECT_CODE	2905	0xb59	For redirect action type the response status code must be in the range 300-399.
NSERR_RESP_ACT_INVALID_HTTP_STATUS_CODE	2906	0xb5A	Invalid HTTP status code.
NSERR_RESP_ACT_EMPTY_REASON_PHRASE_NOT_ALLOWED	2907	0xb5B	Empty reasonPhrase string not allowed.
NSERR_RSP_ACT_MUST_BE_DROP_NOOP_RESPONDWITH_RESET	2908	0xb5C	Policy action must be DROP|NOOP|RESPONDWITH|RESET
NSERR_MAX_STREAMING_PATTERN_LIMIT	2909	0xb5D	Maximum number of streaming patterns exceeded.
NSERR_RESP_UPDATE_HTMLPAGE_NOT_REFERENCED	1223	0x4c7	Responder HTML Page must be referenced in a

ERROR CODE	ERROR CODE(DECIMAL)	ERROR CODE(HEX)	ERROR MESSAGE
			responder action prior to update.
NSERR_PATSET_STREAMING_BINDFAIL_PATLEN_LT_WU_MINLEN	1224	0x4c8	Patterns which are < 3 or > 112 bytes in length are not allowed to be bound to the patset, if patset is being used for streaming search or in rewrite feature.

RISE ERRORS

RISE ERRORS

ERROR CODE	ERROR CODE(DECIMAL)	ERROR CODE(HEX)	ERROR MESSAGE
NSERR_RISE_NOT_ENABLED	3473	0xd91	Feature RISE is not enabled
NSERR_RISE_NS_MODE_APBR_DISABLED	3474	0xd92	Operation not permitted as ns mode RISE_APBR is disabled. Retry after ns mode RISE_ABPR is enabled
NSERR_RISE_NS_MODE_RHI_DISABLED	3475	0xd93	Operation not permitted as ns mode RISE_RHI is disabled. Retry after ns mode RISE_RHI is enabled

ERROR CODE	ERROR CODE(DECIMAL)	ERROR CODE(HEX)	ERROR MESSAGE
NSERR_RISE_RHI_NOT_IMPLEMENTED	3476	0xd94	RISE RHI mode is not implemented
NSERR_RISE_NOT_LICENSED	3477	0xd95	RISE cannot be enabled in Standard NS license
NSERR_RISE_DISABLED_ALREADY	3478	0xd96	RISE mode is already disabled
NSERR_RISE_WARN_DIRECT_DISABLE	3479	0xd97	Disabling direct mode with one or more active RISE profile(s) on DIRECT ATTACH mode may cause corresponding RISE profile(s) to not come UP on reboot
NSERR_RISE_WARN_INDIRECT_DISABLE	3480	0xd98	Disabling indirect mode will cause any active RISE profile(s) to go down

RNAT ERRORS

RNAT ERRORS

ERROR CODE	ERROR CODE(DECIMAL)	ERROR CODE(HEX)	ERROR MESSAGE
NSERR_RNAT_NAME_EXISTS	976	0x3d0	A RNAT with same name exists
NSERR_VLAN_IPV6_LL_N_ALLOW	962	0x3c2	IP address cannot be link-local address

ERROR CODE	ERROR CODE(DECIMAL)	ERROR CODE(HEX)	ERROR MESSAGE
NSERR_VLAN_INTF_BOUND_NSVLAN	963	0x3c3	Cannot unbind interface from nsvlan
NSERR_VLAN_INVALID	964	0x3c4	Incorrect vlan specified
NSERR_SYNC_VLAN_IPBOUND	965	0x3c5	Ipaddress binding/ unbinding is not allowed on sync or default vlan
NSERR_VLAN_SUBNET_ALREADY_BOUND	966	0x3c6	Subnet already bound to this vlan
NSERR_UNABLE_GET_VLAN_IP	967	0x3c7	Unable to get vlan of the IPaddress
NSERR_IP_VLAN_BOUND	968	0x3c8	Ipaddress already bound to a vlan
NSERR_VLAN_UNABLE_ADD	969	0x3c9	Unable to add vlan
NSERR_DEFVLAN	970	0x3ca	Default VLAN or NSVLAN cannot be modified
NSERR_IPV6_ROUTING_BRIDGEGRP	971	0x3cb	vlan is part of a bridge group. Cannot enable routing
NSERR_VLAN_SDXMGMTVLAN_CLUSTER	972	0x3cc	Operation not permitted. vlan is SDX management vlan in a cluster
NSERR_ACL_INUSE_RNAT	973	0x3cd	This ACL is used for RNAT/NAT64/LSN/ FORWARDINGSES

ERROR CODE	ERROR CODE(DECIMAL)	ERROR CODE(HEX)	ERROR MESSAGE
			SION. Action should be ALLOW
NSERR_RNAT_CM_TCPPROXY	974	0x3ce	TCPPROXY should be disabled for connection failover to work properly.
NSERR_RNAT_DYN_PERM_RT	2413	0x96d	RNAT cannot be configured on DYNAMIC/ PERMANENT route
NSERR_RNAT_SRCIPPERSISTENCY_NOTSUPCLU	1218	0x4c2	RNAT SourceIPPersistency is not supported in Cluster.

SNMP ERRORS

SNMP ERRORS

ERROR CODE	ERROR CODE(DECIMAL)	ERROR CODE(HEX)	ERROR MESSAGE
NSERR_INVALTHRESHOLD	3457	0xd81	Threshold value for this alarm is a percentage (1 - 100)
NSERR_WRONGTHRESHOLDS	3458	0xd82	Normal threshold must be lower than alarm threshold
NSERR_INVALID_SRCIP	3459	0xd83	Valid source IP: NSIP, SNIP, MIP. In cluster setup, valid source IP: CLIP, Striped SNIP. In non

ERROR CODE	ERROR CODE(DECIMAL)	ERROR CODE(HEX)	ERROR MESSAGE
			default partition, valid source IP: SNIP
NSERR_THRESHOLD_UNSETTABLE	3460	0xd84	Threshold value cannot be set/unset for this alarm
NSERR_TIME_UNSETTABLE	3461	0xd85	Time interval cannot be set/unset for this alarm
NSERR_WRONG_LOW_THRESHOLDS	3462	0xd86	Normal threshold must be higher than alarm threshold
NSERR_NOSNMPUSER	3463	0xd87	SNMP user doesnt exist
NSERR_WRONG_SNMPVERSION	3464	0xd88	Bind command allowed only for V3 traps
NSERR_V3TRAP_COMMUNITY	3465	0xd89	V3 traps cannot have community
NSERR_IPINUSE	3466	0xd8A	IP is used in SNMP trap configuration
NSERR_DEFAULTONLY	3467	0xd8B	Alarm not supported in non-default partition
NSERR_OWNERNODE	3468	0xd8C	Owner node not specified or invalid
NSERR_FIPS_SNMP	3469	0xd8D	Less secure SNMP configuration is not supported in FIPS mode

SSL ERRORS

SSL ERRORS

ERROR CODE	ERROR CODE(DECIMAL)	ERROR CODE(HEX)	ERROR MESSAGE
NSERR_SSL_CERT	1536	0x600	Invalid certificate
NSERR_SSL_PKEY	1537	0x601	Invalid private key, or PEM pass phrase required for this private key
NSERR_SSL_NOMATCH	1538	0x602	Certificate and private key do not match
NSERR_SSL_CERTTYPE	1539	0x603	Invalid cetificate type
NSERR_SSL_NOCERT	1540	0x604	Certificate does not exist
NSERR_SSL_REFEXT	1541	0x605	Certificate is referenced by a CRL, OCSP responder, vserver, service, monitor, SSL profile, CA Cert Group, another certificate, or a policy expression using XML_ENCRYPT() or XML_DECRYPT()
NSERR_SSL_BIND	1542	0x606	Certificate binding does not exist
NSERR_SSL_LINK	1543	0x607	Certificate cant be linked to the same certificate

ERROR CODE	ERROR CODE(DECIMAL)	ERROR CODE(HEX)	ERROR MESSAGE
NSERR_SSL_NEED_SSLPROTO	1544	0x608	Objects protocol type is not SSL
NSERR_SSL_NOLINK	1545	0x609	Certificate does not have any CA link
NSERR_SSL_BINDOR	1546	0x60A	Current certificate replaces the previous binding
NSERR_SSL_NOSVRCERT	1547	0x60B	Certificate is not a server/client certificate
NSERR_SSL_ISSUBMIS	1548	0x60C	Issuer certificate mismatch
NSERR_SSL_CRL	1549	0x60D	Invalid CRL
NSERR_SSL_NOCRL	1550	0x60E	CRL does not exist
NSERR_SSL_DHCOUNT	1551	0x60F	DH Refresh count should be 0 or >=500
NSERR_SSL_SESSTO	1552	0x610	Session timeout should be > 0
NSERR_SSL_ERSACOUNT	1553	0x611	eRSA Refresh count should be 0 or >=500
NSERR_SSL_DH_SIZE	1554	0x612	DH params of size greater than 4096 bits not supported
NSERR_DHPATH	1555	0x613	DH file path mandatory if DH enabled

ERROR CODE	ERROR CODE(DECIMAL)	ERROR CODE(HEX)	ERROR MESSAGE
NSERR_CERTHEADER	1556	0x614	Cert header tag mandatory if cert enabled
NSERR_SESSHEADER	1557	0x615	Sess header tag mandatory if sess enabled
NSERR_CIPHER_PERM	1558	0x616	Default ciphers/ aliases cannot be added or removed
NSERR_SSL_ERSADISABLED	1559	0x617	Setting eRSA count when eRSA is disabled
NSERR_SSL_DHDISABLED	1560	0x618	Setting DH count when DH is disabled
NSERR_SSL_SESSDISABLED	1561	0x619	Setting session timeout when session reuse is disabled
NSERR_SSL_PKEY_SIZE	1562	0x61A	Certificate with key size greater than RSA4096 or DSA2048 bits not supported
NSERR_SSL_NOT_APPLICABLE	1563	0x61B	Option is not applicable for this type of service
NSERR_CERTDNHEADER	1564	0X61C	Cert DN tag mandatory if cert-DN enabled

ERROR CODE	ERROR CODE(DECIMAL)	ERROR CODE(HEX)	ERROR MESSAGE
NSERR_CERTISSUERHEADER	1565	0X61D	Cert issuer tag mandatory if cert-issuer enabled
NSERR_CIPHERHEADER	1566	0X61E	Cipher tag mandatory if cipher header enabled
NSERR_SSL_INTERNALERR	1567	0x61F	Internal Error
NSERR_SSL_NOCACERT	1568	0x620	cacert does not exists
NSERR_SSL_REFRESHDIS	1569	0x621	crl refresh disabled
NSERR_SSL_SVRPORTNEEDED	1570	0x622	Server/port information is needed for enabling auto refresh
NSERR_SSL_BASEOBJNEEDED	1571	0x623	baseDN is required for enabling auto refresh
NSERR_SSL_CIPHER_REDIRECT	1572	0x624	Send Internal Cipher mismatch error page
NSERR_SSL_NODSA	1573	0x625	Loading of certificate and key of type DSA(DSS) is not supported with FIPS
NSERR_SSL_FIPSREFEXT	1574	0x626	The FIPS key is referenced by a certificate
NSERR_SSL_NOFIPSKEY	1575	0x627	No such FIPS key

ERROR CODE	ERROR CODE(DECIMAL)	ERROR CODE(HEX)	ERROR MESSAGE
NSERR_NOFIPSCARD	1576	0x628	Operation not permitted - no FIPS card present in the system
NSERR_FIPSCARDNOTCONF	1577	0x629	Operation not permitted - FIPS card is not configured
NSERR_SSL_SSLV2_REDIRECT	1578	0x62A	Send Internal SSL protocol mismatch error page
NSERR_SSL_MODSIZE64	1579	0x62B	Modulus size in bytes should be multiple of 64
NSERR_SSL_NONFIPSKEY	1580	0x62C	Configuration of non-FIPS key on FIPS system not allowed
NSERR_NFIPS_FIPS_UPD	1581	0x62D	Cannot update a non-FIPS certificate with a FIPS certificate
NSERR_FIPS_NFIPS_UPD	1582	0x62E	Cannot update a FIPS certificate with a non-FIPS certificate
NSERR_SSL_ISSUER_NOTIN_GLBCERTLIST	1583	0x62F	Unable to find the CA certificate for the CRL
NSERR_SSL_CRLSIGCHECK_FAIL	1584	0x630	Signature check on the CRL failed

ERROR CODE	ERROR CODE(DECIMAL)	ERROR CODE(HEX)	ERROR MESSAGE
NSERR_SSL_PORT REWRITE	1585	0x631	SSL port rewrite can be enabled only when SSL redirect is enabled
NSERR_SSL_SSLV 2_RENEG_CLIENT_ CERT	1586	0x632	Send Internal error page for SSLv2 protocol and client authentication with session renegotiation
NSERR_SSL_BRKLI NK	1587	0x633	All incompatible CA links/Cert bindings were broken during the update operation
NSERR_SSL_CERT _NOT_YET_VALID	1588	0x634	The specified certificate is not yet valid
NSERR_SSL_CERT _EXPIRED	1589	0x635	The certificate has expired
NSERR_SSL_EXPI RED_BRKLINK	1590	0x636	All incompatible CA links were broken during the update operation. (Note: the certificate has expired)
NSERR_SSL_NYVA LID_BRKLINK	1591	0x637	All incompatible CA links were broken during the update operation. (Note: the certificate is not yet valid)
NSERR_SSL_OCSP _RESPCERT	1592	0x638	No such responder certificate.

ERROR CODE	ERROR CODE(DECIMAL)	ERROR CODE(HEX)	ERROR MESSAGE
NSERR_SSL_OCSP_SIGNCERT	1593	0x639	No such signing certificate.
NSERR_SSL_NO_SESS_TKT	1594	0x63A	Session Ticket feature is not supported on FIPS platform
NSERR_SSL_OCSP_INVALID_AIA	1595	0x63B	Invalid OCSP AIA URL found
NSERR_SSL_OCSP_AIA_LEN_NOT_SUPPORTED	1596	0x63C	OCSP AIA URL length is should be <126
NSERR_SSL_SSL3_SUPPORT	1597	0x63D	Enabling of SSLv3 is not supported on this platform.
NSERR_SSL_BOUNDTO_SSLACTION	1598	0x63E	Vserver cannot be removed as its referenced by one or more SSL Actions. Remove SSL action(s) prior to removing vserver.
NSERR_SSL_INVALID_SESS_TKT_DATA	1599	0x63F	invalid session ticket key data len, length should be 64 bytes
NSERR_SSL_SAME_VS_IN_FORWARD	1600	0x640	Vserver to which packets will be forwarded cannot be same as the vserver to which this policy is getting bound.
NSERR_NAME_ALPHANUM_DASH1	3358	0xd1e	Name must only contain alphanumeric

ERROR CODE	ERROR CODE(DECIMAL)	ERROR CODE(HEX)	ERROR MESSAGE
			characters and dashes and cannot start with a number.
NSERR_SSL_PROFILE_SSLI_NOT_ALLOWED	3370	0xd2a	SSL interception can only be enabled on profile bound to proxy vserver
NSERR_SSL_CTX_SSLI_PROFILE_NOT_ALLOWED	3371	0xd2b	Profile with SSL interception enabled cannot be set on non proxy vservers
NSERR_SSL_PROFILE_SSLI_CLUSTER_NOT_SUPPORTED	3372	0xd2c	SSL interception is not supported on cluster
NSERR_SSL_DEFAULT_PROFILE_SSLI_NOT_SUPPORTED	3373	0xd2d	SSL interception cannot be enabled on default frontend SSL profile

SSL2 ERRORS

SSL2 ERRORS

ERROR CODE	ERROR CODE(DECIMAL)	ERROR CODE(HEX)	ERROR MESSAGE
NSERR_SSL_FIPSCARDLOCKED	3584	0xe00	FIPS card locked due to three unsuccessful login attempts
NSERR_SSL_DOMINCOMPAT	3585	0xe01	Certificate is registered to a different domain; use the no domain

ERROR CODE	ERROR CODE(DECIMAL)	ERROR CODE(HEX)	ERROR MESSAGE
			checkoption to force the operation
NSERR_SSL_NOMTHDCHANGE	3586	0xe02	Cannot change refresh method of CRL
NSERR_SSL_URLSRVRNEEDED	3587	0xe03	Either URL or server-IP required on CRL
NSERR_SSL_INVALID_URL	3588	0xe04	Invalid URL
NSERR_SSL_MIXPARAMS	3589	0xe05	LDAP and HTTP parameters cannot both be specified
NSERR_SSL_PKEY_MINSIZE	3590	0xe06	Certificate of size smaller than 512 bits not supported
NSERR_SSL_SYNCINPROGRESS	3591	0xe07	Another synchronization is already in process, please try again later
NSERR_SSL_SYNCFAILED	3592	0xe08	Synchronization failed, please try again
NSERR_SSL_CIPHGRP_REFCNT	3593	0xe09	Cipher group has bound entities or is referenced by an SSL vserver or SSL service
NSERR_SSL_CVM_NODSA	3594	0xe0A	Loading DSA(DSS) certificate and key not supported on this platform

ERROR CODE	ERROR CODE(DECIMAL)	ERROR CODE(HEX)	ERROR MESSAGE
NSERR_CERTHASHHEADER	3595	0xe0B	CertHash header tag mandatory if certHash is enabled
NSERR_SSL_CRLINREFRESH	3596	0xe0C	CRL refresh in progress, details cannot be displayed
NSERR_SSL_CRLMEM_EXCEEDS	3597	0xe0D	CRL memory exhausted. Cannot load any more CRLs
NSERR_SSL_CRLINDELETION	3598	0xe0E	CRL deletion in progress, details cannot be displayed
NSERR_SSL_INDELETE_NOREFRESH	3599	0xe0F	CRL deletion in progress, cannot be refreshed
NSERR_SSL_INREFRESH_NODELETE	3600	0xe10	CRL refresh in progress, cannot be removed
NSERR_NOMIX	3601	0xe11	Cannot specify both HTTP data insertion and SSL actions
NSERR_NOPOLICY_NONTRSVC	3602	0xe12	Cannot bind SSL policy to SSL backend service
NSERR_SSL_SSLPOL_BIND_CONST	3603	0xe13	Cannot bind non-SSL policy to SSL vserver/service
NSERR_SSL_NO_USABLE_CIPHERS	3604	0xe14	No usable ciphers configured on the SSL vserver/service

ERROR CODE	ERROR CODE(DECIMAL)	ERROR CODE(HEX)	ERROR MESSAGE
NSERR_SSL_CERT_NOT_CA	3605	0xe15	Not a CA certificate
NSERR_SSL_CACERT_NO_CRLSIGN	3606	0xe16	Specified certificate is either not a CA cert, or does not have privileges to issue CRLs
NSERR_SSL_CRL_EXPIRED	3607	0xe17	CRL has expired
NSERR_SSL_CRL_NOTYET_VALID	3608	0xe18	CRL is not yet valid
NSERR_SSL_PARSING_DELTA_CRL_EXTN	3609	0xe19	Parsing of Delta-CRL extension failed
NSERR_SSL_DELTA_CRL_MISSING_BASE_CRL	3610	0xe1A	Base-CRL for the specified Delta-CRL is missing
NSERR_NOFIPSCIPHER	3611	0xe1B	Specified cipher/ cipher-alias is not FIPS-approved
NSERR_NOFIPSCIPHERGRP	3612	0xe1C	Cipher group does not contain all FIPS-approved ciphers
NSERR_NONFIPSCIPHERTOGRP	3613	0xe1D	Cannot add non FIPS approved cipher to cipher group
NSERR_NONFIPSALIASTOGRP	3614	0xe1E	Cannot add non FIPS approved cipher alias to cipher group

ERROR CODE	ERROR CODE(DECIMAL)	ERROR CODE(HEX)	ERROR MESSAGE
NSERR_NONFIPSGROUPTOGRP	3615	0xe1F	Cannot add non FIPS cipher group to another cipher group
NSERR_SSL_IMPORT_FIPSKEY_NAME_MISMATCH	3616	0xe20	Specified FIPS key name does not match with the exported FIPS key name
NSERR_SSL_PKEY_SIZE_CA	3617	0xe21	CA certificate of size greater than 4096 bits not supported
NSERR_SSL_CRL_PORT_MISMATCH	3618	0xe22	Port specified in URL does not match -port parameter
NSERR_SSL_PKEY_SIZE_VPX	3619	0xe23	Certificate with key size greater than RSA512 or DSA512 bits not supported
NSERR_SSL_DH_SIZE_VPX	3620	0xe24	DH params of size greater than 512 bits not supported
NSERR_NOENT_CIPHER	3621	0xe25	No such cipher/ cipherAlias/ cipherGroup
NSERR_FIPSFWWRONGMAJOR	3622	0xe26	The current firmwares major version is not supported for update operation
NSERR_FIPSFWWRONGMINOR	3623	0xe27	The current firmwares minor version is not

ERROR CODE	ERROR CODE(DECIMAL)	ERROR CODE(HEX)	ERROR MESSAGE
			supported for update operation
NSERR_FIPSFWUPDATED	3624	0xe28	The current firmware is already updated to 4.6.1
NSERR_SSL_PENDING_CMDS	3625	0xe29	Other commands (card health monitoring/traffic) are pending to FIPS card. Please try the firmware update command after some time
NSERR_FIPSFWUPDATEDOREBOOT	3626	0xe2A	Operation not permitted - FIPS card firmware update done, please reboot the system
NSERR_SSL_SNI_NOTENABLE	3627	0xe2B	SNI feature not enabled on the vserver/service
NSERR_SSL_NO_CN	3628	0xe2C	CommonName not present in certificate, it is must for a certificate to use for SNI
NSERR_SSL_DUP_SNICERT	3629	0xe2D	Trying to bind SNI certificate with duplicate CommonName, operation failed
NSERR_SSL_SNI_NOTVALID_SERV	3630	0xe2E	Operation not permitted - SNI feature not

ERROR CODE	ERROR CODE(DECIMAL)	ERROR CODE(HEX)	ERROR MESSAGE
			supported on internal service
NSERR_SSL_NOT_SUPPORTED_ON_PLATFORM	3631	0xe2F	Feature not supported on this platform
NSERR_OCSP_REFERENCES	3632	0xe30	OCSP responder is bound to a vserver/ certkey pair. Use unbind ssl certkey.
NSERR_OCSP_SIGNER_NOKEY	3633	0xe31	Signing certificate must also have a private key.
NSERR_SSL_NOT_SUPPORTED	3634	0xe32	OCSP responder must be an HTTP server; SSL is not supported.
NSERR_OCSP_TOO_MANY_RESPONDERS	3635	0xe33	Too many OCSP responders configured to add another. Please delete some and try again.
NSERR_OCSP_NO_DNS_SERVER_CONFIGURED	3636	0xe34	Unable to resolve DNS name.
NSERR_SSL_DUP_SNICERT_BRKLINK	3637	0xe35	Some of the existing SNI cert bindings are broken due to presence of certificate with duplicate Common Name.

ERROR CODE	ERROR CODE(DECIMAL)	ERROR CODE(HEX)	ERROR MESSAGE
NSERR_SSL_NO_CN_BRLLINK	3638	0xe36	All exitsing SNI Cert bindings are broken during update operation due to missing Common Name.
NSERR_NGFIPSRESETREBOOT	3639	0xe37	Operation not permitted - FIPS card was reset, please reboot the system
NSERR_NGFIPSINITREBOOT	3640	0xe38	Operation not permitted - FIPS card was initialized, please reboot the system
NSERR_FIPSCMDTIMEOUT	3641	0xe39	Operation timedout on the FIPS card, please try again
NSERR_SSL_SIMTIMEOUT	3642	0xe3a	Operation timed out or repeated, please wait for 10 mins and redo the SIM/HA configuration steps.
NSERR_SSL_NGFIPS_QFULL	3643	0xe3b	FIPS card command queue full. Please try again later
NSERR_SSL_NOMEM_VSVRSRVLISTNODE	3644	0xe3c	Failed to allocate memory for CertkeyVserverServ List Node.
NSERR_SSL_CERTKEY_SIZE64	3645	0xe3d	Certificate with key size (modulus) that is

ERROR CODE	ERROR CODE(DECIMAL)	ERROR CODE(HEX)	ERROR MESSAGE
			not multiple of 512 bits is not supported
NSERR_SNI_ATK	3646	0xe3e	Host header field in the HTTP request does not match with the SNI domain name
NSERR_SNI_NOHOSTHDR	3647	0xe3f	Host header missing in the HTTP header for SNI enabled session
NSERR_CRL_SHMEM_ALLOC_FAIL	3648	0xe40	Crl node allocation in the shared mem is failed
NSERR_SSL_PE_NO_SET_UNSET	3649	0xe41	Set/Unset operation not supported for classic SSL polices
NSERR_SSL_NOT_CTL_POL	3650	0xe42	Not a control policy
NSERR_SSL_NOT_DATA_POL	3651	0xe43	Not a data policy
NSERR_SSL_TYPE_REQD	3652	0xe44	Type is required
NSERR_SSL_NOT_PI_POL	3653	0xe45	Advanced policy attributes specified with classic policy
NSERR_SSL_PE_POL_POL_LBL	3654	0xe46	Cannot bind classic policy to policy label
NSERR_SSL_CERT_MISSING_PARAM	3655	0xe47	Required parameters missing in the certificate. Please

ERROR CODE	ERROR CODE(DECIMAL)	ERROR CODE(HEX)	ERROR MESSAGE
			check the certificate for completeness
NSERR_SSL_NOMEM_CERTKEY_OCSPRESP_LISTNODE	3656	0xe48	Failed to allocate memory for CertkeyOCSPRespList Node.
NSERR_SSL_OCSP_DUPLICATE	3657	0xe49	An ocspResponder is already bound with the specified priority.
NSERR_SSL_BIND_CP	3658	0xe4a	Binding of SSL classic policy is not supported, use equivalent advanced SSL policy
NSERR_SSL_ADD_CP	3659	0xe4b	Adding SSL classic policy is not supported, use equivalent advanced rule
NSERR_SSL_BUNDLE_IC_FILE_EXISTS	3660	0xe4c	Certificate file for intermediate certificate already exists.
NSERR_SSL_BUNDLE_SCERT_MISSING	3661	0xe4d	Server certificate must be placed first in certificate bundle file.
NSERR_SSL_BUNDLE_CERT_MISSING	3662	0xe4e	No certificates present in the certificate bundle file.
NSERR_SSL_BUNDLE_FAILED	3663	0xe4f	Processing of certificate bundle file failed.

ERROR CODE	ERROR CODE(DECIMAL)	ERROR CODE(HEX)	ERROR MESSAGE
NSERR_SSL_BUNDLE_PARSE_ERR	3664	0xe50	Unable to parse the certificate bundle file.
NSERR_SSL_BUNDLE_MAX_CERT	3665	0xe51	Exceeded maximum Intermediate certificates limit of 9.
NSERR_SSL_BUNDLE_MAX_KEY	3666	0xe52	Only one private-key is allowed in the certificate bundle file.
NSERR_SSL_BUNDLE_IC_FILE_CREATE_FAILED	3667	0xe53	Intermediate certificate file creation failed.
NSERR_SSL_SKIPCA_OPNOTPER	3668	0xe54	skipCA is not permitted for this entity.
NSERR_SSL_ISSUER_MISMATCH	3669	0xe55	Certificate Issuer mismatch
NSERR_SSL_SET_POLICY_ACTION_TYPE	3670	0xe56	Action type cannot be changed from the previous configured action type
NSERR_SSL_DTLS_NOTSUPP	3671	0xe57	Server/Service of type DTLS is not supported on this platform
NSERR_SSL_INVALID_CN_NAME	3672	0xe58	Invalid Common Name.
NSERR_SSL_ECC_NOT_SUPPORTED	3673	0xe59	ECDHE ciphers supported only on FE SSL entities on VPX, MPX and BE MPX

ERROR CODE	ERROR CODE(DECIMAL)	ERROR CODE(HEX)	ERROR MESSAGE
NSERR_SSL_NO_PROTOCOL_ENABLED	3674	0xe5a	SSL or TLS protocols not enabled on the service.
NSERR_SSL_SSL2_NOT_SUPPORTED	3675	0xe5b	SSLv2 not supported in this release
NSERR_DTLS_PROFILE_REFEXT	3676	0xe5c	DTLS profile is referenced by a vserver or front-end service.
NSERR_SSL_NO_ECC_CURVES	3677	0xe5d	No ECC curves bound for ECDHE ciphers.
NSERR_CRYPTODEV_MAX_LIMIT	3678	0xe5e	Crypto Device count exceeds maximum available.
NSERR_SSL_PROFILE_ATTACHED	3679	0xe5f	Operation not permitted. Use set ssl profile for setting these parameters.
NSERR_SSL_PROFILE_USED	3680	0xe60	Profile is being used by Virtual Server, Service or Monitor.
NSERR_SSL_PROFILE_NOT_VAILD_PARAM	3681	0xe61	Specified parameters are not applicable for this type of SSL profile.
NSERR_IMPORT_SSL_INVALID_DHFILE_ERROR_OBJECT	3682	0xe62	Import failed:DH file being imported is invalid.
NSERR_IMPORT_SSL_INVALID_CRLFI	3683	0xe63	Import failed:CRL file being imported is invalid.

ERROR CODE	ERROR CODE(DECIMAL)	ERROR CODE(HEX)	ERROR MESSAGE
LE_ERROR_OBJECT			
NSERR_IMPORT_SSL_INVALID_CERTFILE_ERROR_OBJECT	3684	0xe64	Import failed:Certificate file being imported is invalid
NSERR_IMPORT_SSL_INVALID_KEYFILE_ERROR_OBJECT	3685	0xe65	Import failed:Key file being imported is invalid
NSERR_FIPSFW_FILEPATH	3686	0xe66	Invalid firmware file path
NSERR_FIPSFW_FILEOPEN	3687	0xe67	Unable to open firmware file
NSERR_FIPSFW_FILEIO	3688	0xe68	Read error occurred for firmware file
NSERR_FIPSFW_FSTAT	3689	0xe69	Failed to get status of firmware file
NSERR_FIPSFW_AUTH_FILEPATH	3690	0xe6a	Invalid firmware signature file path
NSERR_FIPSFW_AUTH_FILEOPEN	3691	0xe6b	Unable to open firmware signature file
NSERR_FIPSFW_AUTH_FILEIO	3692	0xe6c	Read error occurred for firmware signature file
NSERR_FIPSFW_AUTH_FSTAT	3693	0xe6d	Failed to get status of firmware signature file

ERROR CODE	ERROR CODE(DECIMAL)	ERROR CODE(HEX)	ERROR MESSAGE
NSERR_FIPSFW_BEGIN_ERROR	3694	0xe6e	Firmware update process failed BEGIN command
NSERR_FIPSFW_UPDATE_ERROR	3695	0xe6f	Firmware update process failed UPDATE command
NSERR_FIPSFW_END_ERROR	3696	0xe70	Firmware update process failed END command
NSERR_FIPSFW_STATE_ERROR	3697	0xe71	Firmware update process failed, invalid state
NSERR_FIPSFW_INVALID_CHUNK_TYPE	3698	0xe72	Firmware update process failed, invalid chunk type
NSERR_SSL_OCSP_WITH_CK_HSMKEY	3699	0xe73	Configuration of OCSP signing certificate with external HSM key not supported.
NSERR_SSL_HSMKEY_DEFLOCATION	3700	0xe74	Input HSM Key Simple file not present under the default directory /var/opt/nfast/kmdata/local/
NSERR_SSL_HSMKEY_IDENT_MISMATCH	3701	0xe75	HSM Key Name must be same as HSM Key Ident
NSERR_SSL_HSMKEY_DTLS	3702	0xe76	HSM Key bind with DTLS Vserver is not supported.

ERROR CODE	ERROR CODE(DECIMAL)	ERROR CODE(HEX)	ERROR MESSAGE
NSERR_SSL_HSMKEY_BUNDLE	3703	0xe77	Addition of certificate-bundle with external HSM Key is not supported.
NSERR_CPE_EXPRESSION_PARTITION_INVALID	3704	0xe78	Classic expression is not supported in current partition.
NSERR_AP_INVALID_EXPRESSION	3705	0xe79	Expression value not supported.
NSERR_SSL_DH_BUSY	3706	0xe7a	DH Param in use. Please try later...
NSERR_FIPSFW22_MINKEYSIZE	3707	0xe7b	Key size less than 2048 is not supported on FIPS Firmware Version 2.2.
NSERR_SSL_MAXSAN	3708	0xe7c	Maximum number of SANs reached
NSERR_SSL_BADSNICERT	3709	0xe7d	Bad SNI certificate
NSERR_SSL_AESGCM_SHA2_NOT_SUPPORTED	3710	0xe7e	SHA2 ciphers not supported on VPX and FIPS
NSERR_SSL_NOCIPHERGRP	3711	0xe7f	Cipher Group does not exist.
NSERR_SSL_SVCVSR_NOT_FOUND	3712	0xe80	No SSL vserver/ service found with this name.
NSERR_SSL_DEF_DTLS_PROFILE	3713	0xe81	Cannot modify default DTLS profile.

ERROR CODE	ERROR CODE(DECIMAL)	ERROR CODE(HEX)	ERROR MESSAGE
NSERR_SSL_NOSSLV2REDIRECTOPT_ON_FIPS	3714	0xe82	Sslv2 Redirect option is not permitted on MPX-FIPS platform.
NSERR_SSL_OP_ON_NONFIPS_NOT_PERM	3715	0xe83	Operation permitted only on MPX-FIPS platform.
NSERR_SSL_LOAD_KEYS	3716	0xe84	Loading of internal keys failed.
NSERR_SSL_PASSWORD_DECODE	3717	0xe85	Decode of the encrypted passphrase failed.
NSERR_SSL_NOCERT_FILE	3718	0xe86	Certificate file name not specified.
NSERR_SSL_DEFAULT_CERT_DEL	3719	0xe87	Cannot delete internal default certificate.
NSERR_SSL_FIPS_CERT	3720	0xe88	FIPS certificate cannot be bound as a CA certificate, install the certificate without the FIPS key.
NSERR_SSL_CERTKEY_HSM	3721	0xe89	Cannot bind a certificate with HSM key to a service.
NSERR_SSL_DEFAULT_CERT_BIND	3722	0xe8a	Cannot bind internal default certificate.
NSERR_SSL_CMD_DEPRECATED	3723	0xe8b	Command deprecated, operation not applicable to this platform.

ERROR CODE	ERROR CODE(DECIMAL)	ERROR CODE(HEX)	ERROR MESSAGE
NSERR_SSL_NO_LIC	3724	0xe8c	Only export ciphers are allowed without proper SSL license.
NSERR_SSL_DTLS_NA	3725	0xe8d	Command arguments does not apply to vserver/ service of type DTLS.
NSERR_SSL_ECC_SUPPORT	3726	0xe8e	ECC curve is not supported for this entity/platform.
NSERR_SSL_PROFILE_FE	3727	0xe8f	Cannot bind front-end profile to a back-end SSL service.
NSERR_SSL_PROFILE_BE	3728	0xe90	Cannot bind back-end profile to a front-end SSL vserver.
NSERR_SSL_OP_ON_FIPS_WOFW22_NOT_PERM	3729	0xe91	Operation not permitted on MPX-FIPS platform without FIPS firmware version 2.2 or higher.
NSERR_SSL_NONSSL_VSERVER	3730	0xe92	Vserver is not of type SSL.
NSERR_SSL_UDP_DTLS	3731	0xe93	Cannot bind policy to vserver of type DTLS.
NSERR_SSL_TLS11_12_SUPPORT	3732	0xe94	Enabling of TLSv1.1/1.2 is not supported on this entity/platform.

ERROR CODE	ERROR CODE(DECIMAL)	ERROR CODE(HEX)	ERROR MESSAGE
NSERR_SSL_HSM_KEY_SVC_SVCGRP	3733	0xe95	Binding of certificate with HSM key is not supported on back-end service, serviceGroup or monitors.
NSERR_DTLS_PROFILE	3734	0xe96	Cannot bind DTLS profile to a SSL vserver.
NSERR_SSL_PROFILE_NO_USABLE_CIPHERS	3735	0xe97	No usable ciphers configured for some of the SSL vserver/ service binded to this profile
NSERR_SSL_CIPHGRP_NO_USABLE_CIPHERS	3736	0xe98	No usable ciphers configured for some of the SSL vserver/ service binded to this cipher group
NSERR_SSL_PROFILE_UNSET_DEFAULT	3737	0xe99	Unsetting a default SSL Profile is not allowed
NSERR_SSL_DEFAULT_PROFILE_ENABLED	3738	0xe9a	Disabling ssl default profile is not allowed
NSERR_SSL_PROFILE_DEFAULT_DISABLED	3739	0xe9b	Operation not permitted. To do this Enable default ssl profile by setting set ssl parameter -defaultProfile E

ERROR CODE	ERROR CODE(DECIMAL)	ERROR CODE(HEX)	ERROR MESSAGE
NSERR_SSL_PROFILE_DEFAULT_ENABLED	3740	0xe9c	Operation not permitted. To do this Disable default ssl profile by setting set ssl parameter -defaultProfile D
NSERR_SSL_CIPHER_EXIST_HIGHER_PRI	3741	0xe9d	Specified cipher is already bound with higher priority
NSERR_SSL_DEFAULT_PROFILE_PERM	3742	0xe9e	Default ssl profile can not be removed
NSERR_SSL_DH_SIZE_16B	3743	0xe9f	DH params of size is not 16B aligned
NSERR_SSL_NOCIPHERREDIRECTOPT	3744	0xea0	The cipher redirect option is not supported for SSL Service
NSERR_SSL_NOCLIENTAUTHOPT	3745	0xea1	The client auth option is not supported for SSL Service
NSERR_SSL_NOSSLV2REDIRECTOPT	3746	0xea2	The SSLv2 redirect option is not supported for SSL Service
NSERR_SSL_NONONFIPSCIPHERSOPT	3747	0xea3	The nonFipsCiphers option is not applicable any more
NSERR_SSL_NOCIPHERREDIRECTOPT_ON_FIPS	3748	0xea4	Cipher Redirect option is not

ERROR CODE	ERROR CODE(DECIMAL)	ERROR CODE(HEX)	ERROR MESSAGE
			permitted on MPX-FIPS platform.
NSERR_SSL_NOSSL2PROTOOPT_ON_FIPS	3749	0xea5	Ssl2 protocol option is not permitted on MPX-FIPS platform.
NSERR_SSL_CANNOT_BIND_VSERVER	3750	0xea6	Front end service cannot be bound to a vserver.
NSERR_SSL_SERVERAUTHNOTSUPP	3751	0xea7	ServerAuth Operation is not supported for front end service.
NSERR_SSL_SERVERAUTH_SNI_REQUIRED	3752	0xea8	Argument pre-requisite missing serverAuth == ENABLED or SNIEnable == ENABLED
NSERR_SSL_CN_REQUIRED	3753	0xea9	Argument pre-requisite missing commonName
NSERR_SSL_DTLSPROFNOTSUPP	3754	0xeaa	Cannot set DTLS Profile Name for a ssl service.
NSERR_SSL_PUSHENCNOTSUPP	3755	0xeab	Push encryption trigger is not supported for SSL Service.
NSERR_N3FIPS_MINKEYSIZE	3756	0xeac	Key size less than 2048 is not supported on N3FIPS

ERROR CODE	ERROR CODE(DECIMAL)	ERROR CODE(HEX)	ERROR MESSAGE
NSERR_N3FIPS_MAXKEYSIZE	3757	0xead	Key size more than 3072 is not supported on N3FIPS
NSERR_N3FIPS_MINEXPSIZE	3758	0xeae	N3FIPS only supports F4(65537) as an exponent
NSERR_SSL_OCSPNAMENOTSUPP	3759	0xeaf	Ocsp responder name should not start with ns_internal.
NSERR_SSL_OCSP_NOT_VALID_PARAM	3760	0xeb0	Specified parameters are not applicable for this type of ocsp responder.
NSERR_SSL_NOOCSPSTAP	3761	0xeb1	Ocsp stapling option is not supported for SSL Service.
NSERR_SSL_CREATE_AIA_OCSP_RESP_FAILED	3762	0xeb2	Failed to create/bind Internal Ocsp responder(using certkey AIA URL).
NSERR_N3FIPDEXUPGRADE	3763	0xeb3	N3FIPS VPX does not support upgrade firmware operation
NSERR_N3FIP_MAXKEYNUM	3764	0xeb4	Maximum Fips keys limit reached
NSERR_FIPSCARDCONFIGED	3765	0xeb5	Fips is already configured, please reset first to set fips again

ERROR CODE	ERROR CODE(DECIMAL)	ERROR CODE(HEX)	ERROR MESSAGE
NSERR_N3FIPSCHANGELIC	3770	0xeba	N3FIPS license is changed, please reboot the system
NSERR_SSL_OCSP_SET_NOT_ALLOWED_ON_AIA	3771	0xebb	Set operation on internal ocsp entity is not allowed
NSERR_SSL_TLS13_SUPPORT	3775	0xebf	Enabling of TLSv1.3 is not supported on this entity/platform.
NSERR_SSL_CLEARTEXTPORT_WARNING	3778	0xec2	Cannot set cleartextport for a ssl service.
NSERR_SSL_DH_SIZE_2048	3780	0xec4	DH params of size greater than 2048 bits not supported on this platform.
NSERR_SSL_DEFAULT_CERT_NAME	3802	0xeda	Cannot use this name in non-default partition.
NSERR_SSL_VS_NOT_ALLOWED	3803	0xedb	Vserver should be of type: TCP, SSL_BRIDGE or SSL.
NSERR_SSL_CACERTGROUP_EXIST	3566	0xdee	CA certificate group already exists.
NSERR_SSL_CACERTGROUP_NOTEXIST	3567	0xdef	CA certificate group does not exist.
NSERR_SSL_CACERTGROUP_REF	3579	0xdfb	Certificate group is referenced by SSL action(s).

ERROR CODE	ERROR CODE(DECIMAL)	ERROR CODE(HEX)	ERROR MESSAGE
NSERR_SSL_NOT_CACERT	3580	0xdfc	Certificate is not a CA certificate.
NSERR_SSL_GROUP_HAS_CACERTKEY	3581	0xdfd	CA certificate group already has an entry for this CA certificate-key.
NSERR_SSL_GROUP_HAS_NO_CACERTKEY	3583	0xdff	CA certificate group does not have an entry for this CA certificate-key.
NSERR_SSL_SNICERT_INVALID	3805	0xedd	Cannot bind SNI cert on an internal and backend service. SNI cert can be bound only to SSL vserver.
NSERR_SSL_NO_CERTKEY_FILE	3807	0xedf	CertKey files deleted from file system, do save config, unsaved reboot will not bring deleted certKey files.
NSERR_SSL_NO_PRIVATE_KEY	3827	0xef3	Private Key for the certificate-key pair is not present.

SSLVPN AAA ERRORS

SSLVPN AAA ERRORS

ERROR CODE	ERROR CODE(DECIMAL)	ERROR CODE(HEX)	ERROR MESSAGE
NSERR_ERR_AAA_LICENSE	2624	0xa40	MaxAAAUsers value more than allowed by license, ignored.
NSERR_USR_NOINTRAIP	2625	0xa41	No intranet IP bound to user
NSERR_USR_NOTCONFIGURED	2626	0xa42	User does not exist
NSERR_INVAL_AAA_GRP	2627	0xa43	Group does not exist
NSERR_INVAL_COMBNATION	2628	0xa44	NONE and ALL cannot be used simultaneously
NSERR_INVAL_MIP_IIP	2629	0xa45	Both Mapped IP and Intranet IP must be specified
NSERR_INVAL_MIPOFF_IIPOFF	2630	0xa46	Mapped IP and Intranet IP cannot be OFF at the same time
NSERR_USEREXIST	2631	0xa47	User already exists
NSERR_GROUPEXIST	2632	0xa48	Group already exists
NSERR_USERALREADYBOUND	2633	0xa49	User is already bound to the group
NSERR_NO_RADIUS_IP	2634	0xa4a	ServerIP is not configured in radius params

ERROR CODE	ERROR CODE(DECIMAL)	ERROR CODE(HEX)	ERROR MESSAGE
NSERR_NO_TACACS_IP	2635	0xa4b	ServerIP is not configured in tacacs params
NSERR_USERNOTBOUND	2640	0xa50	User is not bound to the group
NSERR_ENTITYNOTBOUND	2641	0xa51	Entity not bound
NSERR_GROUPNOTEXIST	2642	0xa52	Group does not exist
NSERR_INVALIDLOGLEVEL	2643	0xa53	NONE cannot be combined with other options
NSERR_DH_MISCONFIG	2644	0xa54	The first hop and second hop can not be enabled on the same vserver
NSERR_DH_IPPORT	2645	0xa55	The SG second hop ip and port are required
NSERR_DHINUSE	2646	0xa56	Bound double hop server cannot be removed
NSERR_INVAL_AAAGLOBAL_POLTYPE	2647	0xa57	Only preauthentication policies can be bound to AAA global
NSERR_NO_LDAP_IP	2648	0xa58	ServerIP is not configured in ldap params

ERROR CODE	ERROR CODE(DECIMAL)	ERROR CODE(HEX)	ERROR MESSAGE
NSERR_AAATM_LIC	2649	0xa59	Feature not licensed [AAA]
NSERR_AAATM_DISABLED	2650	0xa5a	Feature(s) not enabled [AAA]
NSERR_NO_AUTH_HOST	2651	0xa5b	No Authentication Host specified
NSERR_AUTH_ON	2652	0xa5c	Turn authentication off first
NSERR_KILLPENDING	2653	0xa5d	Another kill command in progress
NSERR_AAATM_NO_AUTH_VS	2654	0xa5e	No Authentication vserver name specified or the server does not exist
NSERR_AAATM_401AUTH_ON	2655	0xa5f	Turn off 401 based authentication first
NSERR_UNAUTHRZED	2656	0xa60	Unauthorized
NSERR_KILL_INPROGRESS	2657	0xa61	A kill session command is in progress. Try again later
NSERR_WI_FRM_NOTEXIST	2658	0xa62	Farm does not exist
NSERR_WI_FRM_LAST	2659	0xa63	At least one Farm should be configured: can not remove last farm

ERROR CODE	ERROR CODE(DECIMAL)	ERROR CODE(HEX)	ERROR MESSAGE
NSERR_WI_NOTINST	2660	0xa64	Web interface not installed
NSERR_WI_GENFAILED	2661	0xa65	Unable to generate website
NSERR_WI_SITE_EXIST	2662	0xa66	Site already exists
NSERR_WI_SITE_NOTEXIST	2663	0xa67	Site does not exist
NSERR_WI_SITE_INVAL_AGURL	2664	0xa68	Invalid agURL
NSERR_WI_SITE_INVAL_STAURL	2665	0xa69	Invalid staURL
NSERR_WI_SITE_ONLY_MPX	2666	0xa6a	WI can only be installed on Citrix ADC nCore builds
NSERR_WI_INSTFAILED	2667	0xa6b	Installation failed. Please check the log file /var/log/wicmd.log
NSERR_WI_MAXSITE_EXCD	2668	0xa6c	Maximum number of WI Sites exceeded. Please check the log file /var/log/wicmd.log
NSERR_WI_STAWITHOUTAGURL	2669	0xa6d	STAUrl cannot be specified without AGUrl.
NSERR_WI_AGURLWITHOUTSTA	2670	0xa6e	AGUrl cannot be specified without STAUrl.

ERROR CODE	ERROR CODE(DECIMAL)	ERROR CODE(HEX)	ERROR MESSAGE
NSERR_WI_RELWITHOUTAGURL	2671	0xa6f	SessionReliability cannot be ON without AGUrl.
NSERR_WI_AUTHWITHOUTAGURL	2672	0xa70	AuthenticationPoint cannot be specified without AGUrl.
NSERR_WI_TWOTKTWITHOUTREL	2673	0xa71	UseTwoTickets cannot be ON without SessionReliability.
NSERR_WI_TWOTKTWITHOUTSECSTA	2674	0xa72	UseTwoTickets cannot be ON without SecondSTAUrl.
NSERR_WI_SECSTASAME	2675	0xa73	SecondSTAUrl should be different from StaURL.
NSERR_WI_LICENSE	2676	0xa74	Web Interface on NS Feature not licensed.
NSERR_WI_SECSTAWITHOUTSTA	2677	0xa75	SecondSTAUrl cannot be specified without STAUrl.
NSERR_AUTH_NEGOTIATE	2678	0xa76	Negotiate authentication required
NSERR_WI_INSTSITESREDUCED	2679	0xa77	Memory available is not sufficient for the passed maxSites value

ERROR CODE	ERROR CODE(DECIMAL)	ERROR CODE(HEX)	ERROR MESSAGE
NSERR_WI_INCOMPATIBLEAUTHPOINT	2680	0xa78	WI Authentication methods can not be specified unless authentication point is WI
NSERR_WI_SITE_WITHIN_SITE	2681	0xa79	One WI site can not be completely within another WI site
NSERR_WI_SITE_TRANSLATION_NOEXIST	2682	0xa7a	Translation Map does not exist
NSERR_WI_SITE_TRANSLATION_LAST	2683	0xa7b	Cant Unbind Last Translation Map
NSERR_WI_SITE_NOAGURL	2684	0xa7c	AGURL is not set
NSERR_AGSVC_AUTHFAIL	2685	0xa7d	AG-Service request with invalid AAAC cookie
NSERR_TM_INVALID_PERS_CONFIG	2686	0xa7e	Please specify both persistentCookie and persistentCookieValidity parameters
NSERR_WI_SMARTCARD_NOT_SUPPORTED	2687	0xa7f	SmartCard AGAuthentication method is not supported with sites created using wi package
NSERR_WI_SEARCH_NOT_SITEWEB	2688	0xa80	Show Search Attribute supported only for site type XenAppWeb

ERROR CODE	ERROR CODE(DECIMAL)	ERROR CODE(HEX)	ERROR MESSAGE
NSERR_KCD_ACCOUNT_NOT_EXISTS	2689	0xa81	The KCD account does not exist
NSERR_KCD_KEYTAB_NOT_EXISTS	2690	0xa82	The keytab file does not exist
NSERR_KCD_VS_PRINCIPLE_NOT_FOUND	2691	0xa83	The Vserver principle not found
NSERR_KCD_REM_INUSE	2692	0xa84	KCD account is in use and cannot be removed
NSERR_KCD_TOO_MANY_VS_PRINCIPLE	2693	0xa85	Too many host principles
NSERR_WI_REFRESH_NOT_SITEWEB	2694	0xa86	Show Refresh Attribute supported only for site type XenAppWeb
NSERR_WI_UIMODE_NOT_SITEWEB	2695	0xa87	User Interface Mode Attribute supported only for site type XenAppWeb
NSERR_WI_UILAYOUT_NOT_SITEWEB	2696	0xa88	User Interface Layout Attribute supported only for site type XenAppWeb
NSERR_WI_MESSAGESTRS_NOT_SITEWEB	2697	0xa89	App Welcome Message, Welcome Message, Footer Text, Login Sys Message, Pre Login Button, Pre Login

ERROR CODE	ERROR CODE(DECIMAL)	ERROR CODE(HEX)	ERROR MESSAGE
			Message, Pre Login Title Attributes supported only for site type XenAppWeb
NSERR_AAATM_AUTHN_PROFILE_CONF	2704	0xa90	Please turn ON authentication or authn401 setting in order to use authentication profile
NSERR_NEGACTION_UNAME_DOM	2705	0xa91	Domain must either be specified either explicitly or in the username
NSERR_NO_PRODUCTION_FARM	2706	0xa92	WI site will not be fuctional till a production farm is bound
NSERR_SAML_FORM_CONFLICT	2707	0xa93	Traffic cannot have both SAML SSO and Form SSO on same action
NSERR_MAXLOGIN_FAILLOGIN_CONFLICT	2708	0xa94	Please specify both max login attempts and fail login timeout
NSERR_KCD_ACCOUNT_EXIST	2709	0xa95	The KCD account does not exist
NSERR_KCD_ACCOUNT_NOTCONFIGURED	2710	0xa96	The KCD account is not configured
NSERR_NEGACTION_KEYTAB_CONFLICT	2711	0xa97	Action should have either keytab file or domain, username

ERROR CODE	ERROR CODE(DECIMAL)	ERROR CODE(HEX)	ERROR MESSAGE
			and password. Domain need not be specified if UPN is given for username
NSERR_KEYTAB_INVALID_ENC	2712	0xa98	Unsupported encryption type in keytab file
NSERR_HTTP_SSL_PROXY_CONFLICT	2713	0xa99	Cannot configure both HTTP and SSL proxy
NSERR_INVALIDIP_PORT	2714	0xa9a	Invalid IP or Port
NSERR_ADVEPA_NOTSUPP	2715	0xa9b	VServer does not support advanced EPA scans
NSERR_ADVEPA_NOPROFILE	2716	0xa9c	EPA profile with this name does not exist
NSERR_ADVEPA_INUSE	2717	0xa9d	Bound EPA profile cannot be removed.
NSERR_ADVEPA_EXISTS	2718	0xa9e	EPA profile exists.
NSERR_ADVEPA_BOUND	2719	0xa9f	EPA profile already bound to vserver.
NSERR_ADVEPA_NOTBOUND	2720	0xaa0	EPA profile not bound to vserver.
NSERR_ADVEPA_ON	2721	0xaa1	Advanced EPA is ON, can not bind classic preauthentication policy.

ERROR CODE	ERROR CODE(DECIMAL)	ERROR CODE(HEX)	ERROR MESSAGE
NSERR_DEVICEPROFILE_ADDRM_FAIL	2722	0xaa2	There has been a design change in the support of OPSWAT specific EPA scans. EPA Profile Configuration is no longer needed. Please refer to support documents for complete details.
NSERR_DEVICEPROFILE_INTERNAL_ERR	2723	0xaa3	An Internal Error occurred while operating on the Device Profile, please contact your Admin
NSERR_AAA_SESS_INVALID_INDEX	2724	0xaa4	Invalid attribute index. Attribute index should be between 1 and 16 both included
NSERR_AAATM_FORCETIMEOUT	2725	0xaa5	Force timeout setting of START or RESET must accompany a desired time in minutes
NSERR_KCDACCOUNT_KEYTAB	2726	0xaa6	Keytab file cannot be specified along with other parameters in kcdaccount
NSERR_KCDACCOUNT_REALM	2727	0xaa7	Realm must be specified by itself or extracted from delegated username/ keytab

ERROR CODE	ERROR CODE(DECIMAL)	ERROR CODE(HEX)	ERROR MESSAGE
NSERR_KCDACCOUNT_KEYTAB_INVALID	2238	0x8be	Invalid Keytab file
NSERR_WEBAUTH_FULLEXP	2728	0xaa8	Please specify full request expression
NSERR_WI_INVALID_SETTING_FOR_AUTHPOINT	2729	0xaa9	Setting is Invalid for given Authpoint
NSERR_WI_SETTING_VALIDONLYFOR_XENAPPWEB	2730	0xaaa	Setting is Invalid for SiteType other than XenAppWeb
NSERR_WI_PARAMETER_UPGARDE_ERROR	2731	0xaab	Migration of WebInterface.conf parameter to CLI didnt happen successfully
NSERR_SAMLIDP_ENC_CERT	2732	0xaac	Encryption option requires ServiceProvider (SP) certificate.
NSERR_WF_NOT_INST	2733	0xaad	Web Front is not installed
NSERR_WF_INSTFAILED	2734	0xaae	Installation failed. Please check the log file /var/log/wfcmd.log
NSERR_AAA_TOOMANY_EXP	2735	0xaaf	Number of expressions found exceeds allowed limit
NSERR_SAML_REDIR_CONFLICT	2736	0xab0	Signature and Digest algorithms are

ERROR CODE	ERROR CODE(DECIMAL)	ERROR CODE(HEX)	ERROR MESSAGE
			incompatible for Redirect Binding
NSERR_SET_MAXAAAUSER_NOT_ALLOWED	2737	0xab1	Set maxaaaUser not allowed per partition
NSERR_AAATM_INCOMP_NO_PROFILE	2744	0xab8	Profile does not exist or does not have Host configured
NSERR_RENAME_AUTHN_LABEL_NOT_PERMITTED	2745	0xab9	Rename of authentication policyLabel not permitted if it referred as nextFactor
NSERR_ERR_AAA_LICENSE_LOW	2747	0xabb	MaxAAAUsers value less than allowed by license, ignored.
NSERR_ERR_ENABLE_DTLS	2750	0xabe	Cannot start DTLS or PCOIP listener due to existing (ip, port, transport) conflicts.
NSERR_OPENID_ENC_METADATA	2751	0xabf	Token encryption requires Relying Party metadata url.
NSERR_OPENID_MAX_ATTRIBUTES	2814	0xafe	Expression exceeds maximum permissble attributes.

SST ERRORS

SST ERRORS

ERROR CODE	ERROR CODE(DECIMAL)	ERROR CODE(HEX)	ERROR MESSAGE
NSERR_SST_SUBSCR_ALLOC_FAILED	3504	0xdb0	Cannot allocate memory for subscriber in extended memory space
NSERR_SST_SUBSCR_LOOKUP_FAILED	3505	0xdb1	Cannot find subscriber
NSERR_SST_SUBSCR_REQUEST_ERROR	3506	0xdb2	Invalid request for subscriber
NSERR_GX_SERVICE_CONFIG_XOR	3507	0xdb3	service and vserver cannot be set at the same time.
NSERR_GX_NOT_DIAMETER_VS	3508	0xdb4	Not a DIAMETER or SSL_DIAMETER vserver.
NSERR_GX_NOT_DIAMETER_SERVICE	3509	0xdb5	Not a DIAMETER or SSL_DIAMETER service.
NSERR_SST_SUBSCRIBER_ENTRY_EXCEEDED	3510	0xdb6	Size of subscriber entry exceeds total allowed size.
NSERR_GX_PCRF_NOT_CONFIGURED_VSERVER_SERVICE	3511	0xdb7	PCRF not configured. To configure PCRF, please use set gxInteface -vserver/service

ERROR CODE	ERROR CODE(DECIMAL)	ERROR CODE(HEX)	ERROR MESSAGE
NSERR_SST_STATIC_SUBSCRIBER_SESSION_DELETION	3512	0xdb8	Static subscriber profiles cannot be deleted using this command, please use rm subscriber profile
NSERR_SST_SUBSCRIBER_SESSION_EXIST	3513	0xdb9	subscriber profile cannot be added as corresponding subscriber session already exists
NSERR_SST_NOT_RADIUS_SERVICE	3514	0xdba	Not RADIUS Listener service
NSERR_MAX_SERVICE_FUNCTIONS	3515	0xdbb	Number for service functions exceeding the limit
NSERR_NOSUCH_SVCFUNC	3516	0xdbc	Service Function is not present with specified name
NSERR_NOSUCH_SVCPATH	3517	0xdbd	Service Path is not present with specified name
NSERR_INVALID_SVCINDEX	3518	0xdbe	The value of service index is invalid
NSERR_SVCPATH_SVCINX_EXISTS	3519	0xdbf	A service function is already bound with given service index

SST ERRORS2

SST ERRORS2

ERROR CODE	ERROR CODE(DECIMAL)	ERROR CODE(HEX)	ERROR MESSAGE
NSERR_SST_IPANDVLAN_GXONLY	3799	0xed7	IPANDVLAN keytype can only be set in GxOnly interface type mode.
NSERR_VLAN_NOT_ALLOWED_DEFAULT_SUBSCRIBER_PROFILE	3800	0xed8	VLAN ID is not allowed for the default subscriber profile.

SVPN ERRORS

SVPN ERRORS

ERROR CODE	ERROR CODE(DECIMAL)	ERROR CODE(HEX)	ERROR MESSAGE
NSCFG_INFO	2049	0x801	Sending the /cfg information
NSCS_INFO	2050	0x802	Backend server info exists
NSCSPROBE_INFO	2051	0x803	Probe to backend server pending
NSAPPPROBE_INFO	2052	0x804	Client side connection being closed
NSCFG_MP_INFO	2112	0x840	Sending the /mp_cfg Information
NSERR_NOINTRANETIP	2113	0x841	No Intranet IP available

ERROR CODE	ERROR CODE(DECIMAL)	ERROR CODE(HEX)	ERROR MESSAGE
NSERR_ALREADYLOGEDIN	2114	0x842	The user is already logged-in
NSERR_URLINUSE	2115	0x843	Bound URL/ bookmark cannot be removed.
NSERR_VPNAPPINUSE	2116	0x844	Bound VPN application cannot be removed.
NSERR_NOTSUPP_TRANS_INTERCPT	2117	0x845	Transparent interception is not yet supported.
NSERR_CLNT_CERT_RENEG	2118	0x846	Client SSL certificate renegotiation failed
NSERR_DEFAULTCMDPLCY	2119	0x847	Default command policy cannot be removed
NSERR_CLNT_CERT_REQD	2120	0x848	Client SSL certificate is required
NSERR_INVAL_CERTFIELD	2121	0x849	Invalid certificate field
NSERR_VPNAPP_PROXY_IPRANGE	2064	0x810	Proxy interception does not support IP ranges
NSERR_VPNAPP_PROXY_NETMASK	2065	0x811	Proxy interception does not support netmasks
NSERR_VPNAPP_PROXY_DSTPORT_RANGE	2066	0x812	Proxy interception does not support destination port ranges

ERROR CODE	ERROR CODE(DECIMAL)	ERROR CODE(HEX)	ERROR MESSAGE
NSERR_VPNAPP_PROXY_PROTOCOL	2067	0x813	Proxy interception supports only TCP
NSERR_VPNAPP_PROXY_HOSTNAME	2068	0x814	Proxy interception does not support hostname interception
NSERR_VPNAPP_TRANS_SRCIP	2069	0x815	Transparent interception does not support source IP
NSERR_VPNAPP_TRANS_SRCPORT	2070	0x816	Transparent interception does not support source port
NSERR_VPNAPP_NO_INTERCEPTION_TYPE	2071	0x817	Intranet application requires an interception type
NSERR_VPNAPP_CLIAPP_PORT	2072	0x818	Both client application name and destination port cannot be specified
NSERR_VPNAPP_CLIAPP_PROTO	2073	0x819	Protocol can not be specified when client application name is present
NSERR_VPNAPP_PROXY_CLIAPP	2074	0x81a	Proxy interception does not support client application based interception
NSERR_VPNAPP_MISSING_PROTO	2075	0x81b	Protocol must be specified
NSERR_VPNAPP_MISSING_ARG	2076	0x81c	One of destIP, IPRange or

ERROR CODE	ERROR CODE(DECIMAL)	ERROR CODE(HEX)	ERROR MESSAGE
			hostname has to be specified
NSERR_VPNAPP_TOO_MANY_ARG	2077	0x81d	At most one of destIP, IPRange or hostname may be specified
NSERR_FS_AUTHFAIL	2078	0x81e	?
NSERR_NSIPV6NOTPRESENT	2079	0x81f	No IPV6 Citrix ADC IP has been configured
NSERR_REMOVE_SESSION	2096	0x830	No IPV6 Citrix ADC IP has been configured
NSERR_STAWI_EXIST	3250	0xcb2	A STA or WI DBS configuration exists. Unset it first
NSERR_INVALID_SSO_ACTION	2099	0x833	Invalid sso action
NSERR_INVALID_TMTRAFFIC_ACTION	2100	0x834	Invalid tm traffic action
NSERR_INVALFSSO	2101	0x835	SSO should be turned on for setting formsso action
NSERR_INVALIDURL	2102	0x836	Action url should be root relative url
NSERR_INVALID_FLOWTYPE_FORAPPFLOW	2126	0x84E	Flowtype needs to be REQUEST or ICA_REQUEST or OTHERTCP_REQUEST for appflow

ERROR CODE	ERROR CODE(DECIMAL)	ERROR CODE(HEX)	ERROR MESSAGE
			policy when bound to VPN vservers
NSERR_TOO_MANY_SERVERS	2127	0x84F	Too many servers bound
NSERR_BLOCKING_ATTR_WARNING_FOR_VPN_APPFLOW	2961	0xb91	Policy will not apply to VPN traffic if it has blocking attributes
NSERR_SAMLIDP_ACTION	2975	0xb9f	Invalid SAML IDP Action
NSERR_SAML_ACTION_AUTHNCTX	2976	0xba0	None option is mutually exclusive with other Authentication Class References
NSERR_ABSOLUTE_URL	2977	0xba1	URL has to be an absolute URL
NSERR_INVALID_AG_BIND	2978	0xba2	SSLVPN vserver can be bound to only SSL type CS vserver
NSERR_NO_VSERVER	2979	0xba3	Please specify the associated vserver name
NSERR_SSSO_NOT_LICENSED	2980	0xba4	This feature needs enterprise or platinum license.
NSERR_MULT_AGBIND	2981	0xba5	Only one VPN vserver can be bound to a CS vserver.

ERROR CODE	ERROR CODE(DECIMAL)	ERROR CODE(HEX)	ERROR MESSAGE
NSERR_VPN_VSERV_PROTOCOL_NOT_ALLOWED	2982	0xba6	VPN/Authentication Vserver of service type HTTPis not allowed.
NSERR_SET_NOT_ALLOWED_ON_ICAONLY_BIT	2983	0xba7	Changing ICAonlyparameter is not allowed in Cluster Deployment.
NSERR_IIP_INVALID_RANGE	2984	0xba8	The given intranet IP binding does not exist
NSERR_SSLVPN_LICENSE	2985	0xba9	Feature not licensed [SSLVPN]
NSERR_SAMLIDP_REDIR_INVALID_SIGN	2990	0xbae	signAssertion option can only be set to ASSERTION in Redirect binding
NSERR_VPN_AUTH_BIND	2991	0xbaf	Authentication vserver can not be bound to a CS vserver if VPN vserver is already bound to it.
NSERR_AUTH_VPN_BIND	2992	0xbb0	VPN vserver can not be bound to a CS vserver if Authentication vserver is already bound to it.
NSERR_SVPN_INVALID_URL	2993	0xbb1	URL must be an absolute url with a trailing slash

ERROR CODE	ERROR CODE(DECIMAL)	ERROR CODE(HEX)	ERROR MESSAGE
NSERR_MULT_AUTHBIND	2994	0xbb2	Only one Authentication vserver can be bound to a CS vserver.
NSERR_SVPN_INVALID_ICONURL	2995	0xbb3	Only the following image types i.e. .png, .jpeg, .jpg and .giff are allowed as bookmark icons.
NSERR_XM_PKG_CONTENT	2996	0xbb4	XenMobile package content error. Please find more information in /var/log/nsxm.log
NSERR_XM_MISSING_TOKEN_VALUE	2997	0xbb5	XenMobile script tokens replacement failure. Please find more information in /var/log/nsxm.log
NSERR_SAML_ARTIFACT_NO_URL	2998	0xbb6	Artifact resolution url is required when binding is specified as ARTIFACT
NSERR_OAUTHIDP_ACTION	2999	0xbb7	Invalid OAuth IDP Action
NSERR_NOLICENCE_PUSH	3000	0xbb8	Platinum license is required for this operation
NSERR_INVALSAMLSSO	3001	0xbb9	SSO should be turned on for setting samlsso action

ERROR CODE	ERROR CODE(DECIMAL)	ERROR CODE(HEX)	ERROR MESSAGE
NSERR_INVALSAMLIDPATTR1	2512	0x9d0	Incorrect expression at attribute 1
NSERR_INVALSAMLIDPATTR2	2513	0x9d1	Incorrect expression at attribute 2
NSERR_INVALSAMLIDPATTR3	2514	0x9d2	Incorrect expression at attribute 3
NSERR_INVALSAMLIDPATTR4	2515	0x9d3	Incorrect expression at attribute 4
NSERR_INVALSAMLIDPATTR5	2516	0x9d4	Incorrect expression at attribute 5
NSERR_INVALSAMLIDPATTR6	2517	0x9d5	Incorrect expression at attribute 6
NSERR_INVALSAMLIDPATTR7	2518	0x9d6	Incorrect expression at attribute 7
NSERR_INVALSAMLIDPATTR8	2519	0x9d7	Incorrect expression at attribute 8
NSERR_INVALSAMLIDPATTR9	2520	0x9d8	Incorrect expression at attribute 9
NSERR_INVALSAMLIDPATTR10	2521	0x9d9	Incorrect expression at attribute 10
NSERR_INVALSAMLIDPATTR11	3002	0xbba	Incorrect expression at attribute 11
NSERR_INVALSAMLIDPATTR12	2523	0x9db	Incorrect expression at attribute 12
NSERR_INVALSAMLIDPATTR13	2524	0x9dc	Incorrect expression at attribute 13
NSERR_INVALSAMLIDPATTR14	2525	0x9dd	Incorrect expression at attribute 14

ERROR CODE	ERROR CODE(DECIMAL)	ERROR CODE(HEX)	ERROR MESSAGE
NSERR_INVALSAMLIDPATTR15	2526	0x9de	Incorrect expression at attribute 15
NSERR_INVALSAMLIDPATTR16	2527	0x9df	Incorrect expression at attribute 16
NSERR_PUSH_SERVICE_AZURE_ARGS	2550	0x9f6	Azure type service requires Namespace, Hub, Servicekey and Servicekey name but not Trust service
NSERR_PUSH_SERVICE_NS_ARGS	2551	0x9f7	Citrix type service requires Client id and secret
NSERR_INVALID_DFA_BINDPOINT	2552	0x9f8	DFA policies cannot be bound to user defined policylabels/ factors

TD ERRORS

TD ERRORS

ERROR CODE	ERROR CODE(DECIMAL)	ERROR CODE(HEX)	ERROR MESSAGE
NSERR_TD_EXISTS	944	0x3b0	The specified traffic domain is already configured.
NSERR_TD_VLAN	945	0x3b1	The specified VLAN/ VXLAN/bridge group is already bound to another traffic domain. These

ERROR CODE	ERROR CODE(DECIMAL)	ERROR CODE(HEX)	ERROR MESSAGE
			entities can belong to only one traffic domain.
NSERR_TD_NOTEXIST	946	0x3b2	The specified traffic domain is not configured.
NSERR_TD_MISMATCH_WITH_L3VLAN	947	0x3b3	L3 VLANs can not be bound to a traffic domain.
NSERR_TD_MISMATCH	948	0x3b4	The binding entities have incompatible traffic domain identifiers.
NSERR_TD_NOSUPP	949	0x3b5	This feature is not supported in a non-default traffic domain.
NSERR_TD_CONFIG_NOT_ALLOWED	950	0x3b6	This configuration option is not supported in a non-default traffic domain.
NSERR_TD_IP_NOTEXIST	951	0x3b7	The IP address is not configured for the specified traffic domain.
NSERR_TD_VLAN_NOTEXIST	952	0x3b8	The specified VLAN is not configured.
NSERR_TD_DEFVLAN	953	0x3b9	Default VLAN cannot be bound/unbound to/from traffic domain.

ERROR CODE	ERROR CODE(DECIMAL)	ERROR CODE(HEX)	ERROR MESSAGE
NSERR_TD_BDG_NOTEXIST	954	0x3ba	The specified bridge group is not configured.
NSERR_TD_VLAN_BDG_EXIST	955	0x3bb	The specified VLAN is bound to a bridge group.
NSERR_TD_SERVER_MISMATCH	956	0x3bc	The specified virtual server already bound to a service in a different traffic domain.
NSERR_DIFF_TD_NALLOWED_LLB	957	0x3bd	Not allowed for LLB: Services bound to the specified virtual server must be in same traffic domain.
NSERR_DUPIP	958	0x3be	IP address duplication with VMAC address detected. Please try a different IP address.
NSERR_VMAC_VLAN	959	0x3bf	Traffic domain can be either VLAN based or VMAC based.
NSERR_TD_SYNCVLAN	960	0x3c0	SYNC VLAN cannot be bound/unbound to/from traffic domain.
NSERR_TD_NOT_VLAN	961	0x3c1	The specified VLAN/VXLAN/bridge group

ERROR CODE	ERROR CODE(DECIMAL)	ERROR CODE(HEX)	ERROR MESSAGE
			is not bound to given traffic domain.
NSERR_SYNCNSVLAN_TD	975	0x3cf	VLAN bound to traffic domain cannot be configured as SYNC/NS VLAN.

URLFILT ERRORS

URLFILT ERRORS

ERROR CODE	ERROR CODE(DECIMAL)	ERROR CODE(HEX)	ERROR MESSAGE
NSERR_URLFILT_CANNOT_PARSE_XML	3772	0xebc	Cannot parse XML categories file.
NSERR_URLFILT_CANNOT_FIND_GROUP	3773	0xebd	Cannot find the specified category group.
NSERR_URLFILT_XML_FILE_TOO_BIG	3774	0xebe	XML categories file is too big.

URLSET ERRORS

URLSET ERRORS

ERROR CODE	ERROR CODE(DECIMAL)	ERROR CODE(HEX)	ERROR MESSAGE
NSERR_URLSET_CANNOT_INIT_PATTERN	3768	0xeb8	Cannot initialize urlset pattern.

ERROR CODE	ERROR CODE(DECIMAL)	ERROR CODE(HEX)	ERROR MESSAGE
NSERR_URLSET_CANNOT_FIND_URL	3769	0xeb9	Cannot find url in urlset.

URLTRANS ERRORS

URLTRANS ERRORS

ERROR CODE	ERROR CODE(DECIMAL)	ERROR CODE(HEX)	ERROR MESSAGE
NSERR_URLTRANS_MAX_ENTITIES	2960	0xb90	Number of URL Transformation entities exceeds limit
NSERR_URLTRANS_INVAL_PROFILE	2962	0xb92	Invalid URL Transformation profile
NSERR_URLTRANS_ACTION_INUSE	2963	0xb93	Action name is already in use
NSERR_URLTRANS_PRIORITY_EXISTS	2964	0xb94	An object with this priority already exists
NSERR_URLTRANS_REQ_PCRE_ERR	2965	0xb95	Invalid PCRE expression under reqUrlFrom
NSERR_URLTRANS_RESP_PCRE_ERR	2966	0xb96	Invalid PCRE expression under resUrlFrom
NSERR_URLTRANS_REQ_INTO_ERR	2967	0xb97	Invalid expression under reqUrlInto
NSERR_URLTRANS_RESP_INTO_ERR	2968	0xb98	Invalid expression under resUrlInto

ERROR CODE	ERROR CODE(DECIMAL)	ERROR CODE(HEX)	ERROR MESSAGE
NSERR_URLTRANS_COOKIE_INTO_ERR	2969	0xb99	Invalid expression under cookieDomainInto
NSERR_URLTRANS_COOKIE_PCRE_ERR	2970	0xb9a	Invalid PCRE expression under cookieDomainFrom
NSERR_URLTRANS_TOO_MANY_BACKREFS	2971	0xb9b	More than 5 back references in PCRE expression
NSERR_URLTRANS_MISSING_FROM	2972	0xb9c	Each Intoexpression must have a corresponding Frompattern
NSERR_URLTRANS_NOT_SUPPORTED_VS	2973	0xb9d	Specified policy can be bound only to HTTP/SSL/PROXY vserver
NSERR_URLTRANS_BAD_COMMENT	2974	0xb9e	Invalid comment

VLAN ERRORS

VLAN ERRORS

ERROR CODE	ERROR CODE(DECIMAL)	ERROR CODE(HEX)	ERROR MESSAGE
NSERR_INTERFACEBOUND	2080	0x820	Interface is already bound to this VLAN
NSERR_IFACE_NO_UNBIND	2081	0x821	Untagged interface cannot be removed from default VLAN.

ERROR CODE	ERROR CODE(DECIMAL)	ERROR CODE(HEX)	ERROR MESSAGE
			To remove, bind to other VLAN.
NSERR_IFACE_MAX_VLANS	2082	0x822	Maximum number of tagged VLANs bound to the interface exceeded or the binding of this VLAN is not allowed on the interface.
NSERR_VLAN_RTEXIST	2083	0x823	An existing route relies on the presence of this vlan
NSERR_VLAN_LLEXIST	2084	0x824	An existing link-local ip relies on the presence of this vlan
NSERR_BOUNDTONSVLAN	2085	0x825	Interface/Channel bound to NSVLAN
NSERR_VLAN_TD_CONFIGURED	2408	0x968	TD is configured on this bridgegroup/ vlan. Cannot delete this
NSERR_VLAN_BOUNDTO_NETBRIDGE	1216	0x4c0	Vlan bound to Netbridge cannot be removed

VMAC ERRORS

VMAC ERRORS

ERROR CODE	ERROR CODE(DECIMAL)	ERROR CODE(HEX)	ERROR MESSAGE
NSERR_INTERFACE_BOUND	3329	0xd01	Interface already bound
NSERR_VRID_INTERFACE_NOT_BOUND	3330	0xd02	Interface not bound
NSERR_NO_SUCH_INTERFACE	3331	0xd03	No such interface
NSERR_DYNAMIC_ENTRY	3346	0xd12	Dynamic Entry exists.
NSERR_PRIO_INTFVRID	3347	0xd13	Invalid Priority for interface bound VRID
NSERR_INTFVRID	3348	0xd14	Binding interface is not allowed for Dynamic or IP bound VRID
NSERR_NOTSTATIC_VRID	3349	0xd15	Not a Configured VRID
NSERR_IPVRID	3350	0xd16	IP binding is not allowed for Interface VRID
NSERR_IPVRID_BOUND	3351	0xd17	VRID already bound to this IP
NSERR_NOVRID	3354	0xd1a	VRID doesnt exist
NSERR_INTF_TRACKINTF_BOUND	3362	0xd22	Interfaces can either be tracked or bound to a vrID.

ERROR CODE	ERROR CODE(DECIMAL)	ERROR CODE(HEX)	ERROR MESSAGE
NSERR_VRRP_SHAREDVLAN_NOT_SUPPORTED	3408	0xd50	VRRP functionality is not supported on shared VLANs.
NSERR_INTF_VRID_BIND_NOTALLOWED	3868	0xf1c	Binding of interface to this VRID is not allowed. Plesae check if the VRID is allowed for the interface from SVM.

VXLAN ERRORS

VXLAN ERRORS

ERROR CODE	ERROR CODE(DECIMAL)	ERROR CODE(HEX)	ERROR MESSAGE
NSERR_VXLAN_INCOMP	2608	0xa30	Vlan extension and IP address - both cannot be configured on vxlan
NSERR_VXLAN_TUN_BOUND	2609	0xa31	Tunnel is already bound to this vxlan
NSERR_INV_MCAST	2610	0xa32	Another multicast group already configured for this vxlan
NSERR_TUN_NOT_BOUND	2611	0xa33	Tunnel not bound to this vxlan
NSERR_IP_NOT_BOUND	2612	0xa34	IP/IPv6 address not bound to this vxlan

ERROR CODE	ERROR CODE(DECIMAL)	ERROR CODE(HEX)	ERROR MESSAGE
NSERR_INV_TUN	2613	0xa35	Tunnel should be of VXLAN type
NSERR_NO_VXLAN	2614	0xa36	Vxlan is not configured
NSERR_VLAN_EXT	2615	0xa37	Vlan already extended by a different vxlan
NSERR_VXLAN_TD	2616	0xa38	Vxlan is bound to a traffic domain
NSERR_VXLAN_ROTUING	2617	0xa39	Vlan extension and routing - both cannot be configured on vxlan
NSERR_VXLANMAP_INUSE	2743	0xab7	This vlan-to-vxlan map table is in use by a netbridge/pbr-tunnel
NSERR_VXLAN_TBL_BOUND	2738	0xab2	Vxlan is in use by a vlan-to-vxlan map table
NSERR_TUN_MAP_TBL	2739	0xab3	The tunnel currently applies vlan-to-vxlan mappings
NSERR_VXLAN_INUSE	2740	0xab4	Vxlans that have vlan extensions or IP addresses or routing cannot be used in mapping table
NSERR_NETBDG_VXLAN_VLAN	2741	0xab5	Vlan-to-vxlan map table and vlan - both

ERROR CODE	ERROR CODE(DECIMAL)	ERROR CODE(HEX)	ERROR MESSAGE
			cannot be configured on a netbridge
NSERR_NO_VXLAN MAP	2742	0xab6	Vlan-to-vxlan map table is not configured
NSERR_VXLANMA P_TUN_NOSUP	2746	0xaba	Incompatible tunnel type for vlan-to-vxlan map table

WARNING ERRORS

WARNING ERRORS

ERROR CODE	ERROR CODE(DECIMAL)	ERROR CODE(HEX)	ERROR MESSAGE
NSERR_AUTHORIZ E		NS_WARNING| 0x001	?
NSERR_AUTHFAIL		NS_WARNING| 0x002	?
NSWAR_SSL_BIND OR		NS_WARNING| NSERR_SSL_BIND OR	Current certificate replaces the previous binding
NSWAR_SSL_BRKL INK		NS_WARNING| NSERR_SSL_BRKLI NK	All incompatible CA links/Cert Bindings were broken during the update operation
NSWAR_SSL_CERT _NOT_YET_VALID		NS_WARNING| NSERR_SSL_CERT _NOT_YET_VALID	The specified certificate is not yet valid

ERROR CODE	ERROR CODE(DECIMAL)	ERROR CODE(HEX)	ERROR MESSAGE
NSWAR_SSL_CERT_EXPIRED		NS_WARNING|NSERR_SSL_CERT_EXPIRED	The certificate has expired
NSWAR_SSL_EXPIRED_BRKLINK		NS_WARNING|NSERR_SSL_EXPIRED_BRKLINK	All incompatible CA links were broken during the update operation. (Note: the certificate has expired)
NSWAR_SSL_NYVALID_BRKLINK		NS_WARNING|NSERR_SSL_NYVALID_BRKLINK	All incompatible CA links were broken during the update operation. (Note: the certificate is not yet valid)

XS ERRORS

XS ERRORS

ERROR CODE	ERROR CODE(DECIMAL)	ERROR CODE(HEX)	ERROR MESSAGE
NSERR_CIPROF_INTERFACE_NOT_ALLOWED	2436	0x984	IngressInterface and Ingressvlan are not allowed for this contentinspection profile type
NSERR_XS_INTERNAL_RESOURCE_CREATION	3794	0xed2	Resource creation failed, internal error.
NSERR_XS_INVALID_SERVERNAME	3795	0xed3	Unsupported service type. Supported types are TCP and

ERROR CODE	ERROR CODE(DECIMAL)	ERROR CODE(HEX)	ERROR MESSAGE
			SSL_TCP for ICAP, TCP for INLINEINSPECTION and ANY for MIRROR.
NSERR_CI_ACT_IP_PORT_NOT_ALLOWED	3820	0xeec	IP and Port cant set be for this contentinspection action type
NSERR_CI_ACT_SNAME_SIP_CONFLICT	3796	0xed4	Server name and ip cant both be set for this content inspection entity.
NSERR_XS_INVALID_ICAPPROFILE_NAME	3797	0xed5	Invalid ICAP profile name.
NSERR_CI_PROFILE_EXISTS	3798	0xed6	ContentInspection profile exists with the given name.
NSERR_CIPROF_INUSE	3801	0xed9	Cant change MODE as the ContentInspection profile is in use.
NSERR_XS_INVALID_CIPROFILE_NAME	3766	0xeb6	Invalid contentInspection profile name.
NSERR_CI_UNSUPPORTED_PROFILE_TYPE	3767	0xeb7	Invalid Content Inspection type specified.
NSERR_CI_NO_PROFILE	3817	0xee9	Content Inspection profile does not exist.

ERROR CODE	ERROR CODE(DECIMAL)	ERROR CODE(HEX)	ERROR MESSAGE
NSERR_CI_INVALID_CIC	3821	0xeed	Content Inspection callout does not exist or type didnt match.
NSERR_CI_ACT_MISSING_PROFILE	3823	0xeef	ContentInspectionProfile is not set on service(s)